Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies

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

Deliverable D2.2
ETM Model and Components (Initial Version)

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 Model and Components (Initial Version)

Type: Public

Editor(s): Ioanna Papafili, Sergios Soursos, George D. Stamoulis

E-mail: iopapafi@aueb.gr, sns@aueb.gr, gstamoul@aueb.gr

Authors: Ioanna Papafili, Sergios Soursos, George D. Stamoulis, Jan Derkacz, Miroslaw Kantor, Zbyszek Dulinski, Konstantin Pussep, Simon Oechsner, Tobias Hoßfeld, Dirk Staehle, Peter Racz, Fabio Hecht, Marian Callejo. Maximilian Michel

Reviewers: Burkhard Stiller, Nicolas Liebau, Zoran Despotovic

Doc ID: D2.2-v1.7.doc

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.1	5/8/2008	George D. Stamoulis, Ioanna Papafili	Initial version of ToC
V0.2	06/10/2008	George D. Stamoulis, Ioanna Papafili	Update
V0.3	13/10/2008	Ioanna Papafili	IoP & MM bullets
V0.4-5	15/10/2008	Sergios Soursos, Ioanna Papafili	VPN-assisted overlays, Distributed Peer Exchange
V0.6	20/10/2008	Ioanna Papafili	Merge contribution by TiD
V0.7	22/10/2008	Ioanna Papafili	Merge contribution by UniWue
V0.8	22/10/2008	Ioanna Papafili	Merge contribution by AGH, UZH, DoCoMo
V0.81	27/10/2008	Sergios Soursos	Update of ToC, after the WP2 conf call
V0.9	11/11/2008	Sergios Soursos, Ioanna Papafili	Initial merging of contributions from UZH, TUD, AUEB, AGH, UniWue and TID
V1.0	12/11/2008	George D. Stamoulis, Sergios Soursos, Ioanna Papafili	Initial merging of contributions from UZH. Initial reviewing of Chapters 2, 3, 4, 9
V1.1	17/11/2008	George D. Stamoulis, Sergios Soursos, Ioanna Papafili	Merging of input for Sec. 4.3, 4.4 (Maximilian Michel), review of Sec. 2, 7, 10, 11 (Tobias Hossfeld), review of Sec. 9 (Ioanna)
V1.2	22/11/2008	George D. Stamoulis, Sergios Soursos, Ioanna Papafili	Merging of reviews for Sections 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 by AUEB, AGH, DoCoMo, UniWue, TUD, TiD. Addition of Sections 1 and 13.
V1.3	25/11/2008	George D. Stamoulis, Sergios Soursos, Ioanna Papafili	Merging of reviews for Sections 5, 7, 10 by AUEB, DoCoMo, AGH.
V1.4	02/12/2008	George D. Stamoulis, Sergios Soursos, Ioanna Papafili	Final changes to Sections 3.6, 5.2.1.1, 6, 9 (table), 10, 14, ToC
V1.5	04/12/2008	Burkhard Stiller	Review of v1.4
V1.51	11/12/2008	Zoran Despotovic	Review of 1.5
V1.52	16/12/2008	Nicolas Liebau	Review of 1.5
V1.6	23/12/2008	George D. Stamoulis, Sergios Soursos, Ioanna Papafili	Addressing comments of reviewers, sending comments to the authors, merging feedback from the authors

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 project SmoothIT aims at defining, developing, and testing Economic Traffic Management (ETM) mechanisms to optimize the traffic impact of overlay applications on underlay networks in such a way that all network operators, overlay providers, and application users benefit from the approach undertaken; this corresponds to the so-called TripleWin situation. To this end, the WP2 “Theory and Modeling” investigates theoretical foundations of ETM and develops and analyzes new related models. This deliverable presents the work accomplished primarily in the second half of the first year of the project by WP2, building on the work accomplished so far by the project on Self-Organization Mechanisms and their applicability to ETM, on overlay applications and relevant requirements for ETM, and on the SmoothIT architecture. The main content of this deliverable can be summarized as follows:

- An overview of the research literature related to WP2 modeling issues, as well as an assessment of the applicability of certain ideas to SmoothIT.
- The description and justification of a variety of innovative ETM approaches (offering monetary and/or performance-related incentives to the players involved), together with associated components and their required intelligence. The appropriateness of each such approach for video-on-demand (VoD) (which has already been selected by SmoothIT as the application for both the internal and the external trials), and for file-sharing, (which is the fallback solution for the external trial) is also considered.
- A simple yet *complete* SIS-based ETM approach, including an algorithm run by the SIS, which aims to promote locality awareness in an autonomous system in a win-win fashion for both the ISP and the users; the approach is based on BGP information.
- A qualitative evaluation and a detailed classification of all these ETM approaches proposed and a study of their relations, both at the conceptual and the architectural level, having as a yardstick the architecture already documented in SmoothIT’s D3.1 “Economic Traffic Management Systems Architecture Design (Initial Version)”.
- The work accomplished so far by the project on new theoretical models pertaining to the application of ETM on overlay traffic as well as related experimental studies.
- The work accomplished on simulation models and scenarios for validating those ETM approaches proposed, including a reference topology, representative scenarios for experiments, and the components of the simulator developed in SmoothIT, which is based on the abstract simulation framework Protopeer.

Future work will focus on further specification and assessment of the ETM approaches proposed, with respect to their effectiveness, complexity etc. Thus, SmoothIT will provide comparable scenarios and conditions that should apply, in order for each approach to be effective. This will result in a situation that will be beneficial for all players involved, taking into account complex interactions between the overlay application, the charging scheme, and the ETM approach itself. An integral part of the assessment of ETM approaches is the analysis by means of theoretical models, an experimental study by means of simulations and the further investigation of other issues such as interconnection charging the traffic generated by actual swarms. All these studies will have a close interplay and will lead SmoothIT to the specification of the ETM approaches with the highest optimization potential that result in the TripleWin situation targeted at.

2 Introduction-Overview

The project SmoothIT aims at defining, developing, and testing Economic Traffic Management (ETM) mechanisms to optimize the traffic impact of overlay applications on underlay networks in such a way that all network operators, overlay providers, and application users benefit from the approach undertaken; this corresponds to a win-win-win situation, henceforth referred to as TripleWin. To this end, the WP2 “Theory and Modeling” investigates theoretical foundations of ETM and develops and analyzes new related models. This deliverable presents the work accomplished primarily in the second half of the first year of the project by WP2 which already delivered in M6 its first deliverable, namely D.2.1 “Self-Organization Mechanisms for Economic Traffic Management”. [D2.1] provides an overview and classification of Self-Organization Mechanisms (SOMs), an evaluation of the usability of such mechanisms for the applications of interest to SmoothIT, as well as an initial study of how they can be used in Economic Traffic Management (ETM), aiming to offer monetary and/or performance related incentives to the players involved. The present deliverable builds on D2.1, on the theoretical and modeling research undertaken in the course of all active tasks of WP2, and on the work accomplished so far by the project on overlay applications and relevant requirements for ETM (deliverable D1.1 “Requirements, Applications Classes and Traffic Characteristics (Initial Version)” [D1.1], and on the SmoothIT architecture as documented in deliverable D3.1 “Economic Traffic Management System Architecture Design” [D3.1].

The main objectives of this deliverable are:

- To present and justify a variety of innovative ETM approaches (offering monetary and/or performance-related incentives to the players involved), together with associated components and their required intelligence, as well as a classification and qualitative evaluation of all ETM approaches proposed. The appropriateness of each such approach for video-on-demand (VoD) (which has already been selected by SmoothIT as the application for both the internal and the external trials), and for file-sharing, (which is the fallback solution for the external trial) is also considered.
- To present the work accomplished so far by the project, mainly addressing new theoretical models pertaining to the application of ETM on traffic generated by overlays as well as related experimental studies.
- To present the work accomplished on simulation models and scenarios for validating those ETM approaches proposed.

To serve these objectives, the document addresses a variety of subjects and investigates all key relations among them. In particular this has been achieved as follows.

Section 3 contains an overview of the research literature related to WP2 theoretical and modeling issues, as well as an assessment of the applicability of certain ideas with respect to SmoothIT. In particular, the overview covers: (a) research works that comprise analytical models pertaining to cases with information asymmetry, (b) analytical models for BitTorrent and other peer-to-peer systems and their incentives mechanisms, (c) models for interconnection economics, (d) routing and traffic management issues with emphasis on BGP and its impact, (e) incentives mechanisms for peer-to-peer systems, and (f) models for inter-ISP traffic management in the presence of some interconnection agreement.

Section 04 first provides a short review of the structure of the Internet and interconnection agreements between Internet Access Providers and Internet Service Providers (ISP). In particular, peering and transit agreements are presented, together with relevant tariffs for inter-ISP traffic, including an instantiation of the commonly used 95th percentile rule as prominent charging scheme for transit agreements. The sensitivity of this rule to different parameters is investigated subsequently, both experimentally and in terms of an analytical model; this work will be continued in the second year of the project. The consequences of the application of this charging model with respect to ETM are discussed.

The ETM approaches developed are classified into four main categories: (a) ETM approaches that are based on the concept of the SmoothIT Information Service (SIS); (b) ETM approaches based on QoS/QoE-related incentives as well as on mechanisms; (c) an approach employing a new entity in the overlay, namely the ISP-owned peer; and (d) other approaches not falling into any of these three categories. Sections 5 to 8 present the work of SmoothIT on these categories and on the respective approaches.

Section 5 presents those ETM approaches that are based on the concept of the SmoothIT Information Service (SIS) already introduced in deliverable 3.1. The SIS conveys information between the overlay application and the (underlay) network. It is accessed by overlay applications and provided by a network operator in order to achieve ETM of overlay application traffic. Different ETM approaches to be implemented in the SIS are proposed; they are based on promoting locality in order to reduce the interdomain traffic following different strategies. In particular: (a) The *BGP-based Locality Promotion* approach offers to the overlay end users an information service to get a ranked list of peers according to the BGP information, which is usually quite stable; it should be noted that this approach is completely specified and is amenable to implementation. (b) The *Centralized SIS and Dynamic Locality* is related to the previous approach but also takes into account more dynamic information coming from the network status monitoring in order to achieve a reduction of the charge for inter-ISP traffic rather than the reduction of such traffic itself. (c) The *Locality-aware Tit-for-Tat/Unchoking* is based on the selection (with the help of SIS) of local peers among the recently joined ones instead of selecting random peers to optimistically unchoke.

Section 6 focuses on an ETM approach of different spirit, namely the introduction of an *ISP-owned peer (IoP)*. This is an entity that aims at increasing the level of traffic locality within an ISP and at improving the performance enjoyed by users of peer-to-peer applications. The IoP is either a resourceful entity belonging to the ISP itself or is a regular peer that is granted by the ISP with extra resources, *e.g,.* higher downlink/uplink bandwidth, at no extra cost. In both cases, the IoP is assumed to run the overlay protocol, *e.g.,* the BitTorrent, but with certain differences that serve the aforementioned purposes. In principle, the introduction of an IoP is transparent to remaining players. That is, it requires no collaboration between the ISP and either the overlay or content provider, although such a collaboration would improve its effectiveness.

Section 7 presents a set of selected ETM approaches based on incentives as well as on mechanisms for provisioning “explicit” Quality-of-Service and Quality-of-Experience guarantees and/or differentiation, rather than just aiming to improve QoS/QoE. In particular: (a) An ETM approach offering *QoS incentives for service providers and end-users*, which provides guarantees to peer-to-peer overlay applications based on the Control Plane of the NGN equipment. Under the first variation of the approach, the Carrier Class services are provided by the overlay service provider built on agreement with the ISP; the second variation gives the opportunity of QoS guarantees assurance. (b) *Locality-*

based traffic shaping, which would enable the assignment of different classes of upload bandwidth with respect to ISP-internal and remote connections. (c) The *VPN-Assisted Overlays* approach, according to which, VPNs dedicated to specific overlay applications would be established to enable for service differentiation. (d) A *QoE-aware Feedback* mechanism, which aims at predicting and reacting to possible QoE degradation.

Section 8 presents a set of additional ETM approaches with making use of a variety of means and mechanisms that differ from those of the aforementioned ETM approaches. In particular: (a) The *Distributed Exchange of Peer Lists* approach, which assumes that peers share their evaluations of other peers, which are obtained from their past experience with others in the overlay; the ISP can intervene by exchanging information in accordance to its own interests by means of the ISP-owned peer. (b) The *Content Promotion* approach aims is to gather and promote information about the same or similar contents offered in different swarms, or even in different content distribution platforms, so that larger swarms are created and thus resources are exploited more efficiently while locality promotion approaches are more effective. (c) ETM approaches based on overlay enhancements, which include: the *Private and Shared History overlay* incentives' mechanism; the *Tree-based Dynamic Locality* approach, which proposes an interface between the SIS and the Tracker in order to allow the Tracker to get topology information from the ISP thus exploiting the tree structure of the Internet in order to deal more effectively with information replication; and *Locality-aware Overlay Caching*, whereby, a peer is offered the incentive to keeps on offering a file following completion of its downloading, thus leading to the availability of close (mainly with respect to locality) replicas of a file.

Section 9 presents a detailed classification of all these ETM approaches proposed and a study of their relations, both at the conceptual and the architectural level, having as a yardstick the architecture already documented in SmoothIT's D3.1. Three major properties characterize an approach, and determine how it is classified: (a) whether it defines an *Information Exchange* mechanism, (b) whether it includes certain *Traffic Management* techniques, and (c) whether it introduces a new *Architectural Component or Interface* or a *new/enhanced Overlay Entity*. The majority of these ETM approaches proposed are based on the SIS architecture, with some of them providing basic functionalities (such as locality promotion) and others offering extensions and enhancements that strengthen the notion of ETM; the rest of these approaches are decentralized and include hybrid ones. Overall, this section provides a roadmap for the future development of ETM approaches. This study has also led to the identification of the most prominent combinations, namely: a SIS-based locality promotion (or an enhanced version thereof), possibly combined with a self-organization mechanism in the overlay and/or the IoP for achieving more active intervention of the ISP in ETM, and with a QoS/QoE-related mechanism for enforcing performance objectives. Some other important considerations related to the nature of the peer-to-peer video streaming applications as well as the timescales in which specific actions should be made are also discussed.

Section 10 presents an innovative Markov model for modelling BitTorrent swarms. The objective of this model is the analysis and evaluation of optimization approaches such as an insertion of ISP-owned peers in a BitTorrent-like network. Using probabilistic analysis, the model leads to the estimation of the distribution of the number of chunks downloaded by each given peer and other performance measures, such as the upper tail of the distribution of the time required for a peer to complete downloading a file. Since the state-space of the system is prohibitive, the model resorts to certain approximations and exploits symmetry, thus ending being numerically tractable. This work will be continued in the second year of the project, with the evaluation of the accuracy of the model and the derivation of performance related conclusions and guidelines for ETM.

Section 11 presents an investigation by means of measurements in the Internet of the sizes of actual BitTorrent swarms. The main conclusion is that a Pareto-principle governs the total peer distribution: a large share of the peers can be found in a small number of swarms, while there are many small swarms. Also, there is a strong correlation between the number of seeders and leechers. The temporal evolution of population sizes within swarms is studied too; relevant conclusions can be used for developing future models for swarm dynamics. This analysis has a direct influence on the applicability and efficiency of ETM mechanisms, because it reveals that ETM should work efficiently with swarms of all sizes, possibly applying a different approach. There are a critical number of peers per swarm required within an ISP's network in order to successfully use locality promotion and achieve a substantial reduction in inter-domain traffic without simultaneously decreasing the overlay's performance. Also an ISP-owned Peer (IoP) provides a solution for smaller swarm sizes, too. However, the largest impact can be obtained by IoPs participating in larger swarms, where more popular content is cached.

Section 12 presents the work accomplished on simulation models for validating ETM, including a reference topology, representative scenarios for experiments, and the components of the simulator. The simulator developed in SmoothIT is based on the abstract simulation framework Protopeer. This framework was selected, since it offers a built-in logical distinction between the network model and the overlay logic. The latter is modeled based on the Tribler peer-to-peer Video-on-Demand. To be able to simulate larger swarms, the network model will include a traffic flow model (including delay and bandwidth) rather than packet level details. The first ETM approach that will be implemented in the simulator is the concept of a central SIS (SmoothIT Information Service) server that is contacted by peers in its own Autonomous System to receive underlay information about their neighbors. This SIS server will be placed in a reference topology to be able to compare other approaches under the same network circumstances. This reference topology is rich enough to study all important effects applicable to an ISP employing ETM. In particular, it includes a multi-homed AS that will be in the focus of the evaluations, which can forwarding traffic via Tier 1 ISPs (transit agreements), while also being connected with another Tier 2 AS (peering agreement). Also presented is a list of currently configurable parameters for these simulations.

Finally, Section 13 presents the main conclusions of this deliverable and of the work to be undertaken in the future. In particular, Note the ETM approaches presented this deliverable, except for that in Subsection 5.1, are mostly at the level of ideas and proposals, which have been identified and checked in terms of plausibility to be applicable. Future work will focus on further specification and assessment of the ETM approaches proposed, *e.g.*, with respect to its effectiveness or complexity. in order for SmoothIT to identify those ones with the highest optimization potential. To this end, it is necessary for SmoothIT to provide comparable scenarios and conditions that should apply, in order for each of these approaches to be effective. This will result in a situation, which will be beneficial for all players involved, taking into account complex interactions between the overlay application, the charging scheme, and the ETM approach itself. An integral part of the assessment of ETM approaches is the analysis by means of theoretical models and an experimental study by means of simulations. However, future work will also span other directions, namely (a) that of interconnection charging, with the focus on conditions and parameters that affect mostly the 95th percentile rule, and (b) investigations on actual swarms, with a focus on actual traffic they generate. All these studies will have a close interplay, in order to lead the project to the specification of efficient ETM approaches that result in the TripleWin situation targeted at. In several ETM approaches, such as those involving the SIS, the ISP is heavily involved and/or is the main enabler of the approach. Thus, it is conceivable that the ISP can shape the effect of the approach to his own benefit; e.g. attain the maximum possible reduction of inter-domain traffic. However, this should be done subject to the constraint that TripleWin is maintained and the user enjoys performance that is still better than that without ETM, in terms of e.g. download completion times, content availability etc. This does not imply that the ISP can "overdo" this. Indeed, in a competitive environment, users would choose to subscribe to the ISP that offers the best performance. This fact should motivate each ISP to employ ETM so that his users benefit considerably, rather than just being slightly better off.

As already noted the present deliverable builds on D1.1. Indeed, in that deliverable, Video-on-Demand (VoD) was selected as the application for both the internal and the external trials, with file-sharing being a fallback solution for the external trial. These two application classes have important differences, which affect the effectiveness of these ETM approaches. Therefore, it is expected that choices and specifications of the most appropriate ETM approach for each application will be different.

These requirements of aforementioned applications are taken into account throughout the work presented in this deliverable, together with properties of traffic generated by such applications and of the charging scheme applied to the ISP traffic. In fact, the adoption of Video-on-Demand provides SmoothIT with the opportunity for innovative work, since other related research initiatives mostly deal with file sharing. Indeed, the traffic generated by an overlay greatly depends on the class of application it belongs to. This is not only due to the type of application, but also due to the number of users it attracts at a given point in time. In an overlay used for live streaming, all users are online and requesting the same content at the same time, while a VoD overlay may distribute its content over a longer time, with less users online at the same time, but all in all a larger number of viewers.

Therefore, SmoothIT concentrates on two classes of applications, namely peer-to-peer file sharing and VoD, both in a BitTorrent-like system (for more details consult D1.1). One of the major mechanisms influencing the traffic behavior in these two overlay classes are the chunk selection and the peer selection mechanisms, cf. D2.1. To illustrate key effects of these mechanisms, let us first note that file-sharing employs a tit-for-tat mechanism (cf. Subsection 3.5.1), while a promising peer selection mechanism for VoD follows the give-to-get scheme (cf. Subsection 3.5.2). A consequence of these different peer selection algorithms is that different ETM approaches do not achieve the same effect. *E.g.*, for a tit-for-tat strategy, traffic locality will lower both incoming and outgoing traffic. If the charging scheme works on the difference between those volumes of these two traffic streams, the monetary benefit for the ISP charge may be minimal. Since the give-to-get scheme is not symmetric, the effect of the same ETM approach in a VoD system can be expected to be much larger.

To summarize, all application requirements, traffic characteristics, charging scheme properties, and objectives of an ETM approach are interdependent. SmoothIT takes all of them together and in an integrated approach into account, in order to develop effective and practically applicable ETM approaches, suitable for real-world cases and tomorrow's ISP traffic to come.

3 Overview of Related Literature

In this section we overview literature related to WP2 theoretical and modeling issues, as well as an assessment of the applicability of certain ideas with respect to SmoothIT. In particular, the overview covers a wide variety of topics, each of which is related to at least one of the modeling problems addressed, or to the specification and assessment of some of the ETM approaches to follow. These topics are:

(a) research works that comprise analytical models pertaining to cases with information asymmetry,

(b) analytical models for BitTorrent and other peer-to-peer systems and their incentives mechanisms,

(c) models for interconnection economics,

(d) routing and traffic management issues with emphasis on BGP and its impact,

(e) incentives mechanisms for peer-to-peer systems, and

(f) models for management of inter-ISP traffic in the presence of some interconnection agreement.

In the end of this Section, we provide a conclusion relating all works reviewed to the SmoothIT context, the associated models, and the ETM approaches to be studied.

3.1 Traffic Engineering vs. Overlay Routing

In this subsection we overview two articles that address the interaction between an overlay network and the underlying physical network [LZG+04], [ZLG+05].

The authors consider application overlay networks (rather than search overlays), which are logical networks on top of physical networks. Two overlay nodes, *e.g.* two BitTorrent peers, are connected by a logical link. A logical link corresponds to a physical path which is a set of ordered physical links on the physical network. There are two processes ran in this context, each by a different entity; namely: (a) Overlay Routing (OR), which allocates overlay demands on logical links based on the current logical link delays employing an overlay routing algorithm. The delays on logical links are computed by the delays on the underlying physical path. (b) Traffic Engineering (TE), which allocates traffic demands on physical links based on the current physical link delays. The overlay flow on a logical link is interpreted by TE as traffic demand between the two overlay nodes. Thus, while OR takes care only of the overlay users performance, the TE cares about the overall network-wide efficiency, both overlay and extra underlay traffic. The interaction between TE and OR is conceptually depicted by the authors of [ZLG+05] in Figure 3.1.

This interaction is modeled in the articles as a non-cooperative non-zero sum two-player game. The game is assumed as non-cooperative since the two players, TE and OR, have different optimization objectives, *e.g.,* overlay vs. overall optimization respectively, and furthermore is non-zero sum since if one player wins, the other does not lose, *e.g.,* the game can reach a win-win situation.

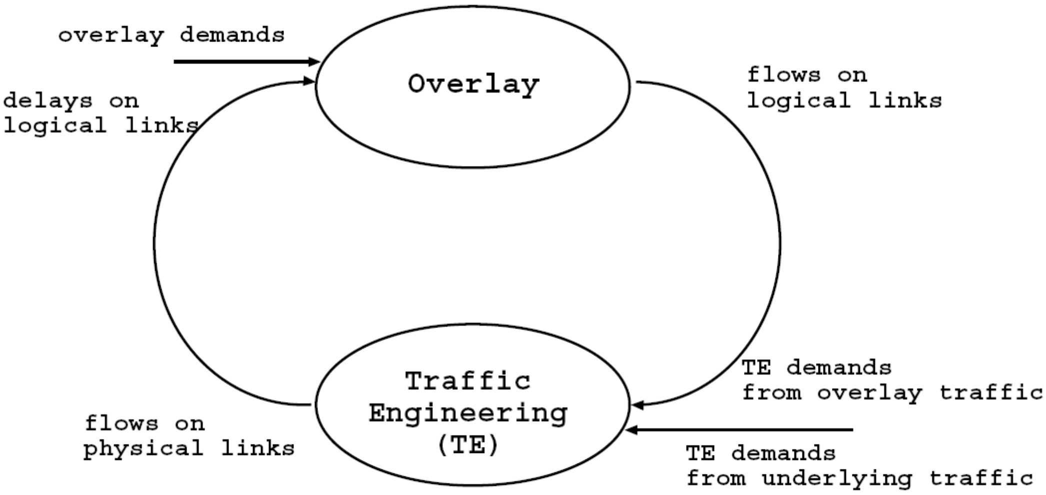


Figure 3.1: Interaction between TE and OR [ZLG+05].

The OR allocates traffic demand between given overlay nodes pairs onto different logical paths. The authors assume that this is done by a central overlay entity, which aims to meet an optimization criterion. The overlay allocation amounts to the strategy of the OR. Overlay traffic on a logical link is physically routed from a specific source-node to a specific destination-node by TE. At the same time, TE also accounts for traffic demands from normal underlay users, playing its own strategy. In the physical network, TE allocates all physical traffic demands to all of the physical links. TE and OR are coupled through the mapping the logical level links to the physical level paths. Congestion delay is adopted as network performance metric to calculate link cost.

Ideally, if the centralized overlay routing entity knew exactly the physical network topology, traffic demand and TE's routing, it would be able to compute its optimal strategy. However, due to the fact that the OR may not have this information, it also won't be able to compute its optimal strategy. On the other hand, since TE knows the physical network's topology and all link capacities, and if TE is assumed to be able to compute its demand matrices accurately, then TE can always compute its optimal strategy. This phenomenon is also known as Information Asymmetry.

The authors first assume that the overlay has the necessary information to compute its optimal strategy and model the interaction as a Nash routing game. This Nash routing game is a discrete time model in which one player completes its optimization before the other player starts. In the interaction process, or best-reply dynamics, each player adjusts its response optimally based on the other player's decisions during the previous round and so on. Mathematical analysis shows that even for a simple topology the Nash game converges to an inefficient Nash Equilibrium Point (NEP) for the overlay. That is, the overlay's performance degrades as the game proceeds. Thus, the best-reply strategy is not the best strategy for the overlay to use when interacting with TE. This performance degradation sounds counter-intuitive since the overlay plays its optimal strategy, but it is rather expected due to the misalignment of the two players' objectives and the fact that each player intervenes between two consecutive actions of the other. On the other hand, OR never improves TE's performance, in the sense that any improvements achieved in the overlay's traffic come at the price of degrading the performance of the underlay's traffic (namely, the non-overlay traffic).

Furthermore, the authors assume that the overlay knows TE's routing algorithm. The interaction here is modeled as a static Stackelberg routing game where the OR (leader) chooses his optimal strategy X*, while the TE (follower) reacts to OR's decision by selecting a strategy Y(X*) – assumed to be unique – that minimizes its cost function, in full knowledge of the OR's decision. Thus the follower's decision depends on the leader's decision, and the leader is totally aware of that. The game is considered as static since the OR does an off-line computation to find its best strategy. Based on the traffic demand matrix, TE solves a Linear Programming problem to get an allocation of traffic on physical links, while the solution to OR's problem comes by using Gradient Projection Search (GPS). Due to the fact that GPS is only a local search heuristic, it can only find a local optimal solution in the neighborhood of the starting point. In order to search for the global optimum, the gradient projection method has to be started many times randomly. The overlay cost with Stackelberg solution is expected to be no worse than the overlay cost at any NEP, since the Stackelberg equilibrium prescribes an optimal strategy for the OR (leader) if the TE (follower) reacts by playing optimally, whenever the leader announces his moves first.

Additionally, a routing game with incomplete information is formulated, and it is again analysis for its Nash equilibrium properties. In particular, the overlay is assumed to be able to measure delays in logical links using packet probing. Based on the probing frequency, two types of overlays are considered: one-step overlays with limited information, and incremental overlays with limited information. Comparing the two overlay types, the trade-off lies between the overhead and the performance improvement. In particular, for the one-step overlay the overhead is small but the performance loss might be very high. On the other hand, the incremental overlay is more efficient in terms of performance improvement; however the measurement overhead might increase dramatically.

Both articles come to similar conclusions. Generally, the selfish behavior of the OR degrades the performance of non-overlay users and the underlay network as a whole. The TE's performance is never improved by OR's decisions. Respectively, OR's cost increases even if OR responses with its optimal strategy to TE's routing. The analytical results of these articles are not directly amenable to SmoothIT, because the respective model does not deal directly with overlay applications that involve multiple alternative sources the selection among which greatly affects traffic patterns. Nevertheless, the conclusions above do advocate for the necessity of cross-layer optimization, such as that targeted by the ETM approaches of SmoothIT.

3.2 Models on Peer-to-Peer File-sharing Performance Evaluation

Although there is a very extensive literature on peer-to-peer performance evaluation, most of the relevant works are exclusively based on analysis of measurements from actual systems and/or simulations. In this subsection, we provide an overview of certain articles that include models for the performance evaluation of peer-to-peer systems with emphasis on file-sharing and particularly BitTorrent, which are the subject covered by most of the related literature.

In [GFJ+03], Zihui et al. present a 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.

In [KR06], Kumar and Ross analyze the minimum distribution time for a file in a system with seeds and lechers. 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. These authors also compare the corresponding time for a client-server model, and show that peer-to-peer is indeed superior.

The work of Fan et al. in [FCL06] deals with the main 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, which is investigated analytically in the paper. Fairness amounts to service discrimination; that is, a peer contributing more to the system receives better service, which when applicable implies incentive compatibility. In particular, the authors employ a model according to which peers are classified into n different classes, where the i-th one is characterized by a maximum upload rate U_i, and a maximum download rate D_i, where $D_i > U_i$. Classes are ranked in terms of their maximum upload rates, with U_1 being the maximum one corresponding to fat peers. The authors then define a family of rate allocation problems, in each of which they derive the upload and the download bandwidth actually used by the peers of each class. Each such problem has a different optimization objective. One of the main conclusions derived from these problems is that fairness and performance are not compatible. This is seen by comparing the properties of the optimal solutions of two rate allocation problems. First, for the allocation optimizing the mean download time, the fairness index is not optimized; moreover, in order to achieve this objective fat peers do not exhaust their maximum download rate, as opposed to the rest of the peers, while in some cases they may have to download at a rate lower than the one of some other classes. On the contrary, under the allocation optimizing a fairness index, each peer has a download and an upload rate both of which are equal to U_i. This allocation is fair in the sense that the peer contributes to the system as much as it consumes, at the expense of a sub-optimal mean download time. It should be noted however that the aforementioned optimal rate allocations can only be centrally imposed in an environment with collaborative peers. In reality though, each peer acts selfishly in terms of whom to offer upload to. Thus, the authors also study the system under this assumption; consider two alternative policies namely selective and random uploading. When peers employ selective uploading, then it is proven that the situation where each peer selects a certain number of other peers with similar upload bandwidth to itself (essentially, tit-for-tat) constitutes a Nash equilibrium. On the other hand, the random selection policy (essentially, optimistic unchoking) results in max-min fairness of the download rates. Finally, the authors verify by means of simulations that their model has good accuracy. They also study the effect of certain design parameters of the system, namely the number of peers unchoked selectively and randomly. It turns out that if the number of random unchokes is 1, then the fairness index is low, but improves considerably when it is raised to 2 or 3.

In [YV06], 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 once A downloads the content from some other peer B, which in turn may have got it from some other peer. The authors thus develop a simple deterministic model that shows the effect of this feature in the transient case, where 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. This result is again derived by means of a deterministic model. Note that these models apply to BitTorrent, although they may also apply for other unstructured systems too. Also note that 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 upload schedules both for the case of a central server and for the case of a decentralized system with peers having equal capacities. The authors 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. The authors also show that when a peer can serve multiple others in parallel, then the exponentially growth of E[N(t)] is slowed down if all peers can be taken as cooperative, but it is speeded up in an uncooperative environment where not all peers continue to serve others. All of the above conclusions show the scalability properties of the peer-to-peer service paradigm. Furthermore, the authors of [YV06] develop a model for the stationary state of the system, which is described by a pair (x,y); x is the number of peers whose requests for the file is currently in progress (leechers), while y is the number of seeds, namely peers that have already completed their download but still remain in the system. New requests are assumed to arrive according to a Poisson process with rate λ. The total service rate equals μ(ηx + y), since except for seeds downloaders can partly serve others too, while seeds also leave the system at a rate γ. The authors investigate this model and the impact of the various parameters numerically. The authors also provide measurements from BitTorrent traces, verifying the conclusions on the transient state; *e.g.,* the initially exponential growth of the service capacity. However, they also observe that the increase later slows down; that is, the system does not exploit all the potential upload capacity. This may be attributed to tit-for-tat. Hence, the authors define certain notions of fairness; that is, expressions giving the resources received by each peer in a system in terms of the resources offered by all of its members. They also derive conditions under which such a fair allocation can be attained. It turns out that these conditions are rather restrictive, and thus can hardly hold in an actual peer-to-peer system; *e.g.,* it would be required that the upload and the download throughput between any two peers be balanced. The authors finally discuss how the right incentives can be provided in practice. In particular, they propose that peers initially entering the system are given priority, while later it is required that they also contribute to the system proportionally to what they receive.

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 [YV06], which was initially published in [VY03]. In fact, the model of this article comprises the same parameters as [YV06]. Yet, *x(t)* and *y(t)* (namely, the number of downloaders and seeders at time t) are taken from first order differential equations, which of course involve the various rate parameters λ, μ, γ. Setting the derivatives of *x(t)* and *y(t)* to 0, the authors obtain expressions for the mean numbers of downloaders and seeders present in the system. Then, they employ Little's law in order to derive the mean time T for downloading a file. The dependence of the *T* on the various parameters is that intuitively expected except for the fact that it is independent of λ, namely the arrival rate of new downloaders. This shows that BitTorrent is scalable. The authors also present a probabilistic model for the evaluation of the parameter that η expressed the effectiveness of BitTorrent in the sense of the degree of the contribution 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 of chunks possessed by a peer are similar to those of the model developed in Section 10 of this deliverable. 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. Finally, the authors complement the analysis of the deterministic model by: a) employing an eigenvalue analysis in order to extract the conditions under which the fluid model for *x(t)* and *y(t)* is locally stable around their respective mean values, and b) by estimating approximately the deviations of *x(t)* and *y(t)* from their respective mean values as Gaussian random variables whose variances and covariance are computed by solving a Lyapunov matrix equation; these results are applicable for large arrival rate λ. Furthermore, the authors of [QS04] define an incentive mechanism that is reminiscent of the unchoking algorithm of BitTorrent, but for tractability purposes it is based on global information. In particular, at each round only one peer selects which other peer to serve based on their ranking of the upload bandwidth. Then, a game is defined, whereby each peer has to decide on his upload bandwidth and in particular whether he will exhaust the corresponding line rate p_i will contribute less. It is shown that there does not always exist a Nash equilibrium in this game peers have different line rates. How if peers can be classified into large enough groups, each of which is characterized by a particular line rate, then there does exist a Nash equilibrium. Moreover, in this equilibrium each peer contributes its maximum possible upload rate, namely p_i. The authors also briefly address the issue of optimistic unchoking and how this may lead to free-riding yet of a limited extent. Finally, they provide simulation results in order to validate their model. Indeed, the model captures with good accuracy the most important effects and dependencies in BitTorrent. Most remarkably, it is confirmed that the time to download a file is indeed independent of the arrival rate λ, thus implying that even very popular files can be downloaded at roughly the same time as popular ones.

It is also worth noting that the model for computation of the parameter η of [QS04] is also employed by Tawari and Kleinrock in [TK07] for the analysis of the effectiveness of the peer-to-peer paradigm in serving live video streaming. In particular, these authors conclude that $\eta \approx 1\text{-}\ S/(\tau R)$, where now η expresses the percentage of the upload capacity of the peers that can be utilized, S is the fragment (i.e. chunk) size, τ is the playback delay introduced to allow fragment exchange, and R is the streaming rate. For each given target value of η, this expression leads to the corresponding value of τ and finally to the required buffer for storing the fragments, namely τR. The authors also show that a peer size group of 15-20 is adequate for an effective peer-to-peer live streaming application, in the sense that with such a group size is sufficient to give rise to the upload capacity benefits characterizing the peer-to-peer paradigm.

In [LHW+07], Leibnitz et al. present a fluid flow model for evaluating the performance, in particular reliability and efficiency, of content distribution services that can be realized by traditional client/server (C/S) architectures or peer-to-peer networks. Since in peer-to-peer systems the load is distributed among all sharing peers, the risk of overloading servers with requests is reduced, especially in the presence of flash crowd arrivals. Although this improves the efficiency in general, peer-to-peer systems face new challenges. Since the shared file is no longer at a single trusted server location, peers may offer a corrupted version of a file or parts of it. They are called malicious or fake peers and the phenomenon is referred to as pollution. When the number of fake peers is large, the dissemination of the file may be severely disrupted. All of this leads to a trade-off consideration between a) high reliability (and inherently costs) at the risk of overloaded servers and b) good scalability where the received data may be corrupt. It is assumed that the user is willing to wait only for a limited time until the download completes. If the downloading process exceeds a patience threshold, the user will abort his attempt. In the C/S case, the server bandwidth is the limiting factor that determines the download time, while in the peer-to-peer system, the download from fake peers prolongs the overall download completion time. The model of [LHW+07] for the peer-to-peer network is described on the example of the eDonkey network. However, it can also be applied to BitTorrent swarms when adapting the size of blocks and chunks accordingly. In the model, the multi-source download mechanism, also known as swarming mechanism, is taken into account, while incentive mechanisms like tit-for-tat or credit point systems are neglected. These incentive mechanisms are approximated by the common max-min fair share assumption, which means that the available upload bandwidth is fairly shared among all downloading peers. The model of is based on the epidemic diffusion of diseases and is characterized by a differential equation system describing the transitions between each of the states a peer traverses. In the following, the states D_i, F_i, A, S, L of the model are briefly reviewed. The current state of a peer is characterized by the amount of downloaded blocks (D_i or F_i) of a certain chunk and whether the peer has downloaded a block from a fake peer (F_i) or not (D_i). If one of the downloaded blocks of a chunk is corrupted, the peer will first notice this after computing the checksum of the chunk and then has to re-download all blocks of this chunk. Due to the peer's patience, she will retry the download attempt (i.e. switching to D_0) or abort (A) with probability p_A. If the peer successfully downloads a file, she shares it (S) with a certain probability p_S for some time or immediately leaves the system (L) with probability $1-p_S$ due to her selfish user behavior. The arrival of new requesting peers or new sharing peers and seeders is also expressed by appropriate differential equations. The ordinary differential equation system is then numerically solved, *e.g.,* by using the Runge-Kutta approach. As a result, the transient behavior of the population sizes, as well as the transition rates between different states is obtained. In particular, this allows deriving the download time and the amount of cancelled downloads due to fake peers. The accuracy of the model is validated by comparing the analytical approach with a flow-level simulation of the peer-to-peer system. The numerical results in the paper show that peer-to-peer systems can be easily made inoperable when many fake sources exist. If the initial number of sources is small there is a risk of these peers leaving the system which would make the network lose content due to churn. For this reason, it is important that incentives are being provided to peers to increase the willingness to share the data. Enhanced error detection mechanisms must be provided to reduce the number of retransmission in case of errors. This could be done in combination with a caching peer which acts like a server but whose content is being determined by the requests of the peers.

The analytical models for BitTorrent and other peer-to-peer systems overviewed above do offer interesting results and some modeling techniques that can be employed in SmoothIT. Most notably, the fluid model of [QS04] provides a simple tool for analysis of BitTorrent that is already used in the literature. This model can be employed by SmoothIT for the analysis of some of the ETM approaches, introducing appropriate modifications.

3.3 Models for Interconnection Economics

In [SS06], Shakkotai and Srikant formulate the price games that occur between the ISPs in the different tiers of the Internet hierarchy. Due to this hierarchy, the business relationships between the ISPs vary, resulting in different types of interconnections like peering and transit agreements, as well as bilateral settlements. In this sense, the authors examine three cases: a) the interaction between the Local ISPs, i.e. Tier 3 ISPs, b) the interaction between Local and Transit ISPs, i.e. the Tier 1, 2 and 3 ISPs and c) the private exchanges that may occur between two Tier 2 ISPs or between two Tier 3 ISPs. For all these cases, using simple models, the authors describe the price games between the stakeholders, find the profit maximizing prices and the Nash equilibrium strategies. To do so, the model captures all the prices and costs inferred by every stakeholder, depending on his position in the hierarchy.

For the first case of interaction between the local ISPs, assuming that the ISPs cooperate, the profit maximizing prices for the single step game are computed. The strategy profile in this case is denoted as $s_{cooperate}$. Additionally, since cooperation is not always the case, a threat strategy Nash equilibrium for a repeated game is computed that is proven to achieve the same cooperative maximum. The strategy profile in this case is denoted as s_{threat}. Finally, the authors show that the strategy profile "Play $s_{cooperate}$ until any player deviates and then play s_{threat} for ever" is a sub-game perfect Nash equilibrium for the repeated game, under a condition concerning the value of the payoff discount factor.

For the case of interaction between the local and transit ISPs, the authors define the respective charges and costs for the local ISPs, the Tier 2 ISPs and the Tier 1 ISP and they formulate the profit function of the local ISP. The interaction of ISPs is formulated as a single step Stackelberg game, where players play in sequence and each player knows that the next player will optimize his play based on what he does currently. The (non unique) solution of such a game is shown and is denoted by $s_{stackelberg}$. The main assumption in the setup of this game is that all local ISPs are considered as a whole. In order to answer to the question of what prevents a member for the group to charge less than the optimal, thus getting all local customers, the authors describe a solution in order to ensure cooperation. The strategy profile in this case will be "Play the $s_{cooperate}$ for the intra-regional traffic and $s_{stackelberg}$ for inter-regional traffic until a group member deviates and then play the single step Nash equilibrium prices s_{threat} for ever". Finally, the authors comment on the case there are multiple transit providers (either Tier 2 or Tier 1) and how this affects the transit costs.

In the case of private exchanges, the authors examine when it is beneficial for ISPs of the same level in the hierarchy to peer and bypass the transit ISPs, therefore decreasing their costs. Peering is not the only case it is examined, since other types of bilateral agreements are also probable and still preserve the property of cost reduction.

While interesting, the aforementioned modeling work is not directly applicable to SmoothIT.

3.4 Routing and Traffic Management

For the overlay applications considered by SmoothIT, there are in general multiple alternative sources for the same content. Thus, a peer has to select one among these sources each time, which overall greatly affects traffic patterns. Therefore, routing can serve as a key mechanism for ETM, if the aforementioned selections are influenced by routing procedures and/or related information. In this subsection, we overview BGP, which is the de facto standard protocol for inter-domain routing, and it is employed in the ETM approach of Subsection 5.1. This performs ranking of candidate peers on the basis of information provided by BGP. We also overview the TEQUILA framework for Traffic

Engineering and Management aiming at QoS provision, since later in Section 07 we present certain QoS/QoE aware ETM approaches.

3.4.1 Border Gateway Protocol (BGP) Overview

BGP belongs to the class of Exterior Gateway Protocols (EGP), these protocols are necessary for the exchange of routing information between Autonomous Systems (AS). The exterior gateway protocols are optimized for the exchange and manipulation of very large sets of prefixes, on the contrary the interior gateway protocols (IGP) deal with small number of prefixes. A number of Requests for Comment (RFCs) define the current BGP specification [BGP1-BGP11]. BGP is the only EGP, which is commonly used in the Internet, today. The current version of BGP is version 4 [BGP1, BGP2]. All networks connected to different providers (upstream providers) needs to use BGP in order to provide any dynamic resilience to their network.

BGP runs over TCP (Transmission Control Protocol) over port 179. Each BGP router establishes TCP sessions with other BGP routers. They compose peering system over which all BGP messages are exchanged.

Each AS in the Internet is recognized by its globally unique identifier - AS number, it is16 bit integer (0 to 65535, 0 reserved, 64512 to 65535 private – not announced in the Internet). BGP is a path vector routing protocol. It possesses significant similarities to the distance vector routing protocol. The path to the destination is described as a list of these AS numbers representing each of the AS back to the originating AS in the order they are traversed. Every time a prefix is advertised to another autonomous system, the router adds its own AS number to this list, the AS_PATH. Thus the AS_PATH is created as the announcement of the prefix passes from the destination towards the source. The AS_PATH is used to prevent from routing loops. Before announcing a router checks if its own AS number is in the AS_PATH, if it is, the related prefix can create routing loop, so the prefix should be removed.

Although BGP is fundamentally designed to distribute routing information between autonomous systems, it is also used to communicate the routing information about external networks to the routers inside the autonomous system. It could be used IGP for proliferation of external networks prefixes inside the AS, but the huge number of such prefixes can be problem for IGP. For the purposes of communication between autonomous systems the Exterior BGP is used and for inside AS communication the Interior BGP is adequate.

3.4.1.1 Policy-based Routing

BGP protocol widely supports mechanism for policy-based routing [BGP2]. The present Internet is seen as ISP grid net. Large national ISPs, smaller regional ISPs, and even tiny local ISPs make up the grid net. The ISPs networks are interconnected. The smallest ISP can link to another ISP and thus allow their users to participate in the global, public Internet. Linking between ISPs is governed by a series of agreements known as peering agreements. Some of the ISPs are peers, others are in the relation provider – customer. These agreements and internal ISP policy related to the traffic requirements are basis for the ISP routing policy.

An AS forms a group of IP networks sharing a unified routing policy framework. A routing policy framework is a series of rules and guidelines used by ISP to apply the actual routing policy in configuration on the routers. Coordination of routing policy frameworks and routing polices between ASs is necessary, especially if the ASs belongs to the different ISPs. On routers a routing policy takes a form of commands that allow advertising and receiving approved prefixes. This way there is obtained some schemas for packet transfer from some sources to specific destinations.

In the SmoothIT project there are suggested some architectures in which routers are involved in the ETM procedures. Routers in ETM mechanism have dual meaning, they are source of information about the network behavior and they are points in the network where the ETM can influence on the peer-to-peer traffic. Usually influence mechanisms are related to shaping and routing. On routers peer-to-peer traffic can be identified and procedures in accordance with ETM requirements can be undertaken. The ETM mechanism can deliver parameters, which can be applied in routing, polices. These parameters can exemplify QoS, bandwidth limits and routing preferences (specifically BGP attributes, which can have local ISP meaning and also global – inter provider meaning).

3.4.1.2 EBGP and IBGP

When BGP routers (BGP speakers) are in different ASes, the routers use Exterior BGP (EBGP). EBGP sessions can be distinguished in following way:

Transit providers: a session over which all Internet routes are learned, usually treated as the route of last resort, since an ISP will almost inevitably be paying for traffic going over the link to the transit provider.

Peers: a session over which two ISPs exchange their own routes and those of their customers at no cost to each other. These routes are preferred over those learned from transit providers.

Customers: a session to a customer, which pays ISP for carrying its traffic. These routes, learned from customer, are preferred over other because they generate income.

EBGP peers have to be one hop away, i.e. directly connected. When BGP peers are within the same AS, the routers use Interior BGP. IBGP sessions are usually only required when AS is multi-homed and has multiple links to others ASes (links can be to the same AS or to different ASes. When two BGP neighbors (EBGP or IBGP) first see each other, they exchange entire routing tables, later they exchange only partial table information when they notify routing changes.

Before running IBGP, some IGP (interior gateway protocol) has to be started. TCP sessions between IBGP routers are established according to the information from IGP. IBGP routers have to form full mesh logical topology (TCP connections). Fulfillment of this requirement allows reaching all destinations.

3.4.1.3 Other Types of BGP

One BGP variation is Multiprotocol BGP (MBGP or MP-BGP) [BGP9]. It supports IP multicast routes and routing information. It is also very useful in carrying information for IP-based VPNs. Also, Multi-hop BGP, allows to connect EBGP peer in another AS that is more than one hop away. Finally, Confederation BGP is a variation of EBGP; it is used inside ASes for scaling purposes [BGP11]. It works on the links connecting sub-autonomous systems (confederations).

3.4.1.4 BGP Attributes

ORIGIN: reflects where from BGP first obtained knowledge of the route.

AS_PATH: is a sequence of AS numbers that lead to the originating AS for the network routing information. This attribute can be important parameter for the *SmoothIT* project; it indicates how many ASes should be traversed on the way to the peering party. It can be used for evaluation if existing other paths are more economical choice or more convenient. This attribute can be used for ranking peers.

LOCAL_PREF: this the local preference of network routing information relative to other routes learned by IBGP within an AS, it is not used by EBGP. In the SmoothIT project this attribute can be used by an ISP in order to select among a few border routers, which router will be responsible to transfer traffic to other ISPs. A particular ISP can apply LOCAL_PREF in routing polices to send the peer-to-peer traffic to ISP which supports ETM mechanisms. This attribute has meaning only on the level of a particular AS, this is not any kind of a global metric. It can be used for ranking peers.

MULTI_EXIT_DISC (MED): is used in order to influence by one AS to another when it chooses among multiple exit points (border routers) that links to the AS. Manipulating MEDs is one of the most common ways that one ISP tries to make another ISP use the links it wants between the ISPs. The influence is limited only to the neighboring AS, because there is no knowledge about the links further then MED generating AS. In the SmoothIT this parameter can be used to send the traffic to a better path from the neighboring AS point of view. If neighboring ASes use ETM mechanisms they can represent some ETM parameters in the form of BGP attributes and straight applied in routing polices on routers. It can be also taken into account as a parameter in the algorithm for ranking peers.

ATOMIC_AGGREGATE and AGGREGATOR: both are used when routing information is aggregated for BGP.

COMMUNITY: this attribute are used to create sets of routes, it helps in applying specific polices for particular sets of routes [BGP4]. In the SmoothIT project this attribute can be used for grouping routers in sets, which cooperate in order to achieve some goals related to the traffic transit. An ISP can choose some routers which are devoted for transit peer-to-peer traffic between ISPs, other group can be engaged in peer-to-peer traffic limited to the ISP domain and others groups for other purposes. Having given the community attribute, we can apply on the routers specific routing polices to the members of the community.

ORIGINATOR_ID and CLUSTER_LIST: these two attributes are used in the scaling method called route reflector used by BGP. It helps preventing from routing loops [BGP3].

3.4.1.5 BGP Scaling

There are two mechanisms that help to solve the problem of maintaining the full mesh of routers. One is a route reflector mechanism [BGP3], which allows network operators to configure only partial mesh of routers while maintaining full connectivity. In AS there are special BGP routers called route reflectors. The route reflectors are permitted to announce routes learned from one IBGP neighbor to another IBGP neighbor, such procedures are forbidden in basic IBGP. In order to avoid routing loops ORIGINATOR and CLUSTER_ID attributes are used. All route reflectors must form full mesh topology. The cluster of routers is related to the route reflector.

The rote reflector can be fully meshed with all clients in the cluster or can reflect announcement from one client to another in the cluster. When a cluster grows, it is possible to form a route reflector hierarchy (clusters within clusters).

BGP confederations are another method for reducing the full mesh of IBGP sessions [BGP11]. In this mechanism a large AS is divided into a set of smaller sub-autonomous systems (confederations or subASes), which are each interconnected in controlled way. The protocol on the interconnecting links is called CBGP (Confederation BGP). Every router belonging to the particular confederation must be fully meshed with others in the confederation or must employ route reflection mechanisms. There is introduced a new attribute AS_CONFED_SEQUENCE for avoiding loops on the level of an AS divided into confederations. It plays the same role like AS_PATH attribute. When BGP routing information traverses subASes, the subAS numbers are added to AS_PATH. When BGP routing information leaves an AS, the subAS numbers are removed. The list of subAS numbers inside an AS is stored in AS_CONFED_SEQUENCE.

3.4.2 TEQUILA

TEQUILA was a European project funded under Fifth Framework Program [TEQ]. TEQUILA stands for Traffic Engineering for QUality of service in the Internet, at LArge scale. The main objective of TEQUILA was to specify, develop, and validate a system that would be capable to dynamically negotiate, invoke, and provision the resources associated to the deployment of Quality-of-.Service (QoS) based IP service offerings over the Internet [ABJ02]. The TEQUILA system provides service guarantees through planning, dimensioning and dynamic control of traffic management techniques based upon the Differentiated Services (DiffServ) architecture [BB98] in a flexible policy-driven manner. The TEQUILA system is composed of a set of elementary blocks that comprise traffic engineering management capabilities [G00], [TAP01a]. It relies upon the use of classical IP routing protocols for the establishment of IP routes, as well as the use of the Multi-Protocol Label Switching (MPLS) technique [RVC01], for the establishment of Label Switched Paths (LSPs) that are expected to comply with the QoS requirements specified by the customers. In TEQUILA, QoS refers to a service offering where one or more traffic/performance parameters (i.e., throughput, delay, loss, and/or jitter) are quantified and guaranteed [G00].

The TEQUILA project addresses the following areas: the customer demands through SLSs (Service Level Specification), the protocols and mechanisms for dynamically negotiating, monitoring and enforcing SLSs, and the QoS-related technologies required for meeting these customer demands (SLS enforcement), including the provisioning, management and intra- and inter-domain traffic engineering schemes to ensure that the network can cope with the contracted SLSs - within domains, and in the Internet at large [G00], [TAP01a], [TAP01b]. TEQUILA system is shown in Figure 3.2.

The functional architecture of the global TEQUILA architecture has the following main functional parts: SLS Management, Traffic Engineering, Policy Management, and Monitoring in addition to Data Plane functionality. The SLS Management is responsible for subscribing and negotiating SLSs with customer, a customer being possibly a service provider. It also performs admission control for the traffic associated/depicted to/in the invoked SLSs. Traffic Forecast component of SLS Management is responsible for mapping and aggregating traffic demands of multiple SLSs having an ingress node and a set of egress nodes requiring a certain QoS and forming a Traffic Matrix. The Traffic

Matrix is then used by the Network Dimensioning (ND) component of Traffic Engineering part. The Traffic Engineering part of the architecture is responsible for dimensioning the network according to the projected demands, and for establishing and dynamically maintaining the network configuration that has been selected to meet the SLS demand. ND is in general centralized and is responsible for mapping the Traffic Matrix onto the network resources by computing a set of optimal routes (by maintaining the link metrics or by setting explicit paths) in order to accommodate the forecasted traffic demands subject to resource restrictions, load trends, QoS requirements, and policy directive and constraints.

From the SmoothIT project point of view two architecture blocks defined in TEQUILA are especially interesting: Dynamic Route Management (DRtM) and Dynamic Resource Management (DRsM). The QoS approach proposed in Subsection 7.1 complements the approach proposed in TEQUILA.

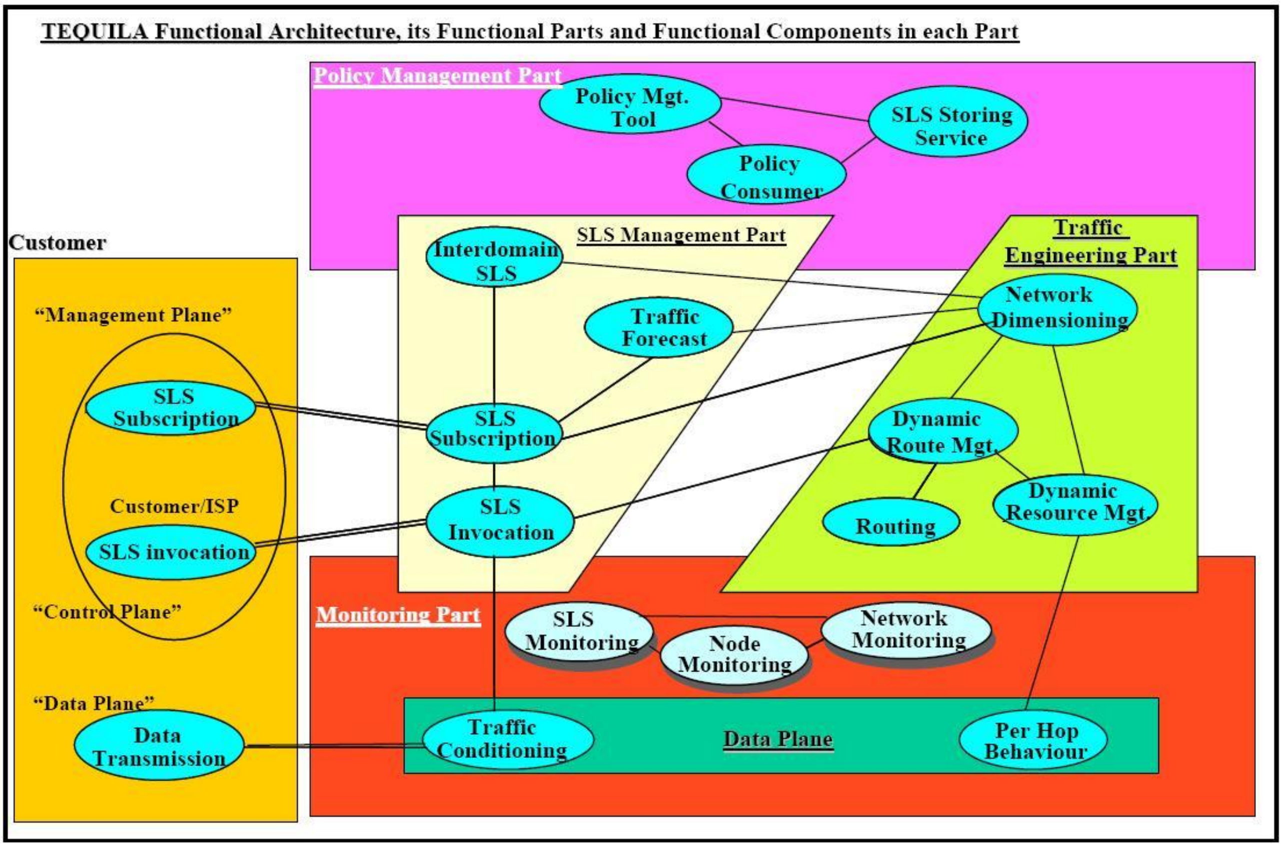


Figure 3.2: TEQUILA Functional Architecture specifying distinct functional parts and components [ABJ02].

Dynamic Route Management (DRtM) is a distributed component located at the routers, responsible for managing the routing processes in the network according to the guidelines provided by ND on routing traffic. This amounts to:

- setting up traffic forwarding parameters at the ingress node, so that incoming traffic is routed to LSPs according to the bandwidth determined by Network Dimensioning,
- modifying the routing of traffic according to feedback received from Network Monitoring,
- issueing alarms/warnings to Network Dimensioning in case available capacity cannot be found to accommodate new connection requests.

DRtM also requests from Network Monitoring statistics about the load incurred by various groups of "addresses". During system operation Network Monitoring also informs DRtM about the QoS performance (end-to-end delay, loss probability and used bandwidth) of the traffic routed through the Label Switched Paths (LSPs) managed by DRtM. In addition, Network Monitoring informs DRtM about the QoS performance of the network PHBs (Per-hop Behavior) used by the managed LSPs.

Dynamic Resource Management (DRsM) is distributed, with an instance attached to each router. It is responsible for ensuring the link capacity is appropriately distributed among a limited number of PHBs sharing the link by setting buffer and scheduling parameters associated with the interface attached to the link, according to Network Dimensioning (ND) directives, constraints, and rules. Specifically, DRsM receives estimates of required resources for each PHB in terms of minimum and maximum bandwidth to be allocated to that PHB, a minimum bandwidth to be allocated in time of congestion (competition from the other PHBs) together with the maximum delay and packet drop probability to be experienced by packets using that PHB. Through these parameters ND specifies an acceptable operational range for the PHB's bandwidth, which has been calculated, based on the traffic forecasts it has received from the SLS Management system. Within the bounds of this margin, DRsM is free to dynamically manage resource reservations. Compared to ND, DRsM operates on a relatively short time-scale (order of minutes). DRsM triggers ND when network/traffic conditions are such that its algorithms are no longer able to operate effectively; *e.g.,* due to excessive high priority traffic, link partitioning is causing lower priority/best effort traffic to be throttled. DRsM may issue over- or under-load alarms to ND respectively if the higher margin is closely approached, or if the PHB's rate has been below the lower margin for a predetermined time. In its simplest form the DRsM is responsible for tracking the utilization of a PHB through the services of a Monitoring system, which is capable of issuing alarms when defined thresholds on PHB rate have been crossed.

3.5 Overlay Incentive Mechanisms

In this subsection we overview tit-for-tat, which is the most prominent incentive mechanism for file-sharing overlays, and give-to-get, which is its counterpart for Video-on-Demand disseminated over peer-to-peer. These studies also shed light to the different requirements of these two applications.

3.5.1 Tit-for-tat

The distribution of a content file in BitTorrent proceeds by peers exchanging data with each other until every peer owns a complete copy of the file. Each peer *p* tries to maximize its own utility, *e.g.,* its own download rate and ultimately minimize its overall downloading time. When two peers experience poor downloading rates they are often able to mutually increase their downloading rates by uploading to each other.

BitTorrent's choking algorithm tries to achieve Pareto efficiency by assuring that peers reciprocate by uploading to peers that upload to them or – the other side of the coin – that peers refuse to upload to peers that do not upload to them. Next, we briefly overview the prevalent such algorithm and explain its rationale; the discussion to follow is based on [C03]. Disallowing downloading from peers is called choking, while allowing is called unchoking, *e.g.*, peer *p* is choked at peer *q* if it is in the neighbour list of peer *q*, but not currently downloading anything. Peer *p* normally becomes unchoked when peer *q* experiences a good download rate from peer *p*. The choking algorithm tries to ensure that when a peer *q* uploads data with a certain rate to peer *p*, it gets the same amount back from *p* (although getting a different part of the common file shared). Thus, some measure of fairness is guaranteed. So-called 'free-riding' is made more difficult by this mechanism. A prerequisite for this method to work is that both peers have content to offer that the other peer needs. This strategy is a variant of the tit-for-tat strategy that was used to play repeated prisoner's dilemma game.

Practically, each BitTorrent peer *p* always unchokes a fixed number of its peers. By default all connections are choked. The default number of peers that can be unchoked by *p* at a time is five. The peer *p* decides once every 10 sec a set of four of its peers to unchoke based on the current downloading rates that *p* experiences by its peers. Each peer keeps track of the downloading rates it receives by all of its peers. Calculating the current downloading rate is difficult and it is based on a rolling 20 sec average, i.e., only the last 20 sec rates are taken into account. This policy deters free-riding, *e.g.,* if a peer *p* downloads from a peer *q* but being a free-rider *p* refuses to upload to *q*, then the next time that *q* will re-determine the list of its unchoked peers, *q* will choke *p* because of the poor (or zero) downloading rate offered thereby.

Additionally, a random peer is unchoked optimistically in order to start new resource exchanges, allowing peers with little content to gain more data to exchange. This is the fifth peer being unchoked *regardless* of the current downloading rate. This policy is called optimistic unchoke because it allows peers to download data even if they have not proven that they will upload something in return and it is applied every 30 sec. This happens basically for two reasons: first, in order to avoid the starvation of new peers in a swarm (which have nothing to offer to other peers initially) and second, for old peers to discover new potentially better connections. For instance, suppose peer *q* is a newcomer and has asked to download some pieces from peer *p*. Peer p not having adequate info about *q*'s behaviour may not upload to peer *q* which could provide a higher downloading rate to *p* than any other of *p*'s unchoked peers. By BitTorrent allowing *p* to unchoke a peer regardless of the downloading rate, *p* could unchoke *q*, which in the next 20 sec will in turn unchoke *p*. As a result *p* has discovered a better new connection. Optimistic unchoke corresponds to always play 'cooperate' on the first round of repeated prisoner's dilemma game.

BitTorrent's choking algorithm implies that a peer p rewards peers that uploaded to p a bit earlier. The purpose of a choking algorithm is to utilize all available resources, *e.g.,* upload capacity of all peers, to provide reasonably consistent download rates for all peers, to "punish" free riders, *e.g.,* peers only downloading and not uploading, and on the other hand reward peers that contribute to the system's overall efficiency. The choking algorithm is not technically part of the BitTorrent protocol, but is an incentive mechanism that assures good overall performance.

3.5.2 Give-to-get

While tit-for-tat has proved to be a successful incentive mechanism for file sharing over BitTorrent, it is not so appropriate for Video-on-Demand disseminated over peer-to-peer, as explained by Mol at al. in [MPM+08]. Indeed, due to the timing constraints inherent to the distribution of the chunks of a video, each peer should select to download some chunks among those whose play-out time is approaching, the so-call high-priority set of chunks. However, for different peers, such sets do not necessarily overlap, due to difference in their arrival times, thus limiting effectiveness of tit-for-tat. Therefore, the authors of [MPM+08] develop “give-to-get”, which is an incentives’ mechanism appropriate for VoD. Below, we briefly overview this mechanism, while maintaining the terminology of BitTorrent as much as possible, in order to also illustrate their differences.

Each peer always maintains a list of 10 other peers, to be henceforth referred to as his neighbors. To this end, a peer periodically checks whether his neighbors are still alive, and if this is not the case it adds new ones. Every δ sec (*e.g.,* every 10 sec), each peer ranks his neighbors according to their *forwarding ranks*, which are computed on the basis of information gathered over the past δ sec. Then, the peer unchokes his top three neighbors. If less than 90% of the upload bandwidth of the peer is utilized, then at most two more neighbors can be unchoked, in order to achieve such a high utilization of this bandwidth. Optimistic unchoking of one peer that is randomly selected from the neighbor list is performed every 2δ sec. Therefore, similarly as tit-for-tat, this mechanism also: a) encourages contribution to other peers, by ranking high in the list those that act accordingly, while b) giving the chance to others to contribute (and thus be subsequently ranked high), by means of optimistic unchoking. The main difference of give-to-get from tit-for-tat lies in how the former derives this forwarding rank. In particular, as stated in [MPM+08] “First, the neighbors are sorted according to the decreasing numbers of chunks they have forwarded to other peers, counting only the chunks they originally received from; if two neighbors have an equal score in the first step, they are sorted according to the decreasing total number of chunks they have forwarded to other peers.” Note that the number of chunks forwarded by a peer’s neighbor is not reported by that neighbor itself; on the contrary, they are reported on request by those peers to whom that neighbor claims to have downloaded chunks, in order for the original peer to avoid selfish misreporting of such information. Also the fact that the first ranking criterion only takes into account the chunks obtained by the original peer (rather than by all peers) results in more peers having an overall opportunity to be unchoked. Notice also here that a peer selects which of his neighbors to unchoke on the basis of their contribution to the *whole system*, rather than to this peer himself, as is the case with tit-for-tat.

For completeness reasons, we also present the chunk selection algorithm of [MPM+08], which of course is not part of give-to-get. In particular, once unchoked by a peer q, a peer p has to decide on which chunk to request from q. In particular, it can only request a chunk i if: q has the chunk, p does not have it and has not requested it before and chunk i is likely to arrive before its associated deadline; these will ensure that the chunk will be in its destination on time in order to be played. Yet, the above conditions do not suffice to fully specify which chunks a peer p will actually request. In particular, the mechanism classifies the chunks missing from a peer p into three sets, namely the high priority, the mid priority and the low priority depending on how close to the current play-out deadline they are positioned. The mechanism prescribes that if p has already started playing the video then it selects to download from a peer q a chunk from the high priority set and in fact the lowest indexed among the eligible ones, i.e. the one that is to be played out sooner; if p is has not started playing the video yet, then it should choose among the eligible chunks of the top priority set the rarest one first, where rarity is estimated on the basis of availability of chunks by the neighbors of p. If no matching arises in the high priority set, then the mid and the low priority sets are visited in this order, and a selection is made on a rarest first basis. Therefore, this selection algorithm tries to first download chunks that are bound to be played among the first ones, particularly if the playback of the video has already started. Rarest-first selection is employed in conjunction with this as a secondary criterion. Finally, playback does not start immediately after the first chunk of a video is downloaded. In particular, in order to start, the peer should first have collected the first h chunks, where h is a parameter. Then it waits further, so that the remaining download time is less than the duration of the video plus 20%. This presumes that the average uploading capacity offered per peer is roughly equal to the video bit-rate plus 20%, so that the downloading of the remaining chunks will be performed timely enough. If this presumption does not apply, then peers should pre-buffer more chunks.

The authors of [MPM+08] evaluate give-to-get by means of simulations and study the impact of the various parameters. It turns out that the mechanism is successful, while it can lead to a satisfactory service for free-riders only in case there are abundant resources in the system. They also mention two limitations. First, give-to-get does not scale with the length of the video, thus, possibly leading to lower quality for longer videos. Moreover, the derivation of forwarding rankings depends on information reported by other peers. As has been extensively studied in other contexts (*e.g.*, reputation-based mechanisms), such reporting can be untruthful or even malicious, thus affecting the effectiveness of the mechanism.

Mechanisms related to give-to-get have been considered in other research works as well. In [CLM+08], Cuoto da Silva et al. use a give-to-get-like peer selection strategy for peer-to-peer streaming in general. Instead of gathering data about the actually uploaded data from potential downloaders, it uses their upload bandwidth as the selection criterion. This Bandwidth-Aware (BA) algorithm selects upload destinations proportionally to their upload capacity. The assumption here is that they will utilize that bandwidth to disseminate the content, and that peers with a larger bandwidth are better sources, similar to the scheme described in [MPM+08]. Since this algorithm is also used for live streaming, the delay for chunk distribution was also tested and found to be improved by the BA approach.

In contrast to [MPM+08], it relies on peers to be trustworthy in their declaration of their bandwidth. Results for the presence of malicious peers show that, depending on the ratio of normal to malicious peers, the latter can severely reduce the improvements achieved.

The upload capacity of peers is also used to improve the performance of a mesh-based streaming overlay by Picconi and Massoulié in [PM08]. Here, the overlay connections are created so that each peer has neighbors with a certain average upload bandwidth. Additionally, the chunk push scheduling algorithm at sources also prefers peers with a high capacity to increase the system performance. The results presented show that with these and other improvements, a mesh-based system offer near-optimal performance in comparison to tree-based approaches, while being easier to create and maintain under churn.

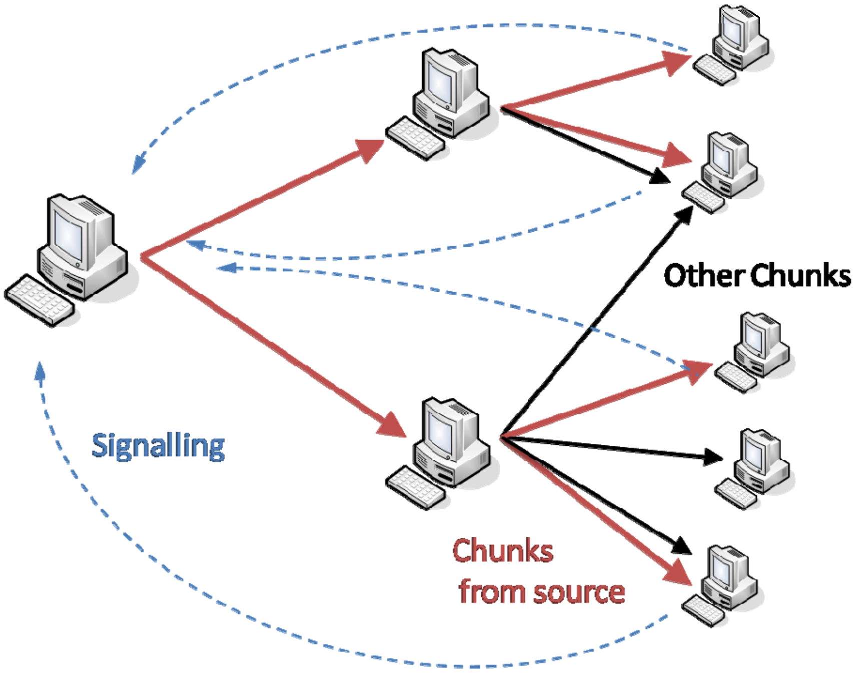


Figure 3.3: Give-to-get.

3.6 Models on Inter-ISP Traffic Generation and Management

SmoothIT places significant attention to the incentives of the ISP, and particularly to monetary incentives associated with the reduction of the charges paid for inter-ISP traffic. Below, we review certain research works that are related to this issue.

In [WCL08] Wang et al. formulate and analyze a game where two ISPs with a peering agreement decide on their access prices charged to their users. The setting studied includes the two peering ISPs, each of which also has a transit agreement in order to route traffic to the rest of the Internet. The paper makes several assumptions, in order to fully specify the relevant model. In particular, these assumptions concern: a) how users select which ISP to subscribe to, b) how much traffic flows locally, in the inter-ISP link and in the link connecting each of them to the Internet, c) caching issues and capacity provisioning by the ISPs so as to keep their users satisfied. Customers are charged at flat rate p_1 (resp. p_2) by ISP1 (resp. ISP2); these two prices can be different. Customers decide on which of the ISPs to join, based on both prices p_1 and p_2 as well as on the utility they obtain from the level of "congestion" in each ISP, a proxy of which is the market share of each ISP. By analyzing this game, the authors derive the main result, namely the equilibrium splitting of customers among the two ISPs, i.e. their market share, together with the corresponding equilibrium prices. Note that this equilibrium splitting can only be derived numerically by solving a non-linear equation. The authors also present extensive experimental investigation of their model. As already mentioned, the authors adopt a detailed model on the splitting of traffic of each ISP j locally, to the other ISP i and to the rest of the Internet. To this end, the authors adopt a *gravity* model. This splitting of the traffic results in a total charge for ISP j and this in turn determines his best-reply price p_j to the price p_i announced by the other ISP. However, for both ISPs, the charge associated to the transit agreement depends on the resulting long term *average* traffic volume, *rather* than on the 95th percentile (mostly assumed in the context of SmoothIT, cf. Section 4), or some other statistical index of the traffic dynamics.

3.6.1 P4P: Provider Portal for Peer-to-Peer Applications

The P4P project [XK08, XY08], as stated in the mission statement of the P4P Working Group (see [P4P]), is "*a set of business practices and integrated network topology awareness models designed to optimize ISP network resources and enable peer-to-peer based content payload acceleration.*" Therefore P4P proposes a network aware peer-to-peer paradigm that envisages maximizing delivery, reduce costs and improve performance. P4P sets to allow the peer-to-peer overlay networks to optimize traffic within each ISP, which aims not only to reduce the volume of data traversing the ISP's infrastructure, but to also create a more manageable flow of data.

The P4P framework consists of a control-plane component and an optional data-plane component. In the control plane, P4P introduces iTrackers to provide portals for peer-to-peer to communicate with network providers. iTrackers allow P4P to divide traffic control responsibilities between peer-to-peer overlays and ISPs, and to also make P4P incrementally deployable and extensible. iTrackers communicate with appTrackers in peer-to-peer applications in order to obtain important information and to provide suggestions related to the decision making process in the overlay. iTracker implements cache discovery providing users with a list of "good peers" (e.g. users in same PoP range, users with higher uplink capacities, etc) and cache servers within the ISP cloud. The benefits expected from this approach are: a) better performance of peer-to-peer applications, e.g. faster downloads for BitTorrent, and b) less inter-domain traffic for ISPs (due to optimization), thus leading to reduced charges. Note that Annex of D1.1 [D1.1A] provides an overview of P4P and a detailed comparison with SmoothIT. Here we focus on the modeling studies of P4P, on the basis of which the aforementioned optimization is accomplished. These studies are presented in [XK08, XY08].

In particular, Xie et al. in [XK08, XY08] formulate an optimization problem whereby the routes of the various application sessions are to be chosen so as to optimize link utilization in the underlay. This optimization problem can be decomposed into separate problems of optimal selection of route for each application session. However, the cost metrics used in each such selection are those of "interest" to the ISP; namely the Lagrange multipliers corresponding to constraints of the ISP's optimization problem, rather than "actual" underlay metrics affecting directly the user's actual performance. This optimization problem is extended to include the charge for inter-domain traffic as an optimization objective. To this end, the authors use the q-percentile charging model to evaluate to which extent an ISP can save traffic costs by using the proposed iTrackers. In particular, the authors consider the scenario where a provider (*e.g.*, a Tier 1 ISP) charges another ISP network (*e.g.*, a Tier 2 ISP) that is connected to it. To this end, it keeps track of the "traffic volume that the ISP generates during every 5-minute interval." Thereby, they consider inbound and outbound traffic separately rather than taking their difference; thus, this analysis cannot apply to the common charging scheme for inter-domain traffic as presented in Section 4. For one charging period of 30 days, the charging volume $p = qt(V,q)$ is calculated. V contains the traffic generated by the ISP with every element v_i of V corresponding to one 5-minute measurement interval. The elements are sorted in ascending order and $p = qt(V,q)$ is the $q*|V|$-th value in the vector V. Thus, the charging volume, i.e., the metric for the inter-domain performance considered by P4P, specifies the amount of data, which in turn influences the resulting charge. When considering inter-domain optimization as well, the authors include a threshold for the charging volume in each inter-domain link.

The authors of [XK08, XY08] perform experiments by distributing a file through BitTorrent clients using the Abilene network and a connected "ISP-B". They conclude that the use of an iTracker reduces ingoing as well as outgoing inter-domain traffic of ISP-B. This consequently leads to savings of inter-domain traffic costs. In addition to the inter-domain performance, they also evaluate the intra-domain performance of the P4P system in terms of link utilization. They show that link usage in the ISP-Bs network decreases significantly when the peers use the iTracker. The reason is that connections between peers involve less links when nearby neighbors are selected. This is also said to reduce costs, but it is not specified in which manner.

Related to the above is the work by Goldenberg et al. [GQX+04] for the case of multi-homing. These authors present a series of smart routing algorithms in order to optimize cost and performance for multi-homed users. Multi-homing has to do with multiple external links (either to a single or multiple ISPs) while smart routing controls how the generated traffic is distributed among those links. Throughout the analysis, the percentile-based charging is considered for the cost optimization objective. In fact, the *($q\% \times I$)-th charging volume* is computed, where I is the number of intervals in a charging period and q is the percentile value, i.e. $q=95$ for a 95th-percentile charging scheme. Note that since the user-generated traffic is considered, the 95th-percentile charging scheme refers to the amount of outbound traffic. One key issue for cost optimization is the determination of the charging volumes for each ISP. Once those volumes are known, then the traffic can be distributed in such a way that the number of intervals in which ISP k serves more than its charging volume of traffic does not exceed $(1-q_k) \times I$. In this sense, the authors define *peak* and *non-peak intervals* depending on whether traffic can be assigned without having any ISP receiving traffic more than its charging volume or not. Having these notions as basics, several algorithms are designed for cost optimization. They are categorized into *fractional or integral flow assignment algorithms*, depending on how traffic flows are treated (i.e. if they can be split among several ISPs or not) and into *offline and online algorithms*, if traffic volumes are known a priori or have to be predicted. The main outcomes of the analysis are two routing algorithms, the GFA-offline (Global Fractional offline flow Assignment) and the GIA-online (Global Integral online flow Assignment) algorithms. Capacity constraints are also considered in these algorithms and extensions are provided to achieve performance optimization (in terms of latency) as well. The proposed algorithms are compared with several simpler algorithms. The evaluation investigates the cost optimizations and the performance optimizations under cost constraints achieved. Furthermore, the effects of smart routing are evaluated, based on if smart routing considers the effects on other flows, which is the outcome when several smart routing users interact or when smart routing users interact with single-homed users. The results assure that the specific smart routing algorithms can achieve lower costs and higher performance, without hurting other traffic.

The relevance of the above overviewed works for SmoothIT is as follows: [GQX+04] deals with allocation of outbound traffic in the case of multi-homing, while SmoothIT deals (among other ISP incentives) with the reduction of the overall charge for inter-domain traffic. Thus, some of the underlying ideas for the algorithms of [GQX+04] could be employed by SmoothIT in cases of multi-homing (see Subsection 5.2), but proper adaptations should be introduced. Furthermore, P4P has similar objectives like SmoothIT, but takes the cooperation between the overlay and the ISP for granted rather than addressing the incentives of the two stakeholders. The ETM approaches of SmoothIT aim to lead to a TripleWin situation. Nevertheless, the techniques employed by P4P in [XK08, XY08] as well as by [GQX+04] to deal with traffic that is charged according to the 95th percentile charging should be considered by SmoothIT, particularly by the approach employing dynamic locality promotion (see Subsection 5.2).

3.7 Concluding Remarks

In this subsection we have overviewed a variety of research works related to the modeling issues addressed by the project as well as to the specification of ETM approaches. The applicability of the concepts, ideas and/or techniques employed in these works to SmoothIT varies. In particular, the research works overviewed in Subsection 3.1 that comprise analytical models pertaining to cases with information asymmetry do offer some game-theoretic modeling concepts that can prove interesting for SmoothIT, but do not deal directly with overlay applications that involve multiple alternative sources the selection among which greatly affects traffic patterns. The analytical models for BitTorrent and other peer-to-peer systems overviewed in Subsection 3.2 offer interesting results and some modeling techniques that can be employed in SmoothIT. The modeling works for interconnection economics overviewed in Subsection 3.3 are not directly applicable to SmoothIT. In Subsection 3.4, we overviewed background on routing and traffic management issues with emphasis on BGP and its impact, which are very relevant to the ETM approaches to follow, particularly to those of Subsection 5.1 and Section 7 respectively. Also, some of the algorithms used in the works for management of inter-ISP traffic in the presence of some interconnection agreement (overviewed in Subsection 3.7) can be employed for the ETM approach of Subsection 5.2 employing dynamic locality promotion. In particular this applies to algorithms for dynamic inter-domain routing under 95-th percentile charging. Finally, the overview of incentives mechanisms for peer-to-peer systems in Subsection 3.6 reveals some basic differences of file sharing and video-on-demand and their associated mechanisms, which should be taken into account by the relevant ETM approaches.

4 Charging schemes for interconnection links: the 95th Percentile Rule

One of the main incentives of an ISP, when employing ETM to traffic generated by overlay applications, is to reduce the charges incurred for inter-domain traffic. Thus, there is a close interplay between the charging scheme for such traffic and what ETM can achieve and how. These motivated the studies carried out within SmoothIT for interconnection agreements between ISP and one of the prevalent inter-domain charging schemes termed the 95th percentile rule. Thus, this section, we review as a basic the structure of the Internet and the interconnection agreements between Internet Access Providers and Internet Service Providers. In particular, we introduce peering and transit agreements and explain an instantiation of the commonly used 95th percentile rule as prominent charging scheme for transit agreements. The sensitivity of this rule to different parameters is later on investigated. Finally, the consequences of the application of this charging model with respect to ETM are demonstrated.

4.1 Internet Structure and Interconnection Agreements

The Internet is characterized by an informal hierarchy of operators. As a result, the commercial arrangements between two operators will depend on where each of them falls within that hierarchy [MPD+07]. Prior to describing the commercial relationships between operators and the charging models that exist, we provide a quick insight into the hierarchical structure of Internet operators.

The Internet is structured in three tiers. The first level includes the ***Tier 1 IAPs*** (Internet Access Providers), which are large telecommunication operators covering large geographic areas and having significant numbers of Points-of-Presence (PoPs). Tier 1 IAPs interconnect with each other and form the “backbone” of the Internet. In the second level, we have the ***Tier 2 ISPs***, which usually have some network of their own, limited to a geographic region, and they partly rely on Tier 1 IAPs to obtain world-wide access, by purchasing some level of transit. At the same time, Tier 2 ISPs can have interconnection agreements with other Tier 2 ISPs. ***Tier 3 ISPs*** are purely re-sellers of Internet access services, provide retail services to end customers, and rely solely on interconnection agreements with Tier 2 ISPs in order to gain access to the Internet.

Based on the aforementioned hierarchy and the business relationships between the stakeholders at each tier, different charging rules apply. These rules are not standardized and depend on the details of the specific agreement and the status of both parties. However, all agreements fall under two types: the ***peering agreement*** (or free agreement) and the ***transit agreement***. The former defines the rules that traffic exchanged should comply with, in order for no charging to occur between the interested parties; the latter defines the way charging is computed, based on the traffic volumes exchanged. However, for both types, penalties are also defined in the case of non-compliance to the rules by the interested parties.

The actual interconnection pricing schemes between operators of the hierarchy are summarized below:

- Between Tier 1 IAPs: the goal here is to preserve the balance of traffic between the interconnecting parties. Hence, no charging is applied as long as balance exists or if an up to 5% imbalance is measured. Otherwise, bilateral interconnection agreements between both parties will take place accordingly. However, details depend on the individual contracts.

- Between a Tier 1 IAP and a Tier 2 ISP: no traffic symmetry is expected here, so the Tier 2 provider pays to Tier 1 provider an amount specified by the details of the traffic rules in place. Depending on the status, the Tier 2 ISP could be charged for the volume of inbound traffic or for the difference between inbound and outbound traffic.
- Between a Tier 2 ISP and a Tier 3 ISP: in this case, a Tier 3 ISP always pays the Tier 2 ISP for the inbound traffic (the data downloaded).

Throughout this section, when we refer to an ISP we consider the case of a Tier 2 ISP. This is the most interesting and richer case, since such an ISP is large enough, offers content, creates large amount of traffic, and has as customers both Tier 3 ISPs and end users. Hence, one objective of a Tier 2 ISP is to reduce his interconnection costs, especially since peer-to-peer applications affect those costs in an unpredictable way. From the ISP's point of view, an ETM mechanism aiming at TripleWin will have to introduce ways to decrease such costs. The case of Tier 3 ISPs is simpler and the objectives are just a subset of what will be addressed in the case of a Tier 2 ISP. More specifically, the ETM approaches that deal with traffic locality can be directly deployed in such a scenario and provide some concrete solutions.

4.2 The 95th Percentile Rule

One of the most prominent charging models for transit agreements is the 95th percentile rule. In fact, such a rule is used in the agreements (a) either between two Tier 1 IAPs for accounting or (b) between a Tier 1 IAP and a Tier 2 ISP for charging. We will focus on the latter case. The difference d_x between inbound and outbound data throughput of the Tier 2 ISP is measured for every $\Delta t=5$ minutes time slice x. Then, the Tier 2 ISP is charged on the monthly 95th percentile of the differences per time slice. The rule can be summarized as follows:

$$C_{month} = P95\{d_x\} \cdot Price/Mbps.$$

4.3 Estimation of Parameter Sensitivity

One of our main tasks related to the 95th percentile pricing scheme, is to determine which parameters affect the generated costs, so that we can devise ETM mechanisms that decrease those costs for the ISP. Since traffic is the only factor that affects the level of the 95th percentile, we consider the following parameters that characterize it: i) the volume of traffic, ii) the pattern type of traffic, iii) the asymmetry (or not) of traffic and iv) the duration of the observation window, during which the amount of exchanged traffic is measured.

As a very simple example, we consider the following scenario to formulate the 95th percentile rule. It has to be noted that this scenario is just used for demonstrating basic relationships between model parameters and the actual 95th percentile. Within the SmoothIT project, we will elaborate on realistic models for the exchanged traffic. Nevertheless, in the simple example we assume that time is discretized and the exchanged traffic volumes in inbound and outbound direction, I and O respectively, are i.i.d. and follow a geometric distribution with parameter q, i.e.:

$$P(X = y) = q^y (1-q). \qquad (1)$$

For the difference *D* of the two random variables *I* and *O*, the convolution of the probability distributions *P(I = x)* and *P(O = x)* is computed

$$P(D = x) = \sum_{y=\max(0,x)}^{\infty} P(I = y)P(O = y - x) = \ldots = \frac{1-q}{1+q} q^{|x|}. \qquad (2)$$

The cumulative distribution *P(D ≤ x)* is calculated from

$$P(D \le x) = \frac{1}{1+q} + \sum_{y=1}^{x} \frac{1-q}{1+q} q^x = 1 - \frac{q^{x+1}}{1+q}, \qquad (3)$$

and the *αth* percentile can be computed as

$$P_\alpha = \begin{cases} \frac{\log(1+q-\alpha-\alpha q)}{\log q} - 1 & \text{, if } q \ge \frac{1-\alpha}{\alpha} \\ -\frac{\log(\alpha - \alpha q)}{\log q} & \text{, otherwise} \end{cases} \qquad (4)$$

In the above model, we consider the symmetric case, since both inbound and outbound traffic follow the same distribution with the same mean (*q*). An extension to this model is to consider different means for two distributions. Doing so, equations (2)-(3) become more complex, yet they still lead to a closed-form expression. Another extension would be to consider full asymmetry, with different distributions for inbound and outbound traffic.

One parameter that can be influenced and has direct effects on the above formulas is the amount of traffic flowing in both directions. One way to do so is applying locality awareness, as already mentioned. However, depending on the application (file sharing or video-on-demand) locality may decrease the traffic, either in both directions (tit-for-tat for file-sharing), or only in one direction (give-to-get for video-on-demand). It is obvious that depending on the case applicable and the corresponding outcome, the 95th percentile rule will give different results in terms of the impact of the reduction of traffic to the reduction of charge.

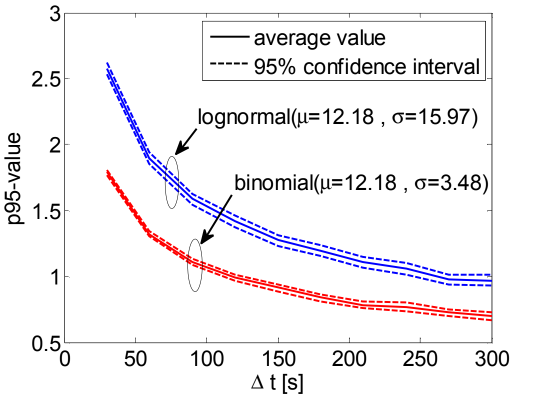


Figure 4.1: Impact of different distributions and duration of capturing window.

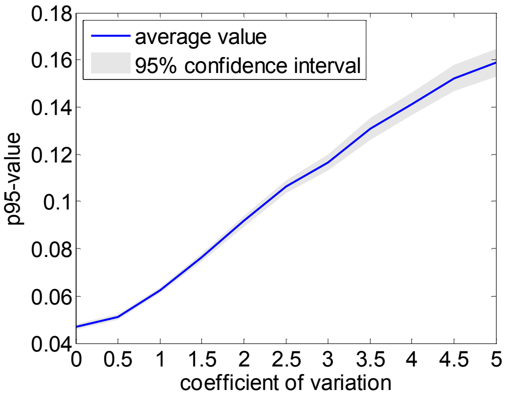


Figure 4.2: Impact of traffic pattern by varying the packet size for window of 5 minutes.

We have also conducted some numerical experiments to study more complex scenarios. Here, we investigate the actual effects of different traffic patterns, as well as the effect of the duration of the observation window with a simple simulation model. The results are depicted in Figure 4.1 and Figure 4.2.

In Figure 4.1, on the x-axis, *Δt* (the duration of the observation window in seconds) is given while the 95th percentile value is depicted on the y-axis. Note that the default value for *Δt* is 5 minutes, i.e. 300 seconds, so we study the effects of shortening the measurement interval. In the simulation, in order to vary the type of traffic pattern, we implemented a procedure where packets arrive according to a Poisson process, while the packet sizes follow a binomial and a lognormal distribution respectively. Note that we have chosen the same traffic pattern in inbound and outbound direction. Then different time intervals *Δt* are used to compute the 95th percentile of the differences in inbound and outbound direction. The simulations were repeated 1000 times. Figure 4.1 shows the average 95th percentile out of the repetitions as well as the 95% confidence intervals. It is apparent that not only the traffic pattern but also i) the time interval and ii) the distribution and the variance of the packet size play an important role in the calculation of the 95th percentile value.

Therefore, we take a closer look on the impact of the traffic pattern on the 95th percentile value in Figure 4.2. We varied the size of the packet size which is given on the x-axis. On the y-axis the corresponding 95th percentile value is plotted. As we can see, the value increases with the variability of the traffic pattern. In this case, we used the standard value for the capturing window of 5 minutes

4.4 Consequences of 95th Percentile Rule

Following the previous analytical and experimental results, it becomes obvious that various parameters affect the 95th percentile rule. Consequently, the solutions to be chosen as ETM-enabled mechanisms will have to take these parameters into consideration. For instance, if we have symmetric traffic reduction which might be a result of a locality promotion mechanism, then this has only a minor impact on the costs. We will further elaborate this in on-going studies.

As future work in the direction of interconnection charging and costs, we have to determine the conditions and the parameters the affect mostly the 95th percentile rule, and validate the results through analysis of real traffic traces. Furthermore, we have to investigate the overall costs when keeping traffic locally; on the one hand we save inter-domain traffic and costs, on the other hand we may increase the number of “internal” hops and therefore costs or we might introduce additional components, like IoPs or SIS which consume CAPEX and OPEX costs. Another direction is to examine different charging schemes. Depending on the results, we may propose more appropriate pricing rules for Tier 1, Tier 2 or Tier 3 providers, so that they can maximize their benefit when providing locality promotion. Finally, even if an ETM approach results in reductions of charges generated by the 95th percentile rule, it is plausible that the Tier 1 provider imposing this tariff adopts a new one that leads to higher charges in the presence of this approach. This research direction may even call for a game-theoretical analysis.

5 SIS-enabled Locality-awareness Approaches

The ETM approaches to be presented are classified into four main categories: (a) ETM approaches that are based on the concept of the SmoothIT Information Service (SIS); (b) ETM approaches based on QoS/QoE-related incentives as well as on mechanisms; (c) an approach employing a new entity in the overlay, namely the ISP-owned peer; and (d) other approaches not falling into any of these three categories. In this section we deal with the first category of approaches. In the next sections, the approaches falling into the rest of the categories are presented.

As presented in D3.1, the SmoothIT Information Service (SIS) conveys information between the overlay application and the (underlay) network. It is accessed by overlay applications and provided by a network operator in order to achieve ETM of overlay application traffic. In this section, we describe a class of ETM approaches that focus on promoting locality in order to reduce the inter-domain traffic. In particular:

- The *BGP-based Locality Promotion* approach offers to the overlay end users an information service to get a ranked list of peers according to the BGP information, which is usually quite stable. The intelligence part of this approach is fully specified.
- The *Centralized SIS and Dynamic Locality* is similar to the previous approach but also takes into account more dynamic information coming from the network status monitoring.
- Finally the *Locality-aware Tit-for-Tat/Unchoking* aims to select local peers among the recently joined ones instead of selecting them randomly.

In this section all these approaches are described; the advantages and disadvantages of the different implementations are also discussed. Note however that to add, that the ETM approaches presented this section (as well as in Sections 6 and 7), except for that in Subsection 5.1, are mostly at the level of ideas and proposals, which have been identified and checked in terms of plausibility to be applicable. The next steps within SmoothIT will fully specify, simulate and evaluate these approaches, typically those ones with the highest optimization potential, some of which will also be implemented in the trials.

5.1 BGP-based Locality Promotion

The *BGP-based locality promotion* ETM approach uses the BGP routing information of an ISP in order to provide a locality information service to overlay applications and to rank potential peers located in other ISPs according to the preferences of the originating ISP. The mechanism aims to reduce interconnection costs by preferring inter-AS links according to business relations of ISPs and to improve overlay performance by selecting connections with shorter AS path length. The mechanism, however, does not provide means for the differentiation of peers located in the AS of the originating ISP. Similar proposals for use of BGP information for locality promotion exist, like the work presented in [AFS07] where no details are given for the implementation of such a service. The mechanism presented here offers a fully-specified algorithm on how to use BGP information to rank a list of peers.

5.1.1 Description of the Approach

This mechanism provides a peer ranking service to overlay applications based on the locality information gathered from BGP. 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 an ISP. Intra-AS peers cannot be differentiated by this mechanism and they are considered being equally ranked. 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 rank peers according to the preference of the ISP. In order to use this service, the overlay application sends a list of IP addresses as input to the service. These IP addresses represent potential peers that the application would connect to. The service performs the peer ranking and sends back the list of IP addresses with a preference value assigned to each address. Based on the ranking and the preference value, the overlay application can adapt its peer selection algorithm.

As input for this mechanism, the BGP routing table is required. (Note however, this is already available to the ISP.) This includes network masks, representing the destination, the local preference, the AS path, and the MED (Multi Exit Discriminator) BGP attributes. 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: 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, 70-79 for providers [CR05]. The BGP-based locality promotion ETM mechanism assumes that local preference values are set according to this scheme and it uses the local preference, AS hop count, and MED BGP attributes. Additionally to the BGP routing information, this mechanism requires a list of IP addresses as input from the overlay application. As already mentioned, this list represents the peers to be ranked. If the list contains a large number of potential peers and that the overlay application would connect only to a subset of the peers in the list, then this approach is expected to optimize the performance of the application. This is the typical scenario for the most popular contents in the overlay applications (at least the Tracker provides 50 possible peers and then, a new overlay is created to discover more peers while the peer usually establishes around 10-20 connections to share the content), which, in fact, are the most important contributors for the overlay traffic.

The mentioned BGP routing information is used to rank the list of IP addresses. The result of this is a list of IP addresses with a preference value attached to each address. Based on the ranking and the preference value, the overlay application can adapt its peer selection algorithm.

The ranking algorithm runs on the SIS server on the ISP's side, while the adapted peer selection algorithm based on the preference value is located in the overlay application. The proposed mechanism includes the following interactions between overlay application and SIS server:

1. The peer looks for potential peers via the overlay-specific search mechanism, *e.g.*, it requests a list of peers from a Tracker. As a result the peer has a (long) peer list, containing IP addresses.
2. The peer sends the peer list to the SIS server and requests a *“peer ranking”*.
3. The SIS ranks the peers in the list according to the BGP information.

4. The SIS returns the ranked list of peers to the overlay application. In the list a preference value is assigned to each IP address.
5. The peer decides, based on the preference values and not just on random selection, which peers in the list he will connect to.

The benefits of the approach include the reduction of interconnection costs by preferring inter-AS links according to business relations of ISPs (customer, peering, provider) and the improvement of performance by selecting connections with shorter AS path length.

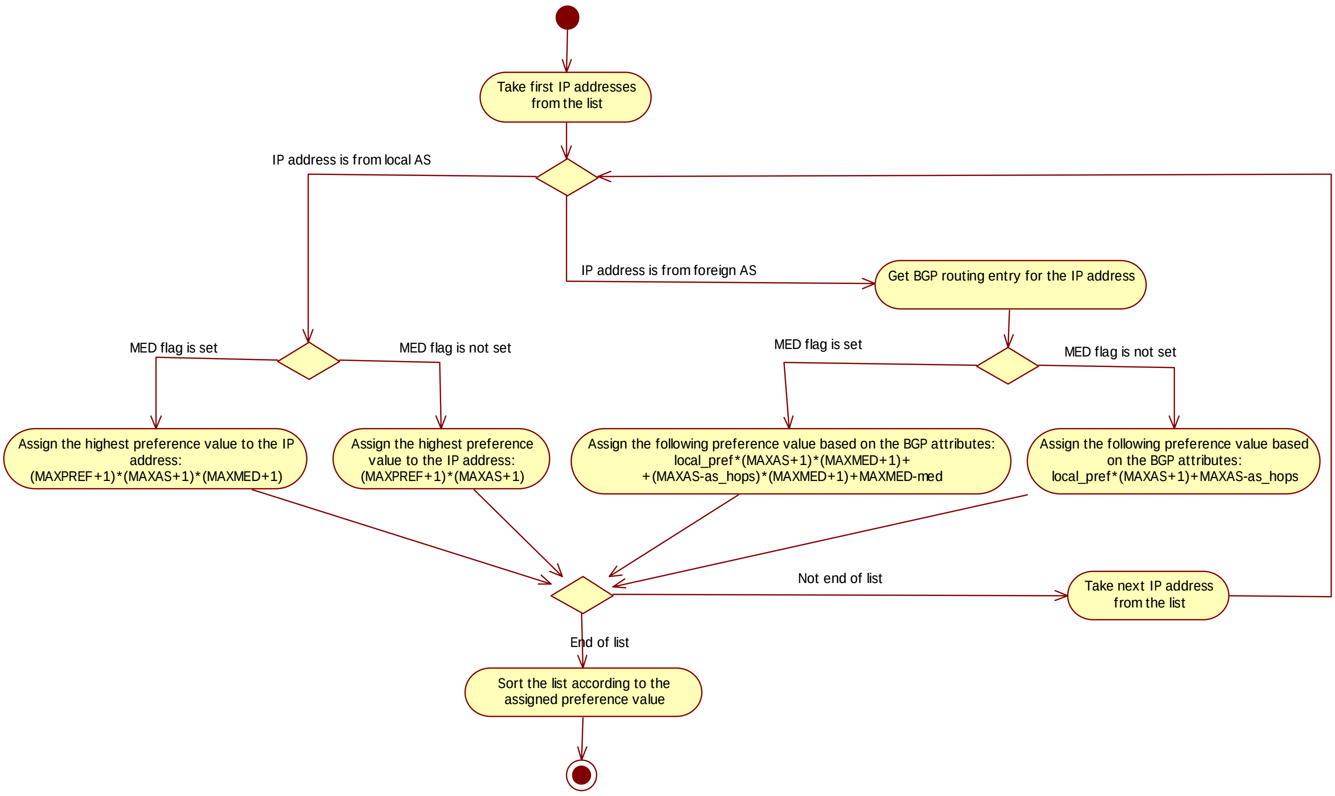

Take first IP addresses from the list
IP address is from local AS
IP address is from foreign AS
Get BGP routing entry for the IP address
MED flag is set
MED flag is not set
MED flag is set
MED flag is not set
Assign the highest preference value to the IP address: (MAXPREF+1)*(MAXAS+1)*(MAXMED+1)
Assign the highest preference value to the IP address: (MAXPREF+1)*(MAXAS+1)
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
Not end of list
Take next IP address from the list
End of list
Sort the list according to the assigned preference value

Figure 5.1: Peer Ranking Algorithm.

The peer ranking algorithm is shown in Figure 5.1. The same algorithm is described in Subsection 8.2.1.2 of D3.1, yet in a somewhat different form of presentation. The algorithm takes each IP address from the list received from the overlay application and assigns a preference value to each of them. Since the MED attribute is set by neighboring ASes, it can only be used in the algorithm if neighboring 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.

The peer ranking algorithm works as follows:

- If the IP address is from the local AS of the ISP that operates the SIS server, the algorithm assigns the highest preference value to the address. The highest preference value equals to *(MAXPREF+1)*(MAXAS+1)*(MAXMED+1)* if the MED flag is set, and to *(MAXPREF+1)*(MAXAS+1)* if the MED flag is not set, where MAXPREF is the maximum value of the local preference BGP attribute, MAXAS is the maximum value of the AS hop count attribute, and MAXMED is the maximum value of the MED attribute. All three maximum values are configuration parameters and they are to be set according to the attribute values used by the ISP.

- If the IP address is from a foreign AS, the algorithm reads the BGP routing entry related to the IP address and assigns a preference value based on the BGP attribute values in the corresponding routing entry. The assigned preference value equals to *local_pref*(MAXAS+1)* (MAXMED+1)+(MAXAS-as_hops)*(MAXMED+1)+MAXMED-med* if the MED flag is set, and to *local_pref*(MAXAS+1)+MAXAS-as_hops* if the MED flag is not set, where *local_pref* is the local preference, *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 list, it has assigned a preference value to each IP address in the list. The list can be sorted based on this preference value in descending order. The higher the preference value, the higher the IP address is preferred. Note also that by the formula presented, the MED values influence the relative ranking of two IP addresses that are external to the ISP only if their *local_prefs* and *as_hops* are equal. Otherwise, the relative ranking gives priority to the peer with higher *local_pref* or if the *local_prefs* are equal to the peer with the smallest *as_hops*.

Recollect that there might be the case that several subASes exist in a single AS (see Subsection 3.4.1.5) for scalability reasons. Obviously, the existence of subASes inside an AS affects the AS_PATH number that is considered by our algorithm. The two apparent alternatives for dealing with the case are either to keep the AS_PATH as it is, i.e. counting also the subAS hops, or subtract from the number of hops the number of subASes traversed, i.e. AS_PATH - AS_CONFED_SEQUENCE. The final decision depends on the objective of the algorithm.

In our approach, we consider the original AS_PATH value, since we believe that it gives a better approximation for "distance" also in case of confederations and it captures better the performance effects that the traversing of multiple subASes introduces. This is partly done in order to consider the local AS traffic as well, hence considering that traversing multiple subASes will have an impact on the performance, in the same way that traversing multiple inter-domain links does. A further consideration is to place an SIS server for each subAS, instead of having one SIS for the entire AS domain. But this approach is left to be examined as an extension of the current algorithm.

5.1.2 Classification and Implementation

The aforementioned approach introduces a new information exchange mechanism as well as some new architectural components: the SIS server, the SIS client and the Metering Component (more specifically the BGP information module that provides all the BGP-related information to the SIS). Considering the design space presented in Section 7 of D3.1, there is a relation to the Control Freak approach.

5.1.3 Qualitative Evaluation

The players that are involved in the proposed mechanism are: the ISP that provides the BGP information and the SIS functionality, the end users (peers) and the Overlay Service

Provider implementing the SIS client in the overlay application. Concerning the cooperation required, besides the standard cooperation between users and the overlay provider, we see that cooperation between the users and the ISPs is also required. In particular, the user should be aware of the SIS and follow its recommendations. Peers cannot really be forced to do so. However, due to the way the algorithm ranks the peer list, it is expected that it is to the user's benefit to indeed follow these recommendations, rather than choosing randomly peers from the list. Nevertheless, it is plausible that peers may deviate from these recommendations from time to time, in order to verify that they are really beneficial for them. Concerning the implementation efforts, such a mechanism does not introduce any heavy implementation requirements in both the SIS server and clients; the former will rank the peers according to the BGP information, while the clients will just have to introduce one query in the code (facilitating the migration of end users applications). Finally, regarding the measurements' load and timescales, the BGP routing table has to be read periodically. Since the routing table is not changing very often, the timescale is large, *e.g.*, once per hour or day.

5.2 Centralized SIS and Dynamic Locality

The approach described in this subsection is motivated by the fact that, for an ISP, locality awareness does not necessarily imply that locality should always be promoted. In fact, locality awareness should be considered only when the network status and/or the status of the interconnection agreements imply that it is beneficial to do so.

In practice, many ISPs are multi-homed, which means they have multiple provider links to reach the Internet backbone. The availability of multiple paths gives the ISP the opportunity to influence overlay traffic so that the Internet infrastructure is optimally used. Optimization refers to the use by the ISP of interconnection links with lower delay and congestion levels, higher throughput as well as to the use of links that minimize the inferred monetary costs. Since paths and routing decisions for an ISP are fixed or rather, decided on a long-term basis, a way to influence overlay traffic is to change the peer selection process so that the resulting traffic matrices serve the optimization purposes. The balancing of traffic among exits does not always imply that locality is promoted. In fact, the smart peer selection will influence the overlay traffic whenever this is necessary. The randomness of peers in a peer list returned by the tracker will be preserved, as long as certain thresholds related to performance and monetary metrics are not violated. Hence, the objective of this approach is dual: increase the performance perceived by the users while at the same time try to minimize the costs incurred b the ISP or, at least, not to increase them.

Note that in the case of a single-homed ISP, the scenario is simpler: the ISP should promote or not traffic locality, depending on the performance and monetary metrics related to the single exit. The decision of the ISP will adapt to the current traffic conditions but it will be binary: either "enforce" locality by affecting the peer selection process, or lave overlay traffic unchanged. It is obvious that this case is much simpler than the case of multihoming. Thus, in the rest of this section we will deal with multi-homed ISPs.

5.2.1 Description of the Mechanism

As already mentioned, the ISP provides some underlay information that can help the overlay to adapt better to network changes. This is done by the SIS entity (owned by the

ISP) and the underlay information includes location and routing information about the remote peers, such as the information about AS number for each peer and the ID of “exit” for each remote (external) peer, based on BGP next-hop information.

Furthermore, the SIS has access to running measurements concerning the performance characteristics of inter-domain links (such as the number of lost packets and delay) and the interconnection agreements (the volume of inbound/outbound traffic). Taking these into account, the SIS can advice each requesting peer which (remote) peers to connect to, based on a ranking algorithm that assigns preferences to each peer of the peer list obtained by the Tracker or by any other mechanism. This procedure is also described in Subsection 5.1, and thus we henceforth focus on the new ideas, beyond what was presented there.

What is new in this approach is that for the ranking, performance and monetary criteria related to the inter-domain links are also considered in a hierarchical way. First, a ranking value is assigned to a peer, depending on the BGP information about the location of the peer. Then, performance and monetary criteria come to refine the ranking. As an example, if a peer is located to an AS than can be reached through exit *A*, then the peer is ranked based on BGP information. Then, depending on the congestion level of the interconnection link, the ranking value can be increased or decreased. In the same sense, if the interconnection costs related to that specific exit are high enough and the metrics affecting theses costs are above a certain threshold, e.g., the current difference between incoming and outgoing traffic is in the top 5% of the measurements, then the ranking value is affected accordingly (in this case, the ranking is decreased).

It becomes obvious that under low congestion and low cost conditions, all requesting peers will receive the same ranking value for a remote peer *p*. However this might not be a desired property since it might lead to flash-crowd behaviors, i.e., all peers connect to peer *p* (assuming that his ranking is high enough), the respective interconnection link gets congested and/or the respective costs are raised. To deal with such situations, some randomness should be introduced to the ranking heuristic so that not all peers that residing in the same AS and belonging to the same swarm, receive exact the same ranked lists by the SIS.

Note also that such a heuristic might as well promote the connection with remote peers instead of promoting locality, in the case that certain ratios between inbound and outbound traffic must be preserved in order to keep charges at a minimum. Hence, the described ranking method combines conditional locality and randomness.

When the network conditions or the costs related to an “exit” change, there is a question whether actions should be taken to inform peers, which have already received a ranked list, that new ranking is required, or if information sent only to new requesting peers suffices.

A possible extension of this mechanism is to provide ranking for internal peers as well, i.e., peers of the list that belong to the same AS with the requesting peer. In this case a domain could be divided into neighborhoods and each peer in the domain should belong to a neighborhood, based not only on proximity criteria but also on performance criteria. Hence, the ranking of peers will consider this information as well, in order to avoid having some parts of the domain congested. It should be noted that the notion of neighborhood is reminiscent of the clusters employed in [P4P].

The proposed mechanism requires the following input:

- List of candidate peers, acquired from a tracker or from other peers
- The AS number for each peer
- The ID of the "exit" for each remote (external) peer based on BGP's next-hop info
- Optionally, the neighborhood ID where the remote (internal) peer belongs to
- Performance metrics for the interconnection links, such as delay and packet loss rate
- Running measurements related to the interconnection agreements, like the volume of the inbound/outbound traffic

The expected output of the mechanism is the following:

- Ranking of peers included in the list of peers
- Optionally, alarm signals to inform peers with outdated lists

Below, we provide the flow of events that summarize the proposed mechanism:

1. A peer requests a peer list from the Tracker
2. Tracker returns a peer list to the requesting peer
3. The peer forwards the list to the SIS
4. The SIS ranks the peers of the list
5. SIS returns the list back to the peer
6. The peer decides which peers of the list to try to connect to (several tries)

The flow of messages developed for the proposed mechanism is presented in Figure 5.2.

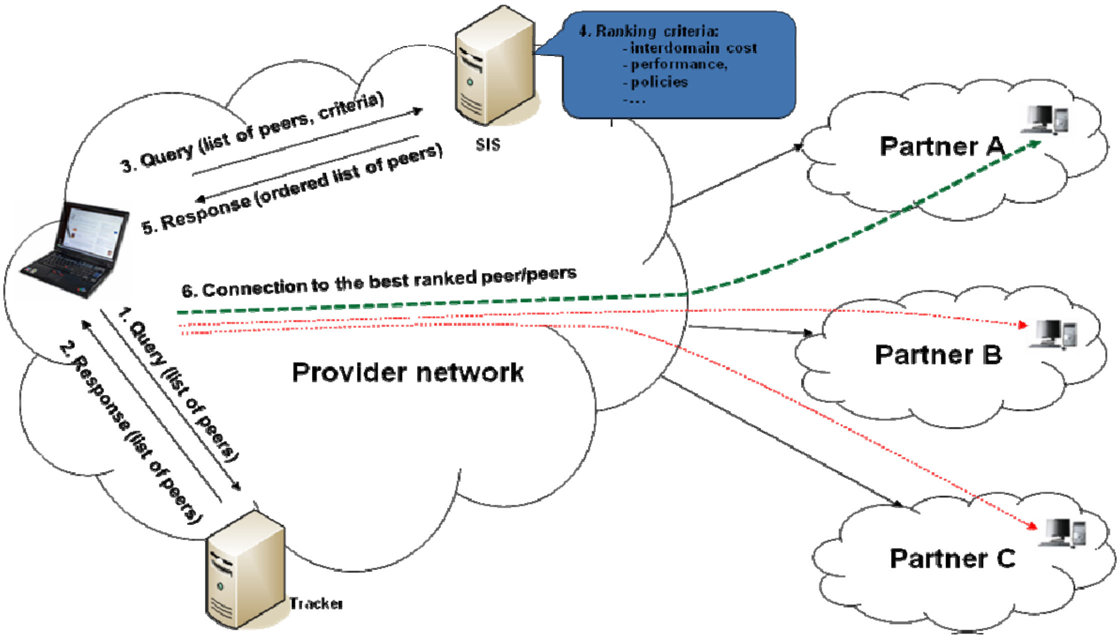


Figure 5.2: The flow of messages developed.

The intelligence of the mechanism is located at the SIS. A *Ranking Module* is comprised in the SIS, with the objective to rank the peers in such a way as to increase the performance levels and minimize the interconnection costs. Thresholds about *when* an action should be taken also apply. These thresholds have to do both with performance and monetary metrics. In the following subsections we provide two examples of how the optimization problem can be formulated, how such thresholds can be defined and what actions can be taken when they are violated.

5.2.1.1 A Formulation of the Optimization Problem

Assume that we have a simple topology where the ISP's domain has only two exits, i.e. to BGP egress nodes. Also, time is slotted. At time *i*, *$x_j(i)$* is the traffic passing through exit *j* (*j=1,2*) and c_j is a target threshold in the utilization of exit *j* possibly related with some performance metrics (*e.g.,* congestion) and primarily to the interconnection charging scheme, i.e. the 95th percentile rule; see Subsection 0 for more details. To this end, the time unit can be taken as 5 minutes, namely the timescale of sampling traffic for the purpose of computing the 95th percentile. At time *i+1*, *y(i+1)* denotes the estimated traffic to pass through both exits. (Our goal is to split this traffic *y(i+1)* between the two exits, having in mind also that a percentage of the traffic at time *i* will still exist at time *i+1*. This splitting can be effected by recommending to the requesting peers, which peers to prefer to download from in time slot *i+1*. More precisely, there holds:

$$x_1(i+1) = \beta\, x_1(i) + \alpha(i+1)\, y(i+1)\,,$$

$$x_2(i+1) = \beta\, x_2(i) + (1\text{-}\alpha(i+1))\, y(i+1)\,,$$

where α(*i+1*) denotes the percentage (to be decided) of splitting the new traffic at time *i+1* and β denotes the percentage of old peers that remain in the next timeslot. Obviously, the parameter β has to do with the statistics of the overlay traffic. Note that the mechanism has some hysteresis, in the sense that violation of the threshold at time *i* leads to an action on the traffic at time *i+1.*

An interesting extension of the above formulation is to consider that not only upcoming traffic is effectively split among the two exits, but also a percentage of the existing traffic can be moved from one exit in order to avoid the violation of certain thresholds. Again, this rearrangement can be effected through appropriate recommendations to peers whose downloads are already in progress. Hence, in this case we will have that:

$$x_1(i+1) = \gamma_1(i+1)\,\beta\, x_1(i) + (1\text{-}\gamma_2(i+1))\,\beta\, x_2(i) + \alpha(i+1)\, y(i+1)\,,$$
$$x_2(i+1) = \gamma_2(i+1)\,\beta\, x_2(i) + (1\text{-}\gamma_1(i+1))\,\beta\, x_1(i) + (1\text{-}\alpha(i+1))\, y(i+1)\ \ ,$$

where γ_j(*i+1*) denotes the percentage (to be decided) of the existing traffic to remain to exit *j* at time *i+1* and *1-* γ_j(*i+1*) denotes the percentage of traffic to arrive from the other exit. It is obvious that it will hold $\gamma_1 * \gamma_2 = 0$, since there is no point in the mutual exchange of traffic between the two exits. Moreover, at any time *i+1*, there would hold that $\gamma_1 \neq 0$ or $\gamma_2 \neq 0$, if $\alpha = 1$ or $\alpha = 0$ respectively, that is, the existing traffic is to be controlled only of control of the new expected traffic does not suffice. One could generalize even more the formulation that all γ_1, γ_2 and α *are* bounded away from 0 and 1, due to the fact that not all traffic is controllable. Indeed, this applies to non peer-to-peer traffic, which has a single possible source.

In all of the above case, the ISP's objective is to minimize some sort of cost related to the volume of traffic passing through the two exits. At this point we do not define strictly the notion of cost, but we provide the relation of the cost with the volume of traffic. Hence, at each timeslot i, the ISP has to derive the appropriate α, γ_1 and γ_2 that minimize the following expression $\sum_j C(x_j(i))$, with $C(x_j(i)) = \dfrac{w_j}{1 - \dfrac{(1-\varepsilon)x_j(i)}{c_j}}$ denoting the cost introduced when traffic of volume x_j passes through the exit *j*. The variable $w_j \le 1$ is a threshold-dependent cost variable, different for each exit, depending on the type of interconnection agreement and $\varepsilon \le 1$ is a small value, defining how flexible we can be when violating the threshold c_j. The rationale behind these variables is that the thresholds to be set are not strict ones and can be violated from time to time without incurring prohibitive costs to the ISP. This is depicted in the diagram of Figure 5.3. There we observe that when the exit utilization is 1, the cost does not go to infinity, as is the usual case for congestion costs associated with the violation of a capacity constraint. On the contrary, the cost goes to a finite though high enough value, so as prevent the thresholds from being violated often.

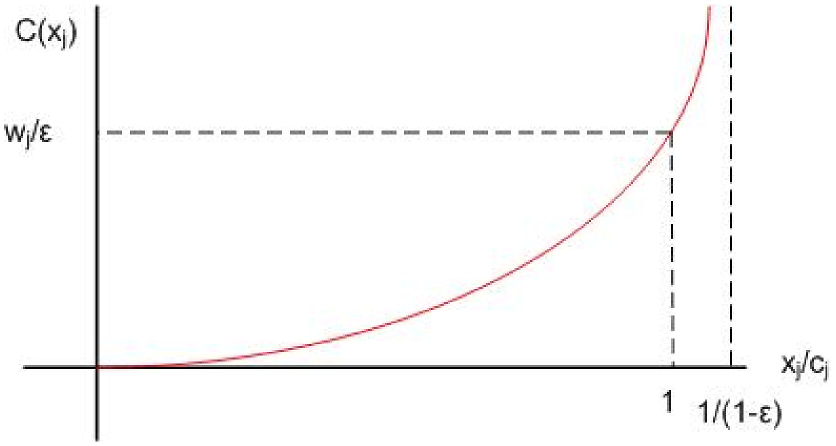


Figure 5.3: The cost function related to the utilization of an exit and the thresholds introduced.

As a concluding remark, note that we have considered the case of inbound traffic only, since the mechanism described in the previous subsection affects the way peers are selected and, as a result, the distribution of incoming traffic among the interconnection links. A point to be further investigated is that we can change the objective function to capture the difference of *outbound - inbound* traffic, as it is the metric considered by most interconnection charging schemes, *e.g.,* the 95th-percentile scheme.

5.2.1.2 A Mechanism for Handling Path Changes

In this subsection we propose a mechanism for managing overlay network by making changes in an underlay network. The handling path mechanism does not interfere directly with overlay network. In this mechanism we change path in an operator domain working on the level of an underlay network. Peer-to-peer traffic is considered in the presence of other type of traffic. By changing the path in a physical network we influence on overlay networks, in the sense that some peers, after change may be not available or are not popular for partners in a peer-to-peer network.

Our strategy is delivering as much traffic as it is possible in actual state of an operator network. Conditions in a network change dynamically and we react to these changes. If some part of an operator network does not provide a sufficient amount of resources, we move part or whole traffic to other regions of the network where these resources are available. We discard the specific traffic if resources are not available or there is no alternative path in the network.

The path change mechanism can be also used in the case when specific traffic (for instance voice) increases and there are not enough resources on actual path for this traffic. Suppose that there are resources on this path that are allocated for other type of traffic. We can free some resources by changing the path of that other traffic and reallocate these resources to voice traffic.

This mechanism operates by building on top of routing. Thus, we assume that standard routing protocols like BGP work in the network. Consider an egress router with one interface for input traffic and two interfaces for output traffic. Figure 5.4 simplifies the real situation with the egress routers of a domain and the multiple transit providers of an ISP, but we make this simplification for clarity purposes.

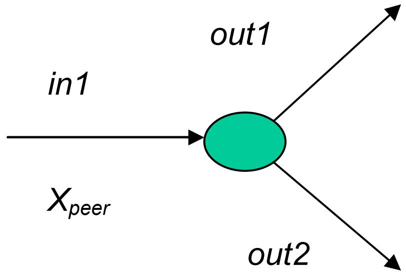


Figure 5.4: Egress router interfaces.

By *in1* we denote the entire input traffic on this interface, which amounts to the entire traffic to be output from the ISP. X_{peer} is the part of this traffic that refers to peer-to-peer traffic. Symbols *out1* and *out2* refer to the traffic on the two output interfaces. Let's suppose that the entire X_{peer} traffic is initially routed through the *out1* interface. We want to influence this amount of traffic, in the sense that we can split the traffic between the two output interfaces if, for some reasons, a violation of some performance or economic thresholds dictates to do so. One possible approach for this purpose is to define a *reaction function*. The shape of this function specifies when a threshold is violated and what part of the traffic should be moved to the other interface. In Figure 5.5 we show an example of the reaction function related to each output interface, as well as the volume of traffic passing through them.

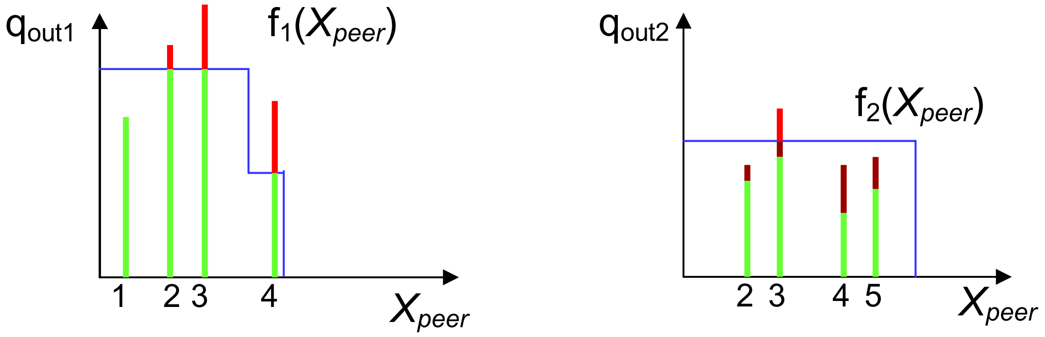


Figure 5.5: Reaction functions.

On the vertical axis, q_{out1} represents the load level of output queue on interface *out1*, including the peer-to-peer traffic and the background traffic. On the vertical axis, we have

different realizations (patterns/volumes) of peer-to-peer traffic (X_{peer}) as time elapses. $f_1(X_{peer})$ represents the reaction function for the interface *out1*, depicted the blue line (we have the respective notation for interface *out2*). In the presence of a particular value for X_{peer} traffic we have the specific load of output queues, which is denoted on the charts by vertical bars. As already mentioned that main output interface is *out1*.
For the traffic pattern 1 we observe that queue load for the interface *out1* is below f_1, meaning that we do not have any reasons to reroute X_{peer} traffic. For point 2, the total queue load exceeds the value of f_1, so part of the X_{peer} traffic has to be moved to interface *out2*. By doing so, we increase the respective traffic load on the interface *out2*. But since the traffic load does not exceed the allowed level specified by the reaction function f_2, no extra action occurs. On the contrary, in the case of point 3, some part of traffic has to be discarded. In the case of point 4, due to dynamic changes of network conditions, the shape of the reaction function f_1 changes and forces us to transfer a larger part of the X_{peer} traffic to interface *out2*, while the shape of f_2 results in no dropping of packets.

In Figure 5.5 the shape of the action function has the form of the step function. For some value of the X_{peer} traffic it is expected that this traffic can generate problems on the path attached to the interface f_1. So in such a situation it is preferred to move more X_{peer} traffic to the interface *out2* by decreasing the previously allowed amount of the X_{peer} traffic in the interface *out1*. The shape of the f_1 function results from some global network information, the increase of the traffic over a specific level on the interface *out1* of the considered router can be critical for traffic management on some other routers in other regions of the considered network. Sometimes the situation can be cured by drastically limiting traffic in the interface f_1, but there is no need to discard traffic; the network allows moving massive amount of traffic to the interface f_2. The point 5 in Figure 5.5 indicates that the X_{peer} traffic was so high that it was decided to move whole that traffic to the interface *out2*.

Generally the shape of the action functions are time dependent; this way traffic changes in the network can be taken into account and the influence on the peer-to-peer traffic is possible. In Figure 5.5, one can treat the presented five points as consecutive in time interface load. In one moment the flow through the interface *out1* is greater, because there are enough resources on the path attached to this interface. Later, when information from the network has been collected, especially that the congestion on the mentioned path is expected or just has appeared, by changing the shape of the f_1 function some X_{peer} traffic can be moved to interface *out2*. Generally information about the actual and predicted state of the network in the presence of a particular traffic pattern in some regions or the whole network can be exemplified in a shape of action functions. Action functions can be considered from a global network perspective taking into account traffic flows on many distributed routers. These functions can be also used only for local traffic management, focusing on flow behavior on a particular router without considering rest of the network.

This approach is rather general. In fact, we expect that we can find self-organizing procedures which allow adjusting shapes of reaction functions in an optimal way. The aim of the optimization can be different, one time we may want to maximize background traffic in the presence of X_{peer} traffic, while in other situation to maximize X_{peer} traffic. The conditions in the network change dynamically, so we expect optimization procedures and traffic engineering procedures to be a dynamic operation.

5.2.2 Classification, Architectural Design, and Implementation

The aforementioned approach introduces a new information exchange mechanism as well as a new architectural component, namely the SIS entity, which now has different intelligence than what was presented in Subsection 5.1. Considering the design space presented in Section 7 of D3.1, we see a relation to the Honey Pot approach, but the objective of SIS is not to attract peers to stored content but help them reach optimally the content. Some notions of the Control Freak are also relevant here, since the SIS (and the ISP) proposes rankings that somehow control the way traffic is routed outside the domain.

With respect to the effort required to implement this approach, the intelligence part may not be very demanding. The crucial point though is the information required as input. If online measurements are possible then there is no problem either. But if online measurements are not available, especially for metrics related to the interconnection charging schemes, then some other way to obtain the required information (probably in a more aggregated form) should be developed. Note also that application-awareness is required. Finally, the alarm-based approach is more complicated since it requires a stateful SIS implementation and also listening capabilities in the peers. Moreover, the neighborhood awareness poses some more complexity. The recommended approach is to first implement the basic functionality and then add the other functionalities incrementally.

5.2.3 Qualitative Evaluation

In the proposed mechanism two players are involved: the ISP, which provides the measurements and the SIS functionality, and the end-users. Both players have incentives to use the proposed mechanism. The ISP has incentives to use such a mechanism since he can this way affect (as much as possible) the decisions in the overlay to his own benefit. Peers cannot be forced to follow SIS suggestions, unless observable performance improvements are offered. In either case, we would like to reach a “win – non-lose” or a “win – win” situation.

Concerning the effectiveness of the proposed mechanism, if the required information is available, then the mechanism can react to network and/or charging changes and affect peer selection decisions which will lead to a close-to-optimal outcome. The available information is a crucial point since real-time measurements regarding performance are required and measurements related to the interconnection charging schemes are expected.

When using the proposed mechanism some questions arise. One of them concerns peers which should be informed when a change occurs. It has to be decided if outdated lists are updated or only new peers are informed. The percentage of the random and “exit-aware” suggestions is one more topic which has to be investigated as well as the level of ranking differentiation among different peers in order to avoid flash-crowd events. When the alarm-based approach is applied the answer for the question how different the alarm signals sent simultaneously to different peer should be has to be given.

The issue of Network Neutrality arises only in the case that overlay traffic is treated in a privileged way compared to background traffic, since the mechanism implements triggers that cause overlay traffic to be rerouted, due to changes in the sources selected. No other legal issues arise, since the ISP/SIS is content-agnostic. These issues will be further investigated. Finally, no security and privacy issues are raised, since SIS belongs to the ISP and handles monitoring information provided by the ISP’s monitoring components. This information may have to be aggregated but still remain useful.

5.3 Locality-aware Tit-for-Tat/Unchoking

In BitTorrent both tit-for-tat and optimistic unchoking are key for the provision of incentives to contribute to the sharing of a file in a swarm. In particular, tit-for-tat provides the incentive for reciprocation, while optimistic unchoking gives the opportunity to new peers to be offered resources for which they will have to reciprocate subsequently due to tit-for-tat. The mechanism proposed in this subsection aims to influence both of them for attaining the objectives of ETM.

5.3.1 Description

This mechanism offers an easy way to integrate network locality information into the peer selection of BitTorrent-like Peer-to-Peer systems.

In case locality awareness is beneficial for both ISP and BitTorrent (more bandwidth available locally), BitTorrent will eventually converge to a state where the nearest neighbors in the network are utilized. The responsible BitTorrent mechanisms for this are tit-for-tat and optimistic unchoking; see also Subsection 3.5.1. A BitTorrent client following the standard protocol maintains typically 50 connections to neighbors, 5 of which are actively used for upload. 4 of these 5 connections are chosen according to the upload rate of the remote peers to the client (tit-for-tat) and one connection is chosen randomly among the whole set of neighbors of the client (optimistic unchoking). The reasons for optimistic unchoking are to give recently joined peers a chance to get chunks that can be traded further and to give a client a way to try out peers in order to eventually discover peers that can give a better upload rate. (Note that a different peer may have a better upload rate as a currently uploading peer, but would not be unchoked as the currently uploading peer has an overall lower upload rate but currently a higher upload rate and is therefore according to the normal tit-for-tat more preferable than the other peer.) Thus, after a sufficient time, each peer will eventually find through optimistic unchoking and tit-for-tat mechanism the peers with the highest upload rate in the neighbor set.

By biasing the optimistic unchoking in a way that gives nearer neighbors a higher possibility to be chosen optimistically, the BitTorrent client will try out local peers first. This will give the opportunity to discover local peers faster. But as still other peers will be chosen optimistically too, the local peers will only be kept if their bandwidth is actually better as the non-local neighbors' bandwidth. The biasing is done in a way that the probability of choosing a peer for upload should correspond to the distance in the network to the client node. It remains to define a proper function that maps physical distance to the probability of unchocking. Of course, two equally distant nodes should have the same probability.

5.3.2 Details

As additional input this mechanism only requires location information about neighboring peer-to-peer nodes. The SIS can be used for this purpose. Alternatively, as the number of neighbors is supposed to be not exceedingly high, the location information could be obtained through local measurements: the peer itself measures the hop count or latency to its neighbors and thus estimates the distance. It is important to note that the information about the distance of the neighbors is required to be concrete and in absolute numbers, as this is needed to calculate the probabilities for biasing the optimistic unchoking.

The additional flow of events is simple; it includes the query sent from the client to the SIS for distance between the client and a neighbor, and the corresponding answer of the SIS to the client.

The complete intelligence of this approach, if the SIS is granted as given, is contained in the modified unchoking algorithm of the BitTorrent client. It will choose the neighbors for uploading according to the distance between itself and the neighbor. The function used for the weighting of locality in the unchoking mechanism assigns probabilities to distance values. Thus, it determines the ratio between near and far neighbors that are optimistically unchoked. This function takes a probability distribution as an input. For robustness and performance reasons the function should give also far peers a reasonable chance to be unchoked. There is of course a trade-off between performance and locality promotion. As already mentioned in the beginning of Subsection 5.2, locality must be promoted whenever necessary. This issue will be considered in the design of the function.

Note that for this approach an order between the neighbor candidates (for instance an ordered list as response from the SIS) is not equally good, although in principle it could also be employed as follows: the probability of optimistic unchoking is computed as a function of each peer's position in the list and a normalization factor, which of course depends on the number of such peers. In this simplified approach, the queries sent by the peer to the SIS ask only for a ranking of the list.

5.3.3 Classification

The proposed mechanism can be classified as an Information Exchange mechanism; the information gained from the SIS is the only input that is used. The ISP has to provide a SIS to be used by the overlay clients. In addition, also the overlay application has to be adapted to use the information of the SIS in the above specified way.

5.3.4 Qualitative Evaluation

The players involved regarding the proposed mechanism are overlay (BitTorrent clients) and the ISP (SIS provider). All applications using a similar mechanism to tit-for-tat can profit from this kind of mechanism. These include other mesh-based file-sharing systems as well as BitTorrent-like video-streaming systems.

This mechanism leads BitTorrent clients to try out local neighbors first.

The ISP has therefore a strong incentive to use this mechanism and offer a SIS system, as using more local neighbors will potentially reduce its cross-domain traffic, inter-domain traffic costs and reduce congestion in inter-domain links.

If local peers give the BitTorrent client a better performance, it will find them faster, thus the performance of the BitTorrent client is improved and the mechanism leads to a beneficial situation for both players. On the other hand, if the local peers have a worse bandwidth than the other peers, the tit-for-tat of BitTorrent will eventually select the other peers as neighbors. Therefore, in such a case the mechanism may extend the time that the BitTorrent client needs to converge to an optimal performance state in terms of selected neighbors. But the mechanism will not prohibit the use of more distant neighbor peers, and therefore not reduce the performance as strict rules for the ratio between near and far neighbors in the neighbor set. As already mentioned, such a strict mechanism has been proposed by Bindal et al [BCC+06]

A limitation of the proposed mechanism is the initial neighbor set selection. In a centralized BitTorrent system the tracker chooses randomly from the whole set of participating clients a subset to be given to new clients on joining the system. Without changing the tracker, this mechanism can then only optimize the search for local clients within this subset. The client could ask repeatedly for new neighbor lists and eventually discover the whole set of clients in the system.

The only change in implementation introduced by this approach is that the optimistic unchocking functionality of the BitTorrent application must be altered. However, the new clients (i.e., those using our locality biased neighbor selection) can interact with the ordinary BitTorrent clients; the protocol between BitTorrent clients is unchanged by this approach. The new clients would use the SIS as described and communicate with ordinary clients as before.

The mechanism takes as a parameter the probability distribution for the weighting of distance, which determines the ratio of far and near clients that will be optimistically unchoked.

Regarding legal or privacy issues, there are no negative effects expected, as the ISP offers the SIS application agnostic and no personal information on users is published.

The biased tit-for-tat fits best to the “Honey Pot” scenario from the Architectural Design Space of D3.1, as users can be attracted with better performance and are not forced to use the new service.

6 A Mechanism Introducing ISP-owned Overlay Entities: Insertion of ISP-owned Peers in the Overlay

This section focuses on an ETM approach of different spirit than those of Section 5, namely the introduction of an ISP-owned peer (IoP). This is an entity that aims at increasing the level of traffic locality within an ISP and at improving the performance enjoyed by users of peer-to-peer applications. The IoP runs the overlay protocol but with certain differences that serve the aforementioned purposes. Its insertion constitutes an active intervention of the ISP to the overlay, that does not require any collaboration between the ISP and either the overlay or content provider, although such a collaboration would improve the effectiveness of the approach.

6.1 Description

An **I**SP-**o**wned **p**eer (IoP) is an entity that aims at increasing the level of traffic locality within an ISP and at improving the performance enjoyed by the users of peer-to-peer applications. The IoP, either belongs to an ISP's premises and is controlled by the ISP itself; or is a regular but **h**ighly **a**ctive **p**eer (HAP) that is granted by the ISP with extra resources, *e.g.,* higher downlink/uplink bandwidth, at no extra cost. In principle, if dynamic adjustment of the end-user's bandwidth is possible, then the end-users might even not be aware of this enhancement. However, agreement between the ISP and the HAP is also meaningful, in order to assure an extended seeding time by the peer. In any case, the IoP is assumed to run the overlay protocol, *e.g.,* BitTorrent, but with some small differences that serve its purposes. For instance, the IoP is capable to unchoke more peers than the regular ones in order to exploit its extra uplink capacity. Since the IoP runs the overlay protocol, we also assume that it is capable of storing the content it downloads and uploading it back to the network.

Four possible cases can be distinguished for the deployment of such a solution:

1. **Plain insertion of IoP in a peer-to-peer network (*e.g.,* BitTorrent):**

   All peers are assumed to run the original protocol of the application, say BitTorrent, which will be henceforth considered as the base scenario*.* No other mechanisms such as locality awareness are employed by the ISP, and no interconnection agreement is considered. That is, the overlay, *e.g.,* the tracker, is *not* aware of the IoP's existence and real identity but treats it as a regular peer. In this case, the IoP is expected to be preferred by other peers due to the tit-for-tat principle employed by BT's unchoking 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. While this can be attained if it is indeed owned by the ISP, it cannot be taken for granted if the IoP corresponds to an HAP that is granted extra resources by the ISP. Under the insertion of the IoP, reduction of inbound inter-domain and consequently increase of intra-domain traffic is expected. Therefore, reduction of inter-connection costs for the ISP may be achieved. At the same time, end-users' QoE is expected at least to be retained or even better improved due to the higher resources of the IoP. However, as mentioned below, such mechanisms should be carefully deployed, since unlimited localization of traffic may lead to maintenance cost increase for the ISP.

2. Combination of IoP with locality-awareness mechanisms

The use of locality-awareness mechanisms that affect the overlay network's operation is considered here imposed by the ISP. Furthermore, depending on the implementation, these mechanisms could be either transparent to the peers – i.e., they run along with the original protocol – or non-transparent – i.e., a modified version of the protocol is needed to run them. Metrics that could be used as proximity criteria are a.o. BGP information, autonomous system, RTT, or the number of hops. Due to these locality-awareness mechanisms the IoP would be mostly preferred by peers that are 'closer' to it according to one or more of the aforementioned proximity criteria, while offering a high upload bandwidth due to the resources it possesses. The combination of the IoP insertion with locality awareness is expected to further reduce inbound inter-domain traffic, while no important end-users' QoE improvement is expected when comparing to the previous case, since locality has proven so far to be more efficient for the ISPs, *e.g.* [BCC+06].

3. Insertion of IoP and ISP-OP agreement

All peers are assumed to run the original protocol. However, there is an explicit agreement between the ISP and the Overlay Provider (OP). Due to this agreement, the OP, *e.g.,* by means of the overlay tracker, favors the IoP when replying to peers' requests. Additionally, when the IoP serves as a seed for a specific file, then the OP could include the IoP's address only in reply messages to peers that belong to the same AS. Otherwise, also connections to non-local peers are required in order to avoid content unavailability. Another way to block connections between the IoP and non-local peers is traffic throttling by the ISP, but then deep packet inspection (DPI) techniques should be employed which implies network neutrality (NN) violation. An agreement between ISP and OP results in mutual benefit, because the ISP's customers receive the OP's service at a better QoS, while the objectives of the ISP are also served. The OP in this case can charge the ISP for promoting its IoP. Additionally, the ISP could grant the OP with resources for the meta-info data service if needed, *e.g.*, web hosting, storage, bandwidth. Such agreements could also be established in environments where locality-awareness is employed.

4. Insertion of an IoP and an ISP-CP Agreement

All peers are again assumed to run the original protocol. The only difference is that the ISP has established some kind of agreement with a Content Provider (CP). Due to this fact, 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. Essentially, the IoP acts as a seed, rather than as a cache that intercepts the requests. Thus whenever a peer wishes to download this specific content, it automatically connects to these IoPs. Moreover, due to this agreement, the ISP can be charged for the content by the CP.

The information required as input is, first of all, since the IoP runs the overlay protocol, the overlay information that each other peer needs to participate in the overlay network, *e.g.,* the swarm in the case of BitTorrent. This information, which is listed below, is already available:

- IP address of the tracker which is contained as meta-info in the torrent file,
- IP addresses of other peers that participate into the swarm which are obtained by the tracker's reply,

- File chunks' ids and number also known from the torrent file,
- Chunks that each peer has; this information is sent by a peer to all of its neighbors.

It is technically impossible for an IoP to participate in all swarms. Thus, in order to decide which files to download, the IoP requires information about:

- Content popularity
- Swarm size
- Distribution of swarm, i.e. whether there are enough peers in the ISP's networks that participate in the swarm.

While the information on distribution can be deduced to some extent by the tracker's reply to the IoP, the content popularity and swarm size are harder to obtain. Thus, the IoP runs the risk of inefficient exploitation of its resources, by influencing to the distribution of a file that it cannot have a considerable impact for the ISP.

Additionally, when more than one IoPs belong to the same ISP, then information about the swarms the other IoPs – or peer-IoPs – participate in is also desired. However, for scalability purposes the most popular content should be downloaded to more than one IoPs. This content duplication could be deployed either in a centralized manner, *e.g.,* by an ISP-owned entity that keeps track of all IoPs' state, or in a distributed way, *e.g.,* by information exchange between IoPs. Probably this information exchange can take place only between the IoPs that belong to the same ISP. If IoPs that belong to different ISPs are also needed to communicate, then extra inter-ISP agreements are required. In the case of HAPs, this information is not required, since the HAP downloads the files that its end-user selects. The key here is to appropriately select the HAPs so that they can really be influential for the ISP.

Also, the ISP-owned peer does not require any network information itself. However, when combined with locality-awareness mechanisms, the information which is used as input to these mechanisms is additionally required.

On the other hand, the information offered as output is the chunk number and ids a peer has. This information is sent to its neighbors in the context of the overlay communication. Furthermore, if we assume more than one IoPs in the same ISP, then content duplication should either be avoided or aimed based on the popularity of the content. If this decision is made by each IoP separately in a distributed way, then each IoP should periodically inform its peer-IoPs about the swarms it participates in. In the case of HAPs, no such communication can be considered.

Below, the event flow both for the IoP and the regular peer or HAP is described:

Event flow for the IoP:

1. After an IoP has decided which content to download, it downloads the torrent file that contains the meta-info about the content file.
2. The IoP sends a request message to the tracker whose IP address in included in the torrent.
3. The tracker sends a reply with a – random or localized – list of peers that participate in that content file's swarm.
4. The IoP sends request messages to its neighbors in order to start downloading the file chunks.

5. Chunk exchange follows the well-known unchoking and chunk-selection algorithms of the protocol until the IoP downloads all file chunks. While the downloading proceeds, the IoP informs its neighbors about the chunks it has. Due to its high downlink/uplink bandwidth the IoP is expected to be unchoked by seeds or other peers more frequently others.
6. After the IoP has a complete copy of the file, it serves as a seed for this file.

Event flow for a regular peer or HAP:

Steps 1 & 2 are the same.

3. The tracker sends a reply to the peer or HAP with a – random or localized – list of peers that participate in that content file's swarm. If there is an interconnection agreement between the ISP and the OP, the IoPs/HAPs are surely included to that list. Since it is possible that a peer re-requests a list of peers during the downloading, if the IoPs/HAPs are not included in the first tracker's reply, they might be in the next ones.
4. The peer sends request messages to its neighbors in order to start downloading the file chunks according to the original protocol.
5. Chunk exchange follows the well-known unchoking and chunk-selection algorithms of the protocol until the peer downloads all file chunks. Due to the IoP's downlink/uplink bandwidth, the peer is expected to prefer the IoP to download from.
6. After the peer has a complete copy of the file, it either serves as a seed for that file, or leaves the swarm.

6.2 Intelligence and Decision-making

In order to optimize its own performance, the **ISP** needs to take some important decisions. These are explicitly described below:

1. Dimensioning of the IoPs/HAPs in terms of
   - Downlink/uplink bandwidth, *e.g.*, how many MBit/s of access bandwidth
   - Storage capacity, *e.g.*, how many TByte of storage
2. Number of IoPs/HAPs

   The ISP faces a trade-off here. More IoPs/HAPs imply improvement of performance but also increase of cost.

Both the two aforementioned decisions can be made by the ISPs, on the basis of either calculations derived by an analytical model *e.g.*, the Markov Model for the evaluation of BitTorrent swarms presented in Section 11, or of results derived by means of simulations. In particular, the Markov Model of Section 10 provides estimates of the download completion times based on the number and resources of the IoPs and can shed light to the relevant trade-offs. Furthermore, simulation experiments in the framework described in Section 12 are expected to show how both traffic, inter- as well as intra-domain, and completions times are affected by the number and dimensioning of IoPs.

3. Physical location of the IoPs

The ISP should decide based on the overlay traffic patterns in which physical locations IoPs should be deployed. There are many different choices here:

- One "large" IoP in a specific location (centralized approach): The ISP deploys an IoP – one IP address – with very high bandwidth and storage capacities. The problem here is that the IoP is a single point-of-failure; however it is easier to control content duplication.
- Multiple "smaller" IoPs in a specific location (less centralized approach): The ISP deploys multiple IoPs, with different IP addresses, with higher bandwidth and storage capacities than the regular peers but not as much as the previous case. The problem of the single point-of-failure is solved now, and it is still easy for the ISP to control content duplication. However, content is still found in a specific location of the physical network and perhaps that incurs increase of congestion in specific links in order to serve all peers in all different locations. Moreover, a performance tradeoff arises: the resources are split, thus reducing the multiplexing gain, but more ISP-owned entities participate in the torrent. Thus, the ISP should select the optimal number of IoPs.
- Multiple "smaller" IoPs in different locations (more decentralized approach): The ISP here deploys many IoPs in many different physical locations. This approach has the advantages of the previous one plus the fact that traffic is now more evenly distributed within the ISP's network. However, now extra overhead – even small – is produced for the meta-information exchange of the IoPs (with each other or the central entity) for the content duplication.

The decision-making in the IoP case is more flexible than the selection of the HAPs location which is already fixed.

4. Selection of HAPs based on activity and location

   The ISP should evaluate its peers collecting information about their behavior. The peers with the highest rank should be those that download a large amount of content in a constant basis and then stay in the swarm serving as seeds. Furthermore, the ISP should evaluate its peers based on their physical location, *e.g.,* the HAPs should be physically distributed along the whole network of the ISP.

   The above issues are related, but not identical to problems on cache dimensioning and placement. Related techniques and results from that field could be employed.

Additionally, the decisions on the dimensioning, number and physical location of the IoPs/HAPs, can be made based on a trial-and-error approach. By trial-and-error is meant that the ISP may originally assign some bandwidth and storage capacity to a certain number of IoPs placed in specific locations within the ISPs network, and monitor performance implications *e.g.*, traffic on inter-domain links. Based on the impact of its decisions the ISP can switch to a better allocation of capacity, reduction or increase of the number or placement of the IoPs in its network.

The IoP has also to make some serious decisions that are expected to have impact on its efficiency. First of all, it should decide on which content to download. The selection can be performed either in a centralized or distributed way. In the centralized case, it could be performed with or without human intervention. In the distributed case, it would probably be more efficient, if it was performed automatically. Clearly, the HAP will decide on which content to download according to the interests of the corresponding user.

In particular, the content selection could be employed based on the following approaches:

1. Swarm-size-based: The selection of content to be downloaded would greatly benefit from information provided by the Overlay Provider (OP), *e.g.,* trackers keep statistics about the number of peers that participate in each file's swarm.
2. Popularity-based: Again, selection of content can benefit from information by the OP or CP, *e.g.*, latest movies or software newer version releases. This approach is different to the previous swarm-size-based one, since a big swarm is a swarm for a popular file, but the opposite does not hold. For instance, a swarm for a new film might be small at the beginning and get bigger as more and more peers know about it. The idea behind the popularity-based approach is that the IoP could download a file before other peers start asking for it.
3. Trial-and-error: Alternatively, the IoP could join randomly selected swarms in popular trackers, monitor whether his intervention has the desired impact for the ISP and decide whether to maintain its position, and/or when to leave a swarm.

6.3 Possible Cooperation Schemes

The IoP/HAP insertion is neither an information exchange mechanism, nor a pure traffic management mechanism. Additionally, it does not require implementation of new architectural components, but only the insertion of new entities in the overlay.

The players interacting with each other when an IoP or HAP is inserted in the overlay are:

Internet Service Provider

The ISP wants to reduce both its interconnection costs as well as its network's maintenance costs.

- *IoP case:* The ISP deploys and controls the IoP.
- *HAP case:* The ISP does not deploy or control the HAP. However, it establishes a bilateral agreement/contract which ensures extra resources to the peer at no cost but requires the HAP to become/remain highly active. Moreover, the ISP can ensure that the extra resources offered to the HAP are dedicated to the specific peer-to-peer application, *e.g.,* by providing the end-user with preconfigured modem-routers that allocate specific portion of the bandwidth to the peer-to-peer application.

The ISP has also the necessary network information, *e.g.*, BGP info, RTT, or the number of hops. required when locality-awareness mechanisms are employed.

Overlay Provider: The OP wants to reduce its end-users' completion times and to improve their QoE. The OP has the necessary overlay information, *e.g.*, peer ids that participate in a swarm, number of peers per swarm, latest releases, or required for the overlay operation and the content selection. Furthermore, the OP controls the tracker. If cooperation of the tracker is needed, *e.g.,* in order to promote IoPs/HAPs, then bilateral agreement between ISP-OP should be established.

Content Provider: The Content Provider owns the content. The CP is interested in finding new ways to distribute its content. The revenues that would be attracted by an agreement with the ISP are a strong incentive for the CP to support this mechanism. In case of copyrighted content that is stored to the ISP's premises, bilateral agreement between ISP-CP is also required.

Highly Active Peers: They download a large amount of content at a constant basis incurring large amount of traffic in the ISP's network. Their activity can be exploited, if appropriate bilateral agreements are established with the ISP.

End-users: The end-users are unaware of IoP/HAP's existence unless the ISP advertises it.

Possible agreements that establish cooperation between these players are:

1. ISP – OP: Possible bilateral agreement ensures that the OP always favors the IoP instead of other peers so that the regular peers discover the IoP more quickly. On the other hand, the ISP could provide the OP with extra resources or advertise this OP's tracker.
2. ISP – CP: Possible bilateral agreement ensures that the CP gives its permission to the ISP to store licensed content to its premises and distribute it to its peers/customers while the ISP is charged for this right. Alternatively, extra resources could be provided to the CP (like in the previous case).
3. ISP – HAP: Possible bilateral agreement ensures that the HAP continues to download a large amount of content at a constant basis and serve as a seed while the ISP provides it with extra bandwidth at no extra cost
4. ISP – Peers: Possible bilateral agreements ensure that the IoP unchokes the specific peers while they are being charged for downloading copyrighted content from the IoP. In case of unlicensed content, no agreement can take place.
5. IoP – IoP: Cooperation is considered to be inherent, since all IoPs belong to the same ISP, *e.g.,* they are considered as multiple instance of the IoP deployed in ISP's network. Agreements are only required if cooperation between IoPs of different ISPs is employed.

No cooperation is considered between HAP – IoP or HAP – HAP.

Essentially, all previous agreements are optional. For instance, in the case of unlicensed content, the ISP does not need to establish agreements with CPs or peers. On the other hand, in case of licensed content, agreement with either CPs or HAPs suffices to overcome the legal issues that arise. Additionally, cooperation with OP is only needed to ensure quicker adoption of the IoP, however is optional, since the IoP will eventually end-up be preferred by peers due to its higher download/upload capacities.

6.4 Performance Optimization and Monetary Gains

The IoP is expected to achieve reduction of the ingress inter-domain traffic of its ISP. It also expected to increase the egress inter-domain traffic to other ISPs that do not deploy IoPs. On the other hand, the IoP is expected to achieve reduction of regular peers' completion times because it reduces delays and congestion in the inter-ISP link. Additionally, it should not introduce intra-domain congestion, and should not deteriorate the performance of other applications. To this end, IoPs should be introduced and dimensioned by the ISP with a global optimization perspective over all applications.

Monetary gains due to the performance optimization expected for all players are summarized below:

<u>ISP:</u> Because of the traffic pattern change, reduction of interconnection costs is possible. Of course, this depends also on the interconnection agreement between the ISP and its neighbor ISPs. It is possible that different charging schemes lead an ISP to different decisions w.r.t. IoP. If performance improvement is significant, then service differentiation is possible. In that case, downloading from/becoming unchoked by an IoP could incur a charge. Furthermore, price discrimination is also possible if different QoE-level agreements are established with the peers.

<u>Overlay Provider:</u> The OP can establish an agreement with an ISP, so that the OP promotes its IoPs when receiving requests from peers that belong to its domain for a charge. Alternatively, the OP can receive resources, *e.g.*, storage, bandwidth, or equipment, in lower prices or even for free by that ISP. Furthermore, service differentiation is possible. For instance, the OP can reply to a peer's request a random list of regular peers for free. On the other hand, it can also include to that list some IoPs but charge the end-user for this.

<u>Content Provider:</u> The CP can establish an agreement with an ISP, so that the ISP is permitted to store the CP's content in its premises for a charge. Alternatively, the OP can receive resources, *e.g.*, storage, bandwidth, or equipment, if needed, in lower prices or even for free by that ISP.

<u>HAP:</u> A peer that is selected as a possible HAP by the ISP has the opportunity to establish an agreement with the ISP that will provide this peer extra resources, *e.g.,* bandwidth and equipment, at no cost or at a lower price. The peer will in return continue to participate in as many swarms as possible even if he has finished downloading.

6.5 Implementation Issues and Qualitative Evaluation

The IoP as an ETM mechanism is application-aware, since it has to run the overlay application protocol to participate in the overlay network, and possibly swarm-aware, since it has to choose which swarms to participate in based on, *e.g.*, the content popularity or swarm size. On the other hand, the HAP is swarm-unaware since it participates only to the swarms that its end-user wishes.

The information needed by the IoP to evaluate the swarms is already available by the OP, *e.g.,* the tracker. Furthermore, after identifying the popular content, a mechanism is required that will automatically start acting according to the protocol, *e.g.*, download the torrent, send request to the tracker, negotiate with other peers. These two mechanisms, *e.g.*, content evaluation and initiation of downloading, can be quite easily implemented using existing hardware and only by developing some software. In the HAP case, none of these mechanisms is required; just the exchange of information between the HAP and the ISP and the means to vary the HAP's extra bandwidth.

Another important issue in the case of HAPs is that the ISP needs to ensure that the peer granted with extra resources (at no extra cost) will keep his part of the agreement. If no DPI techniques are employed then it is quite difficult to control what for the HAP uses its extra bandwidth. In order to avoid using DPI and violate Network Neutrality, the ISP could provide preconfigured modem-routers to its HAPs which would ensure that the extra bandwidth is dedicated only for the specific application, *e.g.,* allocation of 6Mbps out of 8Mbps to the port that is used by the overlay application.

Both inter- and intra-domain links have to be monitored. Traffic measurements on these links are required, in order to monitor the effectiveness of the approach. If ingress inter-domain traffic is high, more IoPs should be enabled. Otherwise, some of them could exit the swarm in order to minimize intra-domain traffic. However, the interconnection charging scheme has also to be considered. For instance, if further reduction of the inter-domain traffic has no impact on the interconnection charges, then the ISP should aim to reduce the intra-domain traffic in order to avoid congestion on its links by enabling smaller number of IoPs. In a medium timescale, *e.g.,* every 5 minutes, the ISP has to calculate the interconnection charge according to the interconnection agreement rule and decide how many IoPs are needed, or their dimensioning. In a longer timescale, the ISP should check the efficiency of the IoPs and revisit their dimensioning, *e.g.,* their storage capacity and bandwidth.

The idea of the IoP insertion is related to the insertion of caches by the ISP that store the content that is downloaded by peers, *e.g.,* [KRP05], [BCC+06]. However, the difference is that the caches solution needs to be combined with interception of peers' messages whereas the IoP is part of the overlay itself. That is, it runs the overlay protocol so no enforcement is required. The connection to the IoP is not enforced but optional for the regular peers. In this sense the insertion of the IoP is an innovative idea.

In the case of IoPs the content downloaded is stored in ISP's equipment. Thus, only licensed or non-copyrighted content can be downloaded by the ISP. Additionally, the ISP could establish agreements with Content Providers, *e.g.*, content distribution networks, software vendors, music industry, movies distributors, or TV channels. On the other hand, in the case of HAPs no licenses or agreements are required since the content is stored in the end-users' premises. Network Neutrality is not violated in any case, since neither message inception nor other DPI techniques are employed.

When licensed content is delivered by the IoPs, identification of the peers that will become unchoked is required, so that accounting can take part. However, identification of peers is not adequate to ensure that these peers won't upload the licensed content to non-authorized peers. Encryption or DRM could then be employed to prevent non-authorized peers 'experience'– watch, listen, read – the content even if they have managed to download it.

The IoP insertion is related to the Honey Pot architecture (see Section 7 of [D3.1]).

7 QoS/QoE-awareness Mechanisms

Different ETM mechanisms based on Quality-of-Service (QoS) and Quality-of-Experience (QoS) are proposed in this chapter:

- QoS incentives for Service Providers and End-Users providing guarantees to peer-to-peer overlay applications based on the Control Plane of the NGN (Next Generation Network) equipment are specified. Two main scenarios are considered here. In the first one the Carrier Class services are provided by the overlay service provider built on agreement with the ISP. The second scenario gives the opportunity of QoS guarantees assurance.
- Locality-based traffic shaping would enable assignment of different classes of upload bandwidth with respect to ISP-internal and remote connections.
- The VPN-assisted overlays approach assumes that for specific overlay applications dedicated VPNs would be established enabling for service differentiation.
- QoE-aware feedback mechanisms aim at predicting and reacting to possible QoE degradation.

7.1 QoS Incentive for Overlay Service Providers and End Users

The goal of this mechanism is to provide QoS guarantees (in terms of Network Performance capabilities) to peer-to-peer overlay applications. In order to do that, SmoohIT is focusing on the capabilities that are provided by the Control Plane of the NGN (Next Generation Networks) equipment [Y.2111], which according to the current specification provides interfaces that allow the dynamic enforcement of policies for specific flows or the configuration of the user profile (*e.g.*, allowing the usage of ISP services with dedicated bandwidth for Internet access).

Following these capabilities, the mechanism is focusing on two main scenarios:

- In the first scenario, the overlay service provider can provide carrier class services using ISP network capabilities. Therefore, the overlay service provider makes an agreement with the ISP to provide information about application traffic characteristics. The ISP configures its traffic management mechanisms in such a way that it can guarantee some QoS performance objectives to the application. With this mechanism, the ISP obtains benefit from third party applications, the Overlay Service Provider (as, *e.g.,* a peer-to-peer streaming based TV transmission) provides the service with more quality and can, *e.g.,* save some costs related to its servers; *e.g.,* the traffic coming from the servers will be prioritized among other traffic, thus leading to an improved service for the infrastructure available, or the ISP can also provide server installation facilities in its premises. At the end, the end 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.
- Secondly, following the SIS architecture defined in D3.1, the SmoothIT Information Service offers to the Overlay End Users not just only the capability to sort the list of peers but also the provisioning of QoS guarantees for specific connections. This would be an excellent option for VPN provisioning. This mechanism will be

integrated as part of the SIS centralized model as another service that can be provided to the end users.

The main advantage of this mechanism is that is based on the capabilities offered by commercial equipments, in such a way that the SIS will interact with the NGN Control Plane Capabilities as a Service Plane entity that will request the application of specific policies.

7.1.1 Details

While QoS-related incentives are clearly central to ETM, the provision of QoS always involves a variety of contractual and technical details. Next we discuss issues on the implementation of QoS for the overlay provider and for the user, including parameters of interest to each of them and other SLA issues.

7.1.1.1 QoS for Overlay Provider

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 IPs used by the servers, in order, *e.g.,* to allow the prioritization of the traffic from the Overlay Services. This option is relevant for overlay solution also supported by servers (as, it is usually the case of peer-to-peer streaming applications, such as Joost, as presented in D1.1), because 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), each CoS will provide its own network performance capabilities (in terms of IPLR, IPTD and IPDV), as it is specified in [Y.1541]

Following the SIS Architecture Design (see D3.1), this mechanism will be implemented using Admin Interface and QoS Manager modules. In particular, the Admin Interface must provide the capabilities to install the SLA between the Overlay Service Provider and the QoS Module must provide the capabilities to apply the QoS enforcement policies agreed in the negotiation.

In order to implement this solution, the following issues must 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 in terms of performance. 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 connection 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 constitute a scalability problem to the solution.

7.1.1.2 QoS for Overlay End Users

As stated before, in this scenario the End Users can request specific guarantees for specific connections. In order to implement this incentive, in the SIS centralized architecture, the end users will not just provide the list of peers to be sorted but they will also provide the QoS requirements for these connections. The SIS will send back the ranked list of peers, the QoS responses, and possibly the charges applicable to improved QoS.

When the SIS receives the request(s), it will interface 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 or 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, the end 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 and to get more bandwidth for Internet access by means of reducing the dedicated bandwidth for its ISP IPTV).

In initial tests with a SOAP implementation of an Rq ([Y.2111]) interface, the response time of the Control Plane is around 0.5s to configure the policies for a specific end user client. If this response time 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, which could pay an extra charge for this enhanced service by just reusing the NGN Control Plane capabilities that are being deployed in the different ISP networks.

7.1.2 Classification and Implementation

The approach presented introduces new capabilities to the SIS system that can be introduced progressively: the SIS just needs a new component, the QoS Manager, which will be in charge of interfacing the NGN Control Plane capabilities.

7.1.3 Qualitative Evaluation

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

- The Overlay Service Provider can provide carrier class services by cooperating with the ISP that can be implemented by just reusing the NGN Control Plane capabilities. Moreover, the Overlay Service Provider could get economical benefit from third party applications.
- In this approach, the peers will have more incentives, in terms of performance to follow the SIS suggestions.

- If the user requests QoS guarantees, then he can take advantage of enhanced network capabilities and the ISP can obtain extra revenues for this usage.

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.

7.2 Locality-based Traffic Shaping

Different rates of upload bandwidth are assigned to high-bandwidth users by ISP using the *locality classes*: ISP-internal and remote upload bandwidth. For typical ISPs (DSL connections) the remote bandwidth will be only a fraction of the ISP-internal bandwidth. The ISP equipment (*e.g.,* router at the access link) enforces that only the allowed amount of bandwidth is used per class. The destination IP address is used to differentiate ISP-internal and remote connections. Therefore, the users are incited to keep their *upload* traffic inside of the ISP domain. Note that the policy should be part of the contract between ISP and customers. Additional class of bandwidth could be offered for uploads designated for ISPs with peering agreements. The overlay can either measure the available upload bandwidth (inexact) or query the SIS entity to obtain the class of the anticipated connection and the bandwidth available for each class.

7.2.1 Details

The following information is required as input for this mechanism:

- Locality classes: own ISP, peering ISP, other ISP. This information must be provided by the ISP and is used by the traffic shaping devices (routers) to enforce the amount of utilized bandwidth. Furthermore, the overlay can use this information to select local peers with high probability. Note that there are three alternatives how the overlay can deal with traffic shaping:
  - Built-in overlay self-optimization mechanisms will react on the higher internal bandwidth and prefer local connections.
  - Locality-aware peer selection can be done by tracker (similar to Subsection 8.3.1 and the approach proposed by Bindal et al [BCC+06]).
  - SIS can be used to obtain ISP-related information.

  So this approach can be combined with Subsection 8.3.1 and SIS service if it turns out that the self-optimization mechanisms are not enough.
- Peering policy is used internally by the ISP to define locality classes.
- Maximum amount of bandwidth available for each locality class (from ISP).
- Destination IP addresses are mapped to locality classes. This mapping can be a simple table that maps IP prefixes to classes. Its size is relevant for efficient implementation in traffic shapers.
- IP addresses of remote peers are provided by peers that want to upload data. This can be either done either on regular basis upon a download request or (if based on IP prefixes or AS-IDs instead of IP addresses) in advance. In the latter case peers will have the same copy of locality classes used by traffic shapers. Note that there are two possibilities to obtain locality-class information from SIS:

- Contact SIS each time you have to select a new neighbor based on IP addresses of candidate peers (introduce additional delay). The result would be the mapping IP address→ locality class
- Obtain the mapping AS-ID → locality class from the SIS only once and then the tracker attaches AS-ID to peer addresses requested by peers. Based on the AS-ID and the locally known mapping rules the locality class can be determined by peers offline.

The basic interaction assumes that the peer does not know the mapping "IP prefix → locality class". Therefore:

1. Each peer requests the SIS for locality classes and available bandwidth per class (static information)
2. A peer receives a connection request (including IP address) from another peer (this can be also done periodically for a list of intermediate requests)
3. Only if the locality class for this address is unknown:
   a. The peer forwards the address(es) to the SIS
   b. The SIS returns the locality class(es)
4. The peer inserts the new request into it's upload queue using the locality class as priority parameter. Note that there is no shaping at the user device. However, peers will benefit if they try to communicate with local peers.
5. The ISP infrastructure at the access link (QoS-aware routers) enforces that only the allowed bandwidth is used.

The whole knowledge about the locality of nodes, the assignment of nodes to locality classes and the bandwidth restrictions per class are located at the ISP side. SIS is responsible to expose this information to peers. The internal ISP infrastructure (that could possibly be considered as part of SIS too) defines the locality classes, available bandwidth, and mappings from IP prefixes to locality classes.

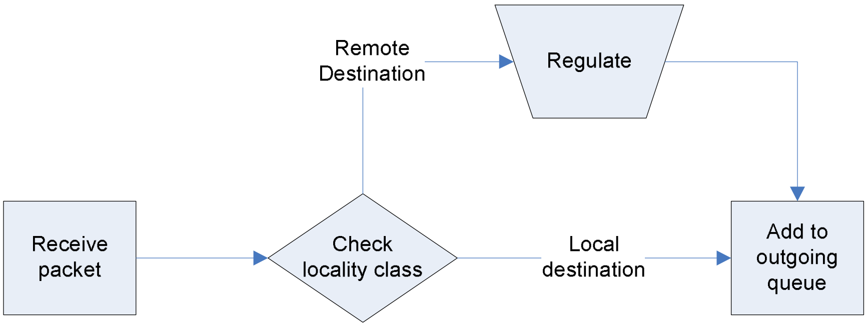


Figure7.1: Locality-aware Traffic-Shaping.

7.2.2 Classification

This approach can be classified as a pure traffic management mechanism. Further on, it requires a new architectural component: traffic shaping devices that handle the upload traffic of peers differently. In most cases, existing equipment can be reconfigured to provide this functionality. This mechanism is mostly relevant for high-bandwidth users, *e.g.,* high-speed DSL, cable and fiber-optic connections that typically anyway require sophisticated equipment at the ISP side with QoS-support. There will be however a tradeoff between the cost of implementation and the benefits in terms of costs (for the ISP, due to traffic reduction) and performance (for the end users), but it is expected that the benefits will be higher than the costs. This issue will be further investigated.

The mechanism further requires information exchange between ISP (SIS server) and overlay (SIS client).

Mechanism class	**Classification**
Information Exchange Mechanism	Yes
Pure Traffic Management Mechanisms	Yes
New Architectural Component	Yes / QoS-aware routers -> extend existing routers and the SIS component

7.2.3 Qualitative Evaluation

Besides ISP and overlay there are no further players involved in this mechanism. By supporting different bandwidths for different communication partners the ISP offers a *locality incentive* to the overlay. The latter can benefit by being locality-aware, however, it can also route traffic across domain borders if required, but with a lower speed. In fact cooperation with the overlay is not essential for this mechanism to work since the bandwidth limits are enforced by the ISP. Nevertheless, the **overlay can** avoid a “trial-and-error” behavior to find good connections and immediately communicate with local peers mostly.

The mechanism allows ISPs to save money in the following manner:

Currently ISPs typically offer users flat rate contracts with a fixed upload bandwidth. If they offer too little upload bandwidth the customers will be unable to use such applications as peer-to-peer Video-on-Demand. Therefore, they might be unwilling to pay for expensive high-speed connections even if they receive a lot of download bandwidth. On the other hand if there is too much upload bandwidth the customers might use peer-to-peer applications too extensively. This can result in high transit costs for the ISP.

This especially applies to Tier 2 and Tier 3 ISPs. By limiting the external upload bandwidth more strictly the ISP can save costs while still attracting peer-to-peer users. Of course, the potential for savings also depends on the agreement between the ISPs as well as on whether the other ISP also applies some locality-aware policy. The core point is to show that a proper-designed locality-aware peer-to-peer application can offer the same quality to the user even with a lower inter-domain upload rate.

This mechanism requires configurable routers with QoS support as already discussed above. **Error! Reference source not found.**7.1 shows the basic control flow in the router in order to provide the required functionality. Modern Cisco routers and Linux-based routers offer this. No application awareness is required, since we expect this mechanism to affect only upload-intensive application.

The maximum upload rate assignment is done in a static way, only the measurement and enforcement of the utilized (external) bandwidth is done continuously. This can be done by applying a leaky-bucket regulator to the router queue. The most relevant parameter is the bandwidth ratio allocated to single classes (ISP-local = max. available, peering ISPs and other ISPs reduced). It has to be analyzed by simulations which mix of bandwidth ratios is efficient to reduce inter-domain traffic while preserving good QoE for the users.

It is important to mention that typical ISPs already apply traffic shaping methods to save costs, avoid bottlenecks or prioritize delay/bandwidth sensitive applications. However, they do not announce their policy to applications and therefore applications (such as peer-to-peer overlays) cannot react in an appropriate way.

The fact, that this approach is content and application unaware makes it easy to implement and avoids most of net neutrality issues. The only concern might be the different treatment of packages based on their destinations. But this is something very common for traffic engineering methods since ISPs can route packets according to their own cost and performance metrics anyway, i.e. they don't have to use the shortest path to another domain but could use a slower path through a cheaper exit.

There are no specific privacy or security issues for this mechanism. Finally, regarding the Architectural Design Space of D3.1, Section 7, this mechanism fits to the Control Freak scenario because it enforces that most of the uploads are done locally.

7.3 VPN-assisted Overlays

This mechanism can act as a wrapper of other ETM mechanisms. The main idea is that a VPN is formed that offers higher performance for a specific overlay application. This VPN may span across many domains. This can be seen as a premium service, where peers join if they chose to pay some extra money.

7.3.1 Description

Inside the VPN there may or may not exist a variety of mechanisms that employ ETM techniques. For simplicity reasons, we refer to all the possible mechanisms under the term "SIS". Hence, the SIS along with mechanisms providing some QoS, similar to the mechanism proposed in Subsection 7.1, is deployed inside the VPN. Users in the VPN are assumed to be willing to follow any rules deployed in order to indeed enjoy an enhanced service, since they have paid for it. ISPs can place boxes that, *e.g.*, police traffic or check violation of interconnection agreements.

In order to deploy a VPN, the IP addresses of the VPN users are required. The IP address of each VPN-user is acquired by the ISP/VPN-provider. Furthermore, the registration of the peer is performed also the first time that it enters the VPN, probably to a centralized entity, so as to be able to join the VPN the next times as well. We expect that this operation is trivial. The centralized entity can be a server owned by the ISP/VPN-provider – that keeps track of each peer/VPN-user's activity – *e.g.*, when he is logged in, accounting information, or traffic sent/received to/from the VPN. Based on this information, the peer is also charged for the use of the VPN/premium service. Last but not least, any other information required by the SIS is also needed, if such ETM mechanisms are also employed. Besides the credentials provided by the ISP, the end-user can be provided with statistics of its activity as well as billing information, if a pay-per-view approach or some other method of charging is employed. Of course, if SIS is deployed, any information generated by it, *e.g.,* peer ranking, is provided to the VPN-user.

The event flow for both premium and best-effort service is presented:

<u>Initial phase:</u>

The ISP advertises two types of services per overlay application: a premium charged one, the VPN and a best-effort but without charge.

1. The peer decides whether or not to join the VPN of the overlay application that it wishes to use:
    a. Pays extra to the ISP/VPN-provider but has increased performance
    b. Or pays nothing and uses the regular overlay

Premium service scenario:

2. Registration of the user to the centralized VPN component, acquiring credentials.
3. A filter/special front-end application is installed to the user's machine.
4. Login to the VPN using the credentials.
5. Overlay query/response messages are routed through the SIS.
6. SIS's suggestions are adopted by the user – "enforced" by the front-end application

Best-effort service scenario:

2. The peer's traffic is treated by the default overlay protocol.
3. The ISP treats this as best-effort traffic.

7.3.2 Intelligence and Decision-making

The ***ISP*** needs to dimension its network so that it can guarantee that the VPN-service is indeed superior to the best-effort one. Scalability is very important, since participation of new users must not degrade the performance of the existing ones.

If no other ETM mechanisms are employed in combination with the VPN, then regular VPN Traffic Management is required, *e.g.,* components that help the VPN functionality. That is VPN traffic passes over dedicated links. Furthermore, application-awareness is required in the sense that the ISP deploys a unique VPN, however it employs different rules and behavior per application, *e.g.,* different time constraints for file-sharing or streaming applications.

The ***peer*** has to decide whether it is beneficial for him to pay for the premium service. If performance improvements are insignificant or the peer has a 'low' – in terms of volume of downloaded/uploaded data – profile, then the best-effort service is rather sufficient.

The VPN peer runs a front-end application which is a modified version of the overlay application client. To do so, a gateway should be deployed inside the VPN, allowing the communication with the rest if the Internet. The client can participate simultaneously in both VPN and public data exchange. In order to login to the VPN the client logins using the user's credentials. Based on the content availability, *e.g.,* number of peers and seeds that participate in one file's swarm, and other possible metrics such as estimated delay or congestion or estimated time elapsed (information deduced by the SIS), the client selects which swarm to participate in. It is assumed here that the torrent file, chunks number, and ids are the same for a specific file whether that file is distributed within or outside the VPN.

The ***SIS*** has to decide the algorithms that it employs for the VPN-traffic management and traffic shaping/engineering as well as the information required as input to these algorithms. The SIS is different for each different kind of overlay application and has to decide the optimization criteria and the algorithms employed for the VPN-traffic management, as well as the information required as input to these algorithms.

The VPN is not an information exchange mechanism. It is a pure traffic management mechanism that comprises new architectural components (the VPN components).

The players involved are:

ISP: The ISP belongs to the alliance of the ISPs that form the inter-domain VPN network. The VPN lies on top of the ISPs' physical network. In fact, the VPN is a different overlay network but for the same overlay application. It also performs the VPN-traffic management.

The ISP has an incentive to deploy a VPN since this way he can better monitor and manage the peer-to-peer traffic and furthermore he can charge it. Up to now, the ISP had to serve this traffic without getting revenue for this service. Additionally, the ISP can increase its revenue by selling value-added services through the VPN network, *e.g.,* IPTV.

Peers/VPN-users: They are regular peers that can choose either to use the premium service of the VPN or the regular best-effort service of the (original) overlay. The VPN-users experience superior performance to the non-VPN, best-effort peers. Of course, measurable performance improvements are expected here. Moreover, if SIS is also deployed, the VPN-users are expected to follow SIS suggestions/decisions since performing differently implies performance degradation. Actually, SIS decisions would be rather enforced by the front-end application running at the peers' side.

Overlay Provider: The OP can possibly provide special versions of application clients capable to run inside the VPN, especially if communication with non VPN peers is desired. The OP has the incentive to establish agreement with the ISP, if they share the income from the premium service.

Content Provider: The CP can also establish agreement with the ISP, so that this content is stored in physical locations from which it can be accessed by the VPN-users as well. Agreement between ISP-CP is considered very important since it is expected that users would have a strong incentive to pay for the premium service, if they could also get access to licensed content besides the improved performance offered. For instance, IPTV (LiveTV and VoD) could be offered by the ISP through the peer-to-peer VPN, instead of more expensive centralized architectures. Furthermore, a VPN per subject section could be deployed, *e.g.,* a sports-VPN with channels broadcasting sport events, available video content of former sport events, and other related content such game results or program of upcoming games, The CP has interest to establish agreement with the ISP so that the VPN-users have access to its content, if they share revenues coming from the premium service.

Possible cooperation agreements established between the players are:

- ISP – ISP: Cooperation between ISPs is needed in order to form a VPN spanning across many domains. If such cooperation is not accomplished, then each ISP actually deploys a walled-garden.
- ISP – Overlay Provider: Cooperation between ISP and OP is also important so that VPN-users can communicate with the non-VPN peers. Otherwise, content availability issues may arise.
- ISP – Content Provider: Agreement between the ISP and CPs can be established so that CPs' content is accessible by the ISP's VPN.

- Peers/VPN-users: Peers entering the VPN actually agree to cooperate with ISPs. In fact, once inside the VPN, the peers are "enforced" to follow ISP's decisions.

7.3.3 Optimization

The ***ISP*** is able to better manage the traffic passing through its network. In particular, deploying a VPN would have as a result less congested links in the ISP's network which implies maintenance cost reduction. Furthermore, the ISP charges the VPN-users for the premium service offered and therefore gets revenue for traffic that previously passed uncharged though its network. Additionally, value-added services like IPTV can be offered through the VPN to the premium users and bring extra revenues to the ISP.

The ***OP*** establishing interconnection with the ISP's overlay, *e.g.,* the VPN, achieves improvement of the VPN-peers that use the premium service, which eventually are also customers of the OP.

The ***CP*** establishing agreement with the ISP manages to make its content available through a new distribution network. An agreement between ISP-CP implies split of revenue and increase of the CP's income.

7.3.4 Qualitative Evaluation and Implementation

The intelligence part is rather demanding. The VPN creation and operation requires effort from the ISPs. Existing and possibly new network components will be used. For instance, traffic shapers at edges, policy and congestion monitors inside the VPN are required. The VPN traffic management is already performed by the ISPs for VPNs offered as services to business customers. However, the interconnection of ISPs in order to deploy single, spanning different domains VPN is more challenging.

Two questions arise here: Do we need one application offering jointly the VPN and overlay functionality or one generic VPN application (with the basic and enhanced functionality) and several modified overlay applications? In the first case, we can run only one such client at a time, since we cannot connect to more than one VPNs at the same time, while in the second case the peer registering by the VPN needs to state its preferences (file-sharing vs. streaming).

The VPN traffic management runs at a short time-scale as it would run in any other VPN network deployed by the ISP for a business customer. The SIS functionality, if employed, runs at a longer time-scale. Furthermore, in a quite large time-scale the ISP must re-dimension its own VPN network and offer more resources if the VPN-users' number has increased.

The ISP has to dimension the VPN accordingly to the number of the VPN-users. This can be performed by measurements of delay/congestion on the dedicated VPN links. If the delay/congestion overcomes a specific threshold, then increase of the dedicated links' capacity is required in order to guarantee the VPN-users' improved performance. Additionally, parameters required for the SIS functionality have also to be selected (see Subsection 5.1).

The VPN approach is quite innovative since it combines QoS improvements and offers differentiation through premium services, while it can take advantage of the innovations at the SIS.

The VPN functionality rises no legal or NN issues, as long as non-VPN traffic is not degraded, that is it remains best-effort rather that less-than-best-effort. Additionally, any other issues related to the SIS functionality have also to be considered (see Subsection 5.1).

Important privacy and security issues arise with the VPN implementation. First of all, each VPN-user needs to have a unique username and a password in order to login to the VPN. Identification of the VPN-user is required in order to treat its traffic towards other VPN-user as premium one, in contradiction with the traffic to/from non-VPN users which is treated as best-effort one. Additionally, licensed content that might be available to the VPN-users due to agreements of the ISP with CPs, should not be accessible by non-VPN users, or if it is accessed – the ISP cannot control the VPN-users which might upload the licensed content to unauthorized peers – it should be protected by DRM or encryption.

The VPN is related, but not identical to the Control Freak architecture (see Section 7 of D3.1).

7.4 QoE-aware Feedback Mechanism

SmoothIT aims at achieving a TripleWin situation. Thus, ETM mechanism shall improve inherently the user perceived quality while reducing costs for ISPs. The proposed QoE-aware feedback mechanism describes a basic concept how to control and improve the QoE of a user by taking into account feedback about the current QoE. To estimate the QoE within the network, appropriate QoS parameters have to be monitored and passed to a QoE metric function. The obtained QoE values are then used to perform particular actions, e.g. by modifying resource reservations to overcome decreasing QoE.

The purpose of the mechanism is to predict the danger of QoE degradation and react in advance. QoS parameters that QoE is sensitive to are monitored and the results are passed to a QoE metric. By means of this metric, the actual QoE value is calculated. If the value of the metric is below a predefined threshold, then the ETM system is intended to react and prevent the QoE degradation by modifying resource reservation, routing in the underlay and peer list (ranking) or changing parameters of QoS service differentiation for individual flows or peers. The QoE metric is not intended to give a very precise MOS value expressing the objective quality. It should give a good approximation to make appropriate decision regarding the reaction to the upcoming degradation of perceived quality. The key objective is that the system reacts before the user reacts and aborts the service.

The history of the QoE estimations can be used in decision process (regarding the selection of peers and resources) in the future. The system may learn from the past system performance. It especially adheres to the decrease of start-up delay for future requests for the same content.

The mechanism requires the following input information:

- Current values of QoS parameters: IPLR, IPTD, IPDV. To be provided as a result of active, timely measurements.
- Measurements of start-up delay (in case of streaming applications). This can be provided (probably) only by the peer. As this is done at the user side, a mechanism verifying the credibility of the information has to be applied in order to avoid that a malicious user utilizes the QoE control mechanism to always get a better performance by steadily complaining about its current startup delay.

- A user may additionally indicate if its user perceived performance is bad. This can be used a) to detect cheating users and b) to have a subjective, supplementary QoE indicator beside the objective measured and mapped value. It again raises the issue of credibility of such information.
- Information about chunk availability and upcoming danger of chunk starvation resulting in video freeze.

The latter two types of information can be provided to the mechanism if a peer software modification is possible. Otherwise, the source of information is void and this functionality cannot be implemented. The function mapping the possibility of chunk starvation to QoE value could be implemented in a peer (application software) or in SIS. In the latter case, the peer notifies the SIS about current state of buffer map and after checking the credibility of the information SIS calculates the QoE metric.

As a result the mechanism provides at least the following information to be utilized for optimization and avoidance of QoE degradation:

- QoE metric (value indicating the level of QoE and its upcoming degradation)
- Modified list of peers
- Modified resource reservation or QoS service differentiation, *e.g.,* increase capacity of individual peers if possible

The algorithm of the QoE aware mechanism is to be developed. However, the potential order of events and messages in the system could be initially as follows:

1. Measurement of parameters influencing QoE is performed:
    a. The measurement of IPLR, IPTD, IPDV is performed in the network (Metering Component of SIS) and provided to the QoE metric
    b. Peer measures the start-up delay and notifies the SIS. SIS can use this information for future decisions
    c. Peer monitors the availability of chunks in the buffer in relation to the current playback point. Peer timely notifies the SIS about its buffer map, or sends the estimation of QoE metric related to chunk starvation probability if it is below a predefined threshold and requests a modified peer list
2. The QoE indicator value(s) is calculated using the QoE metric,
3. If the QoE degradation is predicted (QoE indicator value is below a predefined threshold) take the action:
    a. modify the resource reservation
    b. send a peer a modified list of peers having desired chunks

The mechanism is to be implemented in SIS server, particularly in QoS Manager and Metering Component.

Optimization can be performed with respect to threshold levels, minimization of the probability of QoE degradation. The suggestion from SIS on how the peer should behave in order to avoid quality degradation takes into account ISP's strategy regarding modification of resource reservation, as well a reordering of peer lists. In combination with locality information, the list reordering can serve to the user's and the ISP's targets.

7.4.1 Classification

QoE-aware Feedback Mechanism types can be classified as follows:

- Information Exchange Mechanism. Information from the end user, like startup time, and the network, like IPDV, is combined.
- Pure Traffic Management Mechanisms. In case of QoE degradation, one option is to use pure traffic management mechanisms, e.g. modifying resource reservation or parameters of QoS service differentiation.
- New Architectural Component. The mechanism can constitute a separate element in the architecture or it can use available components of SIS, especially: QoS manager, metering component; only new functionality is added, i.e. QoE mapping function and triggering of reservation mechanisms/QoS service differentiation; metric for sorting peer list is changed.

7.4.2 Qualitative Evaluation

In the basic solution two players are involved: ISP and end-user peer. Solution not involving peers is possible and easier to develop but it would not be fully functional. On the other hand if the peers are involved in providing SIS with some information the solution become more complex and difficult to develop and would require a dedicated credibility mechanism.

The incentive for peers would be improved quality. If the peer follows SIS suggestions it would experience a better QoE than if it relied on intrinsic peer-to-peer application mechanism. Better resource management due to dynamic reaction to possibility of QoE degradation would also be an incentive to the peer.

In order to effectively (*e.g.*, by means of peer-perceived quality and ISP’s goals, including traffic locality, or cost-effectiveness) react to the upcoming degradation of QoE, peers should communicate with SIS and follow its suggestions. It should be decided which players should have an established cooperation between themselves. The peer’s incentive is an improved QoE. The ISP’s incentives are cost reduction and improved QoE for its customers. This means that there will be cooperation between peer and SIS (ISP).

This criterion should answer the question in which cases is the ETM approach expected to perform better. As degradation of QoS is not always directly related to user perceived quality, the reservation of resources or any other mechanisms can be more effectively used; *e.g.,* an increase of packet loss of 1% is dramatic from 0% to 1%, but causes no noticeable problem from 3%-4% as the quality is already around say 3 on a MOS scale; thus, in the first case, the mechanism should react, in the second case it is not a critical issue; thus, resources and therefore money can be saved.

When considering the effort required the feasibility of function capable of mapping QoS parameters such as IPLR, IPTD, IPDV to QoE values should be evaluated. A mapping function of such parameters as start-up delay and playback freezing probability should be found. The mapping function would become more complex when multiple criteria are taken into account. Implementation of a feedback system and measurement tools (including modification of the peer software) would be necessary in order to evaluate and improve the efficiency of the mechanism. Before its implementation the estimation of how realistic the approach is should be done.

It should be decided with what frequency the periodic measurements of parameters such as, IPLR, IPTD, IPDV, should be done. Chunk availability could be measured on periodic basis or only if it is necessary (*e.g.,* upcoming danger of chunk starvation). Start-up delay can be applied after the actual start of the playback.

If the strict Network Neutrality is considered there might be a problem with the QoS service differentiation as well as resource reservation for flows/peers that might be modified by the proposed mechanism. There are no legality or NN concerns if the mechanism updates the list of peers taking into account QoE.

Since partial information to the feedback mechanism is provided by peer software, there is a potential danger that peers would try to cheat the ETM/SIS mechanisms. Misleading information can be used as a form of attack. The QoS measurements are performed in the network by ISP-owned mechanisms. Thus, there is no concern about the QoE indicators based on pure QoS parameters provided to the feedback mechanism.

The Honey Pot mechanism can be implemented if the mapping function involves only QoS parameters such as IPLR, IPTD, IPDV. The honey pot architecture assumes no modification to peer client software, so monitoring of start-up delay and chunk availability/starvation is not possible. Without the modification of the peer's software it could be also not possible to provide a peer with a modified list of peers (not requested by the peer). Implementation of planned functionality seems to be possible for the case of Control Freak and Optimal Anarchy mechanisms.

The considered concept is quite new, as it focuses on QoE. There are several approaches and frameworks in literature trying to maximize QoS, like IntServ [RFC1633], DiffServ [RFC2475], or Next Generation Networks [ITUY2001]. Also several research projects aims at implementing QoS mechanisms and maximizing their efficiency, e.g. [EuQoS]. However, QoE is much more complex than QoS, as QoE links user perception and expectations on one side and technical QoS parameters, management, pricing schemes etc. on the other side. The scientific field about QoE is at its early stages and new communities around QoE issues are currently built as indicated for example by the ITC Specialist Seminar on QoE [ITCSS18].

8 Other ETM Approaches

The ETM approaches presented so far have some characteristics that allow their classification. Thus, we have seen centralized approaches that try to improve the performance and decrease the costs, using either overlay metrics and tools (cf. Section 5) or focus on techniques dealing with Traffic Management and QoS (cf. Section 7). We have also seen an approach that tries to influence traffic in a transparent for the overlay way (cf. Section 6). In this section we present some approaches that do not strictly belong to one class or another, but can be seen as enhancements or extensions of the approaches already described or as approaches that have the same objectives with the previous ones, but achieve them in a different way.

In particular, we present here:

- A distributed exchange of overlay information and peer evaluations that can help optimize the decisions of each peer in a decentralized way
- An content-centric approach that suggests the peer which swarms to join, based on the content he requests
- Three enhancements of the overlay protocol that introduce
    - A mechanism for peer selection that considers the structure of the Internet
    - An incentive mechanism that can extend the tit-for-tat algorithm
    - An incentive mechanism that takes advantage of “old” content in the tit-for-tat procedure

8.1 Distributed Exchange of Peer Lists

In this approach, we consider a peer-to-peer environment where peers are allowed to exchange information that may be useful for enhanced peer selection strategies or unchoking heuristics. The purpose of this approach is to provide peers with information resulting in the introduction of some level of locality awareness to the overlay, in a way that is beneficial to these peers. In particular, this category of ETM mechanisms allows peers to share the knowledge gained due to their participation in the chunk exchange procedure. After obtaining a chunk, the downloading peer does a personal evaluation of the perceived service, allowing him to characterize the sending peer as “good” or not, and use this experience in the future so as to optimize his peer selection or chunk selection strategy. Moreover, the mechanism we describe allows peers to exchange personal evaluations, since this knowledge can be beneficial for others too, particularly since it is augmented with the relevant knowledge of others. In addition, a light version of the IoP (see Section 6) is introduced, where the IoP does not offer any content to the peers of a swarm, but provides instead his evaluation for other peers with respect for how good they can be based on underlay information. The IoP offers this information only to the local peers. In turn, the IoP will not be evaluated for the exchange of chunks (the respective value could remain at a default value), but for the accuracy of evaluations he gives. IoP’s evaluation however is not based on personal experience from chunk downloads, but on underlay information that the IoP has access to, thus eliminating any legal concerns. Hence, by the distributed exchange of personal evaluations and the introduction of the IoP a hybrid mechanism is designed, where decisions are made in a distributed manner, based on information provided by peers and by a central entity, i.e. the IoP.

8.1.1 Description of the Mechanism

As already mentioned, after obtaining a chunk, the downloading peer evaluates the transaction. Obviously, the evaluation will take into consideration the time it took the download to complete and the throughput. In other words, the evaluation will be based on measurements related to overlay performance metrics. Using this evaluation, the peer is able to assign a value/weight to each peer he has downloaded chunks from. More specifically, he assigns values/weights to every peer in the list of peers received by the tracker. Hence, the list is ranked, according to the experience of the peer. The respective weights are then periodically communicated to all the peers of the list and/or to peers with whom such information was exchanged before. The recipient-peers take into consideration the weights provided by the sender-peers and adjust their personal ranking. After a chunk transaction is complete, the downloading peer assesses the outcome in relation with the rank value of the uploading peer. If the outcome is contradictory to that expected due to the rank value, then the downloading peer lowers the value/weight of the uploading peer (which results in falling in a lower position of the ranking) as well as the credibility value of the peers that suggested this uploading peer. On the contrary, if the outcome is in accordance to the rankings received, the peer increases the credibility value of the other peers that sent such rankings. Following this, a new ranking is generated and the new weights are communicated to the relevant peers.

The ranked list can be used in various ways by the peer. Since it is based indirectly on “proximity” information, the list could help the peer to decide which peers to try to connect to first, or which peers to unchoke more often than others in order to be reciprocally unchoked by them. With the approach described, we construct a self-organizing mechanism with respect to peer selection and/or unchoking criteria. To enhance this approach and to have the ISP *intervene* in the peer selection and benefit accordingly, an ISP-owned peer (IoP) should be inserted in the overlay network. This peer would only participate in the distributed exchange of ranked lists. Contrary to the IoP-based approach of Section 6, this version of the IoP would not exchange content chunks with the regular peers. Thus, the ranking criteria of the IoP would not be based on the outcome of overlay transactions, but rather on underlay information related to each peer in the list. Such underlay information is acquired from an SIS module that resides in the IoP and interfaces with underlay network operated by the ISP, so as to get BGP and any other available locality information (see Subsection 5.1 for more information). Hence, the information circulating in the overlay is enriched with proximity information provided by the underlay through the IoP-SIS component.

The proposed mechanism requires the following input:

- List of peers from tracker
- Ranked lists from other peers of the same swarm. The fact that one (or more) peers may be IoP is transparent to the regular peers.
- The BGP information provided by the underlay through the IoP-SIS component

The expected output of the mechanism is the following:

- Ranking of peers included in the list of peers
- Credibility values for peers exchanging lists

Below, we provide the flow of events that summarize the proposed mechanism:

1. A peer receives a peer list from the tracker
2. The peer starts downloading chunks from peers in the list
3. The peer evaluates the outcome of the chunk transactions, based on some overlay performance metrics (for normal peers), *e.g.,* download time.
4. The peer ranks the list of peers according to its own evaluations. In particular, the peer
   a. Updates the ranking and weights, based on the outcome
   b. Updates the credibility values of the peers that suggested the involved peer, based on the outcome
5. The peer sends the weights to the peers which he communicates with. The IoP might be included in those peers.
6. The IoP evaluates the peers of the list he receives, based on both overlay and underlay metrics, *e.g.,* downloading time and utilization of inter-ISP link employed. It is important to note that the ranked list is not the same for every peer. The ranking also depends on which is the sending peer and the IoP has to evaluate each candidate peer on a pair basis. The list of peers is obtained
   a. from other peers that communicate their owned ranked lists in the distributed manner already described. In this case the size of the list is limited to the number of "neighbor" peers the sending-peer has
   b. directly from the tracker. In this case the list is much larger and includes almost all peers of the swarm. This assumes cooperation between the overlay provider (tracker) and the ISP (IoP)
   c. both from the tracker and from other peers. The list from the tracker is used for bootstrapping and in order for other peers to come in contact with the IoP. When peers start communicating *ad hoc* with the IoP, then the IoP needs only to rank the peers in the list offered by the sending peer.
7. When a ranked list and the corresponding weights are received, the peer takes into consideration the ranking provided along with the credibility value assigned to the peer that sent the list, which serves as a weight.
   a. unknown peers have an intermediate default credibility value, say 0.5, (with 0 being the least credible and 1 the most credible)

The intelligence of the mechanism is located both at the peer-to-peer application (distributed part) and the IoP-SIS entity (centralized part). At the peer-to-peer application three different modules exist: a Evaluation Module, a Credibility Module and a History Module. The ***Evaluation Module*** provides the values/weights and the ranking for the list of peers, depending on the outcome of the chunk transfers. As already mentioned, the algorithm in place must consider a set of overlay metrics. The ***Credibility Module*** keeps and updates the credibility values and the ***History Module*** keeps track of the suggestions other peers offered in order to update the credibility values. It should be noted that these two modules are not swarm-specific. At the IoP-SIS entity, there also exists an ***Evaluation Module*** which provides ranking considering both overlay and underlay metrics. A desired characteristic of the ranking taking place at the IoP-SIS entity is that no two peers should receive the same weights in a ranked list (even if the peers in the list are the same for both), unless they belong to the same "neighborhood" of the domain. This should be done in order to help avoiding flash-crowd events. Apart from the fact that there are many peers in the system, we can have more than one IoP-SIS entity in a single AS domain. In fact, one of the ISP's main concerns is the decision of how many such entities to place in his domain in order to achieve the desired result at the minimum cost and side-effects.

8.1.2 Classification, Architectural Design, and Implementation

The aforementioned approach introduces a new information exchange mechanism as well as a new architectural component, the IoP-SIS entity. Since distributed information exchange is involved, none of the existing options in the design space described in Section 7 of D3.1 is appropriate to implement it, except for the IoP-SIS component that is similar to the Honey Pot solution. In fact, for such a mechanism to be implemented, some changes to the client application must be introduced in order to include the three basic modules.

With respect to the effort required to implement this approach, there are some points that need consideration. First, the exchange of peer lists and weights should be realized. We assume that the lists are obtained both from the tracker and the peers. If peers do not participate in the exchange of lists under the original peer-to-peer protocol or if we have the case of a tracker-less system, then a peer-to-peer protocol for the exchange of lists should be implemented. Apart from the exchange of peer lists, the weight should be also included in the messages exchanged. As already mentioned, the intelligence of the system will be the algorithm that evaluates the outcome of the downloading of a chunk based on overlay metrics, the algorithm that keeps and updates the credibility values and the algorithm for updating the rankings based on both the personal evaluations and the evaluations/rankings of others along with their credibility values. Equally serious effort is needed to design and implement an IoP's evaluation and ranking algorithm, since the ranking should depend on which peer is receiving the ranked list and which peers are included in the list, as well as some overlay and underlay metrics. One last issue is how the peers learn about the IoP, since the latter does not participate in the chunk exchange and might not be included in the tracker's lists. One solution is for the overlay provider and the ISP to cooperate so that the tracker always includes the IoP in his lists. Another solution is that the IoP registers to numerous swarms so that the tracker can suggest him to other peers. Note that this version of the IoP does not participate in the chunk exchange.

8.1.3 Qualitative Evaluation

The players involved in this approach are the peers (end users) with their client application and the ISPs providing the IoP-SIS functionality. Both players have incentives to use the proposed mechanism. Performance improvement is the major advantage of the approach for the end users, while for the ISPs it is the fact that they can indirectly affect the overlay decisions according to their preferences (*e.g.*, performance improvement or possible monetary incentives due to decrease of costs), in a transparent way for the end user. For the mechanism to work, no cooperation between players is required. We assume that the peers have freedom to choose whether or not to follow the suggestions of others or the IoP, even though it is to their benefit to consider such information. The only case of collaboration would be between the ISP and the Overlay Provider, so that the latter includes in his peer lists the respective IoP(s) belonging to the ISP.

The main parameters of this approach is what will be the initial credibility value for all the unknown peers, how often should the lists be exchanged (if it's not standard) and how many IoPs will be used by the ISP. We expect that no legal issues are raised for the ISP, especially since he is not involved in the content exchange. Moreover, no Network Neutrality issues exist, since peers not following this approach will be treated as usual and the performance improvements offered to peers that follow this approach will be transparent. No security issues are expected to arise. The privacy matters arising can be dealt with as in most reputation systems, i.e. with the use of pseudonyms.

To conclude, we believe that the proposed approach is quite innovative since it is both distributed and includes the ISP's involvement. The Ono plug-in for Azureus [CB08] could be considered as a similar idea, but we focus more on distributedly acquired information for overlay performance and on the injection of pseudo-centralized (one or more IoPs per domain) information related to underlay in order to reach a win-win situation both for the peers and the ISP.

8.2 Content Promotion

Content promotion is a new and innovative ETM concept. The key idea is to gather and promote information about the same or similar contents offered in different swarms or even in different content distribution platforms, like eDonkey or BitTorrent. The idea emerges when taking a closer look at the BitTorrent measurements in Section 11. From the top ten of the most popular swarms in terms of population size, there is one movie which is offered in two different swarms. The only difference is the used video codec, one is encoded as *xvid* and one as *divx.* The main idea is to promote the content which is better suited to reach a TripleWin situation. For instance, a BitTorrent swarm with a larger number of sources allows the user to faster download the video while the ISP has more possibilities to adapt the overlay topology to its own needs in contrast to a swarm with only a few peers. In the following, we focus on different BitTorrent swarms only to explain this mechanism clearly. However, the approach can be extended to support also different content distribution platforms.

In general, there exist different swarms in which the peers offer the same or similar contents, however, using different technical parameters. For instance, regarding video contents, there may be differences in the used video or audio codec, the resolution of the video, the actual length of the offered video (*e.g.*, including end titles or not) or the inclusion of subtitles in various languages. Some of these technical parameters are of minor interest for the user, while others are crucial. However, this depends on the individual interests of a user. If the user specifies his interests, the list of available contents can be sorted accordingly. From performance's point of view, it would be more efficient, if all peers are subsumed into a single swarm offering the same content. As a result, as the number of peers increases, a larger upload capacity is provided, less storage capacity is required, and the increased number of peers leads to higher reliability. In order to foster this, it is necessary that the ETM mechanisms gather information about the available contents to the peers and provides them to the peers, such that a requesting peer may select the most appropriate swarm to its needs. While the incentive for the end user and the overlay provider is an improved QoE in terms of higher throughput and reliability as well as less storage costs, the incentive for the ISP to support this ETM mechanism derives from the interaction with other ETM mechanisms. For example, locality promotion can be efficiently used for large swarms as they provide enough possibilities to change the overlay topology to reduce inter-domain traffic. Content promotion is intended to create and maintain large swarms and, thus, supports locality promotion.

8.2.1 Description of the Mechanism

Content promotion as ETM mechanism can be described briefly as follows. A user who wants to download content *A* contacts the SIS and sends information about the desired content, *e.g.,* by sending the address of a tracker offering *A* or by sending a hash value of the content *h(A)*. On this request of the user to download certain content *A*, the SIS (as wrapper/enabler for exchanging information between overlay and underlay) returns a sorted list of swarms which offer the content *A*. The list is ordered to fit the ISP's needs and the user's needs according to a certain metric *M*. In particular, the best swarm might be the one with the highest uploading capacity for the end-user and the highest number of peers in the own ISPs network, such that again locality promotion can be effectively applied. The SIS may also offer references to swarms with similar contents *A'~A*, which fit better to the user's needs. For example, a swarm *B'* offers the same video content like another swarm *B* but in a lower resolution, *e.g.,* for a PDA. For a mobile user, this resolution might be sufficient and also saves resources in terms of download bandwidth, as a smaller video is downloaded. To support the promotion of similar contents, a user may tell the SIS its individual requirements, *e.g.*, in terms of resolution or audio and video quality.

Required information to implement this ETM mechanism include (a) information about the available swarms, i.e. the tracker addresses, (b) for the different contents a list of swarms offering the same or similar contents, and (c) the differences between the swarms with same or similar content. In addition, the user may specify (d) its own requirements regarding the content. In Section 82 we discuss how an ISP can obtain information about the available swarms. Other useful information required to combine and support other ETM mechanisms with content promotion depends on the actually supported ETM mechanism. For locality promotion as a promising example to be beneficially enhanced by content promotion, information about (i) the swarm sizes and (ii) the number of peers in the own ISP's network are of interest. The information (a)-(d) and (i)-(ii) is used to define the metric *M* for sorting the list of swarms. It has to be noted, that the identification of similarity between contents is a difficult task and requires a similarity metric *S*. However, it is not necessary that the SIS implements this metric to determine similarity, but the SIS can also rely on the overlay to get this information. As a first step, *S* can be implemented by comparing the names of the offered files to identify for instance different versions of a particular content. As the user decides whether to follow the SIS's recommendation or not, such a rough similarity metric may already improve the system's performance.

More formally, the flow of messages of the content promotion mechanism looks takes into account the steps (1)-(4) for each user request to the SIS. Step (0) has to be done beforehand and must be executed in regular time steps to keep the information up-to-date. Step (5) means the regular execution of any other ETM mechanism after performing content promotion.

0. Overlay sends information about swarms and content to SIS / SIS collects and processes all information of swarms and contents itself
1. User sends request to SIS to download certain content
2. SIS processes request and orders list of swarms of same or similar contents
3. SIS sends ordered swarm list to user

4. User decides to follow SIS recommendation or not
5. (After this initial request, locality promotion or any other ETM mechanism can be applied)

8.2.2 Classification, Architectural Design, and Implementation

Content promotion as ETM mechanisms can be implemented using the SIS. It belongs to the class of information exchange mechanisms, since information from the overlay is passed to the SIS which processes this information according to the ISP's, the user's and the overlay provider's needs.

Depending on the actual application of the similarity metric, the intelligence may be either located in the overlay or in the SIS itself. On the one hand, if the overlay provides the information which contents are identical or which contents are similar, the intelligence is located in the overlay. Such an overlay provider could be a web platform for trackers which are currently widely used in the Internet to search for contents, i.e. to locate trackers. On the other hand, the SIS could also actively try to determine different swarms with equal contents (*e.g.,* based on hash values) or similar contents. However, then the SIS needs to know the trackers for the different swarms and needs to employ the similarity metric. As the number of swarms, i.e. also trackers, is quite large and there is no central lookup system which knows all trackers in the Internet, it is hard to gather the information from all trackers and then afterwards to process this information with respect to the identification of similar contents. If the SIS has to determine itself which contents are identical/similar, i.e. which swarms offer equal/similar contents, the approach needs very large effort, as a) a large number of swarms have to be checked for the offered content and b) a similarity metric has to be defined and implemented.

Additional effort for implementing content promotion together with locality promotion includes the measurement of (i) the swarm sizes and (ii) the number of peers in the own ISP's network. As a consequence, (a) the swarm sizes have to be checked or estimated or (b) may be offered by the overlay. This is no problem for a single swarm (*e.g.,* get the information every 5 minutes is sufficient), however, for a large number of swarms the measurement load can be quite high.

8.2.3 Qualitative Evaluation

Content promotion is an interesting ETM approach which is based on incentives for all players to approach the TripleWin. The overlay should provide information about swarms, the offered contents and similarities between swarms. The SIS gathers and processes the overlay information and defines a metric for ordering swarm lists. The user sends the initial request to the SIS to get an ordered list of appropriate swarms. The SIS returns this list and then the user decides to follow the recommendation or not. Thus, cooperation is required among the players. The overlay and the SIS need to exchange swarm and content information; the user and the SIS interact with each other to start the process. As mentioned above, this means the user, the overlay provider, and the ISP have direct incentives, which are QoE and higher throughput, an improved reliability, and less costs when applying different ETM mechanisms afterwards. The user will get the best swarm with highest capacity and gets the content fitting best to its wishes (they can be indicated in the request). The ISP can return good swarms, which allows for effectively applying other ETM mechanisms, like locality promotion.

The effectiveness of content promotion strongly depends on the co-existing ETM approach. On the example of locality promotion, we can illustrate this. As in small swarms, the throughput is limited by the upload capacity of the providing peers, the performance can be significantly increased when making the user connect to a swarm with same/similar contents but a larger number of providing users. This improves the throughput and the QoE of the end-user. Furthermore, large swarms offer more possibilities to adapt the overlay topology in such a way that costs due to inter-domain traffic are reduced. Users with different requirements, *e.g.,* mobile users using a PDA with a low resolution or DSL users expecting HDTV quality, can download appropriate contents. This saves bandwidth for the end-user and thus also for the ISP. The video playout for example can be improved if the downlink of a mobile user is the bottleneck. (The downlink bandwidth of a mobile user may not be able get the desired throughput for HDTV quality, thus, several video frames get lost. However, the HDTV resolution is then scaled down to a PDA resolution that is offered as a similar content, in order to show the video on the PDA).

Privacy issues are critical in the context of content promotion, as the user requests the SIS to offer him the best swarms for certain content. Regarding legal issues, it might be critical if the SIS “understands” the offered contents in different swarms. This means that the SIS might also understand whether content is legal or not. Then, it seems to be a problem that the SIS offers the best swarm containing illegal content to the requesting user.

8.3 Overlay Enhancements

The approaches presented in this subsection do not stand as an ETM mechanism by themselves. In fact, the approaches presented here are more of overlay enhancements that introduce new objectives and new methods for the overlay optimization. They can be seen as complementary to ETM mechanisms that pursue similar or, better, incentive-compatible objectives but on a lower level, with little or any cross layer exchange of information.

8.3.1 Tree-based Dynamic Locality-aware Neighbor Selection

This mechanism offers a way for peer-to-peer file-sharing applications to group other peers according to the network topology and use the grouping for efficient data dissemination.

8.3.1.1 Description

The Internet topology is interpreted as a logical tree, with the Tier-1 backbone networks as root; other ISPs’ networks are connected to this root, end customers are leaves to the other ISPs. Therefore, the higher a data path in the tree comes, the more expensive it is.

Many peer-to-peer systems have centralized components that can be given a special role in the system. An example is the tracker in BitTorrent. Similar entities can be found in other file-distribution or video streaming systems, such as data-driven video-streaming systems; thus, the approach can in principle be applied to such applications too. The tracker maintains information about each client in the system. In particular it maintains download progress of the clients and which ones are seeds. This information is obtained by the peers themselves, who update the tracker, somewhat infrequently though. From this information, the tracker can derive the progress of the download for each subtree of the aforementioned logical Internet tree. The tracker can construct the tree by querying the

ISP provided service i.e. SIS for the locations of all clients. At this point it can be assumed that each ISP runs its own SIS, and thus the location of each peer is requested from the responsible SIS.

While the rough location of peers could be retrieved by public IP-Geolocation services, more detailed information may be provided by ISPs. For instance, the ISP could give information about peers in the same POP that possibly could not be retrieved from public information services. Additionally, by providing the logical position in the tree, the ISP may also give information about preferred network paths, which should be used from/to a specific peer.

As the tracker is responsible for the maintenance of the distribution graph, it can use its knowledge about download progress in the subtrees to provide the optimal file-distribution. Ideally, this would go as follows. A file being distributed should be first replicated in the two top subtrees, thus it should flow through Tier 1 backbone network only once. The distribution of chunks per nodes in each of these two subtrees does not matter at the moment. It is only important that all chunks are available on both sides. Once this is the case, the whole process repeats for the subtrees of each of the mentioned subtrees and continues until the download is completed.

As this strict splitting of subtrees into smaller subtrees after receiving a full copy of the file may lead to performance degradation, different policies such as allowing a certain numbers of out-of-subtree neighbors may be applied.

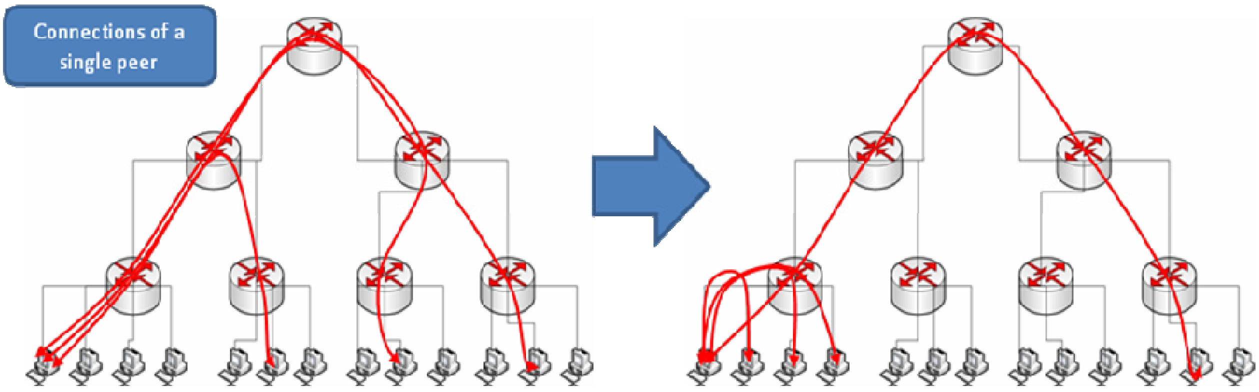


Figure 8.1: Example of a network reorganisation according to the proposed algorithm.

The implementation of the above algorithm requires the following input:

- The tracker needs information about the Internet topology (*e.g.,* AS map)
- The tracker needs information about the location of each connected client within this Internet topology from the corresponding ISP of the client
- The tracker needs a policy that determines the composition of the neighbor lists (*e.g.,* X neighbors outside the subtree are allowed at Y replication rate)

This input will be used by the tracker to compute a list of neighbors for each requesting client that will meet the current distribution status (replication rate in the subtree) and the supplied target policy.

The standard flow of events would be

1. Peer connects to the tracker and requests a list of neighbors.
2. Tracker requests position of the peer from the SIS that is responsible for the client.

3. Tracker computes replication rate in the subtrees.
4. Tracker computes a list of neighbors and sends to the client.
5. Peer connects to the other clients specified in the received list and continues normal operation.

The intelligence and decision making of this mechanism is concentrated in the tracker. The clients do not need to be changed. At the ISP the required information about the network nodes' locations must be published using a SIS. The main optimization objective of this mechanism is the reduction of the amount of traffic traversing nodes close to the root.

It should also be noted that there is a strong correlation of the information learned and the calculations made by different trackers. Therefore, an "internet map service", which delivers a network map with the backbone and the transit networks, would surely be a useful entity to avoid duplication of work at different trackers.

8.3.1.2 Classification

The proposed mechanism is an Information Exchange mechanism. It uses the location information provided by the ISP (through the SIS entity) about the node locations and the overall Internet topology. Most of the logic is moved to the tracker, since the SIS is just giving the locations of the peers and nothing else.

8.3.1.3 Qualitative Evaluation

The players involved in this mechanism are the ISPs and the Overlay tracker.

The ISPs have a strong incentive to support the adoption of the mechanism, as they can save on the expensive inter-domain traffic costs or improve the network as localized traffic aids in congestion-avoidance. On the other hand, overlay providers may not have an incentive as strong as the ISPs, as the overlay and its users may not always benefit from the reduction of inter-domain traffic. Indeed, in the case when the number of replicas in the client's subtree is not high enough and the remaining peers are not able to completely fill the download bandwidth of the peer, then performance may degrade for the clients. In the case where the local peers are able to provide enough bandwidth to all peers, the download speed of a client will benefit from the mechanism and the overlay will have an incentive to apply it.

However, the actual gains for the overlay and the ISP strongly depend on the neighbor-list-composition policy of the tracker. Thus, the policy could be set to never degrade the performance of the overlay (*e.g.,* allow always a certain amount of connections outside the subtree), giving an argument for the overlay to implement it. It should be noted here that according to [BCC+06], allowing ca. 20 % of all connections to "outside" peers does not affect the performance for the user. So there could be chosen a policy, which is very conservative and does not affect the user in a noticeable way, at the cost of having improvements for the ISP in cross-domain traffic savings.

The ISP and the overlay provider in the form of the tracker must closely cooperate; the tracker relies on the information provided by the ISP. The tracker will then optimize the traffic of the overlay in the interest of the ISP, which as explained above can in general be made compatible to the tracker's interest.

If ISPs are willing to implement SIS or similar services, the effort for implementation of this mechanism is low: As only the tracker needs to be changed and client software can continue to use the standard peer-to-peer protocol, the introduction and adoption of this mechanism would be feasible.

From the viewpoint of network neutrality no problems for the ISP should arise, as its offered service is application agnostic, same holds for legal issues.

Regarding the Architectural Design Space of D3.1, none of the offered scenarios fits this mechanism, but the proposed solution fits the general architecture.

8.3.2 A New Incentive Mechanism – Private and Shared History

In this section, we introduce a new incentive mechanism, called Private and Shared History (PSH), to overcome the limited local view of peers for assessing the reputation of other peers. These reputation values are then used to support the peer selection mechanism of overlay applications. PSH increases the success ratio of transactions in an overlay network, creating less overhead for peer-to-peer applications and less overall traffic for ISPs. In particular, PSH is not a new ETM mechanism, but improves the functionality of the overlay by providing this kind of incentive mechanism. Therefore, there is no impact on the architectural design, cf. D3.1, and the implementation of any other ETM model but PSH could be used in conjunction with other ETM approaches. In Subsection 8.3.2.1, we describe the PSH mechanism, while its performance is qualitatively evaluated in Subsection 8.3.2.2.

8.3.2.1 Description of the Mechanism

While peer-to-peer systems have several advantages over centralized systems, *e.g.,* load balancing, robustness, scalability, and fault tolerance, there are challenges still open in peer-to-peer systems. Free-riders [AH00], malicious peers, Sybil attacks [Do02], self-interest [SP03], and other forms of attacks [NCW05] can be addressed by proper incentive mechanisms.
A popular and widely used incentive scheme is Tit-for-tat (TFT), a variant of which is used in BitTorrent [Co03]; see Subsection 3.5.1. With TFT, users may only download as much as they upload, given an initial credit limit. This incentive scheme keeps a per-peer history of resource transactions that is solely based on local observation. Thus, peers have a very limited view of all transactions to peers with direct reciprocity. The Transitive TFT mechanism, based on shared history, makes transaction information accessible to other peers. This allows indirect reciprocity to be detectable. However, such shared history approaches are prone to false reports and collusion or have scalability issues [FLS+04].

PSH (Private and Shared History) [BKH+08] is an enhanced incentive mechanism which is a combination of private and shared histories. PSH uses shared history to propagate resource exchange information, and private history to verify the correctness of this information.

To allow an initial transaction, a peer may consume resources until a credit limit is reached. The credit limit must be low enough to discourage peers from creating new identities (white-washing). The workload is placed on the requesting peer, preventing DoS attacks. PSH behaves like TFT if two peers are about to exchange resources and previous transaction information is present in the other's private history. If it is not present, PSH looks for a path, that is, a linked list of peers with transitive history information, from the source peer to the target peer. Each of those peers in the path is requested to issue a check, which means to transfer their signed credit balance between the source and the next in the path or the target.

Figure 8.2 shows the general architecture of the request / response handling in PSH. Arrows between big rectangular gray boxes indicate a network connection between two peers. The figure shows that the workload is mainly on the requesting peer, because path searching is done on the requesting peer. In Figure 8.2, peer s sends a resource request to peer t. If the request succeeds, then both private histories from peer s and t are updated as in TFT. If it fails, peer s searches for a path. If a path cannot be found, the request fails. If the path cannot be traversed, the request fails as well. If the path can be traversed and a valid check can be returned, peer s asks peer t again with this valid check.

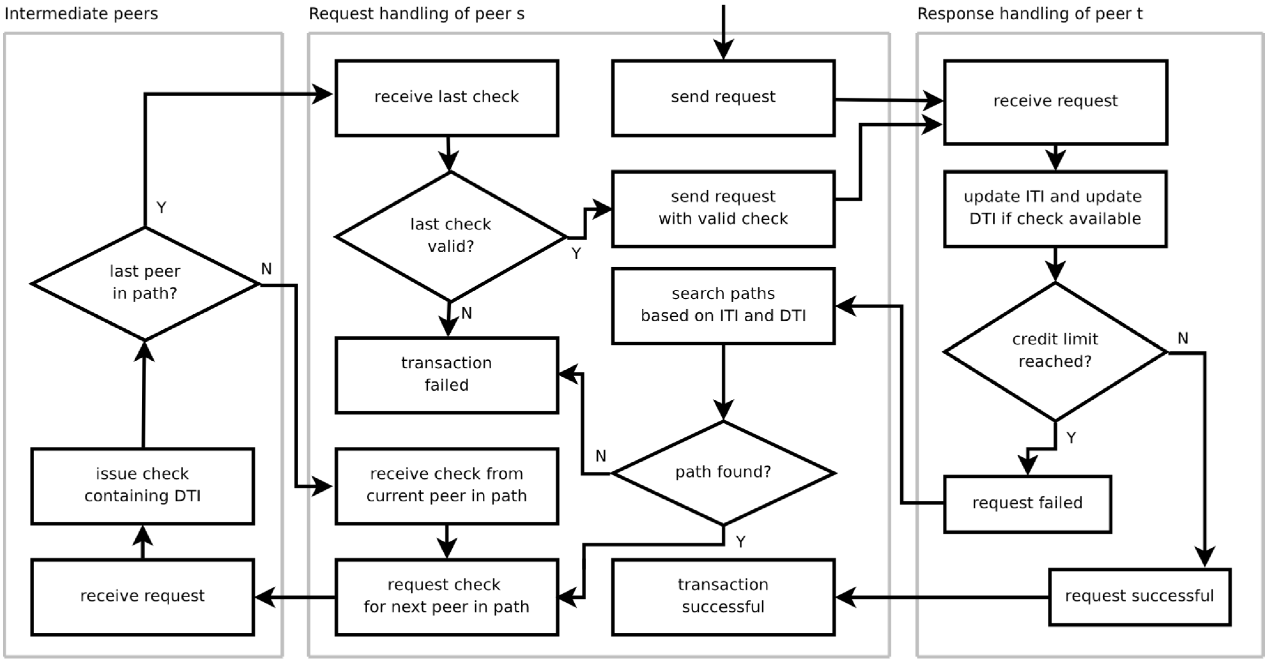

Intermediate peers
Request handling of peer s
Response handling of peer t
receive last check
send request
receive request
last check valid?
Y
N
send request with valid check
update ITI and update DTI if check available
Y
last peer in path?
N
search paths based on ITI and DTI
credit limit reached?
N
transaction failed
Y
issue check containing DTI
receive check from current peer in path
N
path found?
request failed
Y
receive request
request check for next peer in path
transaction successful
request successful

Figure 8.2: General PSH incentive mechanism.

As Figure 8.2 depicts, each peer stores two tables of history information, a direct transaction information (DTI) table and an indirect transaction information (ITI) table. A DTI table contains information based on direct reciprocity (private history). An ITI table is based on indirect reciprocity (shared history). A DTI entry for peer x and peer y $DTI_x(y)$ is defined as the amount of exchanged resources, for example bytes transferred if the resource in question is bandwidth. After a successful resource transaction from peer x to peer y, the former updates $DTI_x(y)$, while peer y updates $DTI_y(x)$. Along with each request and response message, a subset of DTI entries with the highest timestamps is exchanged in order to spread data about past transactions while avoiding creating new connections. The ITI table contains accumulated DTI from other peers. The ITI and DTI tables are used to find a path from a given source to a target. If such a path exists, then indirect reciprocity can be inferred.

Since many paths may exist, the size of a complete ITI table has polynomial complexity $O(n^2)$, where n is the total number of peers in the system. A reduction of complexity can be achieved by evaluating a limited path length L instead of $|n|$, where $L<n$. Further

complexity reduction can be achieved by expiring transactions in the *ITI* table using a time decay function $f_{decay}(transaction)$. Therefore, not all existing paths are found.

Once a path has been found, using the shared history, the verification process starts. This process queries every intermediate peer on that path $P(s,x,...,t)$, where s is the source, t is the target and $x,...$ are intermediate peers, to issue a check. An intermediate peer receives a request containing the source, the predecessor and successor peers. The intermediate peer verifies and accounts the balance of the predecessor and successor peers, with the effect that the intermediate peer transfers its debts from the predecessor to the source. The intermediate peer sends a check with the new balance to the source peer. In order to verify the identity of the peer issuing a check, peers must exchange public keys on first contact and sign all outgoing checks. Each intermediate peer is requested sequentially to send a check to the requesting peer until the successor peer is the target peer. If an intermediate peer fails to send a check, *e.g.,* because of an imbalance due to old history data, then the path is invalid.

8.3.2.2 Qualitative Evaluation

Two versions of PSH have been simulated and compared to TFT: PSH as described above, and PSH with a reduced number of message transfers (PSH_r), both with $L=3$. The reduction has been achieved by sending a subset of the *DTI* only if a request failed. PSH_r does not retry to send the request after a failed transaction, while PSH retries up to 3 times. A retry in PSH can be successful if a path can be found and verified. A retry in TFT would always fail because a peer only updates its history after a successful transaction. In contrast, PSH may update its history with a check from another peer and a retry may be successful.

Simulations show that PSH success ratio is always higher than TFT, especially when the number of unique resources is around 32, at which point the success ratio is 70% higher. The higher success ratio is due to the shared history data. PSH_r is at an intermediate position, performing better than TFT, but worse than PSH, since the algorithm does not retry transactions. The use of PSH and PSH_r cause an increase in the number of messages, which is higher when compared to TFT, since PSH involves the exchange of history information. PSH requires up to twice as many messages as TFT per transaction, peaking on 32 unique resources, when it requires twice as many messages than TFT. PSH_r has a reduced number of messages; on average, only 2.22% above TFT, at maximum 6.4%. More detailed results can be seen in [BKH+08].

PSH aims at improving overlay functionality by reducing effect of free-riding in overlay networks. It provides means to reliably obtain transitive reputation values from other peers. The information must be used by the overlay network when performing peer selection. Since other presented ETM mechanisms work by influencing peer selection, it is imperative to combine those with an appropriate reputation mechanism in order to assure proper and fair functionality of the overlay.

8.3.3 Locality-aware Overlay Caching

In normal BitTorrent the main goal is to replicate one file as fast as possible. Once a file is completely downloaded a peer has no incentive to continue sharing the file. This often results in a very low percentage of seeders and can result in a too early “death” of a torrent if all seeders leave the swarm even if there are still some leechers in the swarm. The situation is even worse if we try to find local replicas of required chunks. The ides behind the overlay caching is to increase the availability of chunks both on the global scale and, additionally, in peer's network proximity.

The overlay caching requires a cross-tracker information exchange that can be done as follows:

a) peers can offer "old" content and get currently required content in return and

b) peers acquire a better reputation for offering "old" content

For this to work the reputation information and who-has-what information can be stored in the overlay (alternative 1) or in the SIS node (alternative 2).

8.3.3.1 Description

The nodes must be able to find local nodes interested in the same content. This can be done as follows: In a tracker-based system such as BitTorrent, nodes that exchange some information, *e.g.,* have-messages or even file pieces, can additionally exchange the information on what content they have consumed up to now. This allows finding nodes with the biggest interest overlap. In a tracker-less system the peers attach the list of consumed files to search messages. This way peers with similar interests can be found and become neighbors. In both cases this results in an overlay where peers are organized in "interest clusters". In order to avoid the separation of the overlay, additional random interconnections are added.

The second kind of information is the underlay locality. The peers preferably create connections to peers located in the same AS. The information about the AS is either obtained by GeoIP service, such as MaxMind [MM] or provided by SIS component (optional). We combine both interest similarity and locality in one metric, expressed for two peers P1 and P2 as:

$$Sim(P_1, P_2) := \alpha \cdot InterestSim(P_1, P_1) + (1-\alpha) \cdot SameLocation(P_1, P_1)$$

Here, α with 0<α<1 is the relative weight of the content similarity criterion.

The content similarity is asymmetric and defined as:

$$InterestSim(P_1, P_2) := \frac{|\,Files(P_1) \cap Files(P_2)\,|}{|\,Files(P_1)\,|}$$

The peers keep downloaded content in their local caches that are limited in size. Once the cache is full the peers decide what content to delete based on the following parameters: content age, content popularity, number of copies in the peers neighborhood cluster.

All of this information can be approximated by the overlay. A possible optimization is to offer a SIS service that collects this information on behalf of the overlay to increase the cache efficiency.

The peers that keep their content in cache can be seen as long-term seeders because they don't leave the network once they finished the download. Since they have to distribute their upload capacity among several swarms caching peers have to decide how much bandwidth they want to allocate to each swarm. This decision is made depending on the popularity of the file and the number of concurrent seeders. The information can be either provided by the overlay itself, by the tracker or by additional entities such as the SIS. Here, SIS could provide the information about content popularity and availability in the ISP domain.

In summary the input information for this mechanism is:

- current swarm size (for each swarm)
- content age (per swarm)
- number of seeders in the swarm

The main output of the mechanism is the decision which content is to keep in which caches and how much upload bandwidth to contribute to which swarm.

The information flow of this mechanism is as follows:

1. Peer downloads a file completely.
2. If the cache is full the peer asks his neighbors caches for their state.
3. In periodic intervals the peer analyzes the popularity of single swarms and decides for which of the cached files it should act as seeder.

8.3.3.2 Classification

This mechanism can be classified as follows:

Mechanism class	**Classification**
Information Exchange Mechanism	Yes, inside of the overlay and optional with tracker and SIS
Pure Traffic Management Mechanisms	No
New Architectural Component	Optional, if SIS is used to estimate the swarm population

8.3.3.3 Qualitative Evaluation

The main player involved in this mechanism is the overlay. It is responsible for the caching of files and their seeding. Additional parties can get involved: Trackers can be modified to improve the overlay performance, by providing additional information compared to the locality-unaware version. Furthermore, ISPs can offer SIS components that help local peers to find each other.

This mechanism offers incentives for participation to all parties: an ISP is interested to localize peer-to-peer traffic in its AS. The user is interested in content availability and fast downloads. Finally, the content provider (who offers the initial seed and tracker) is interested in shifting the load from his servers to peers. The main goal of the mechanism is to increase the content availability in swarms that lack enough seeders while distributing the seeders' contribution more efficient among swarms. The idea behind it is that in very popular swarms the content availability and available upload capacity is high enough while in smaller swarms the content availability is low. This is even worse if peers try to download content from local sources since in medium and small-sized swarms there might be not enough local content copies. Here, caches that previously downloaded the content can provide the content preferably to nearby peers.

The cooperation is needed to exchange the relevant information honestly, *e.g.,* the tracker must report the real size of the swarm and the real number of seeders in order to achieve high performance.

This approach is expected to perform well in scenarios where a significant number of peers is localized inside of each AS (other AS will benefit less). Note that these peers do not have to download the same file simultaneously. It is enough if there is enough overlap in their interests.

The measurements done by overlay caches have to be performed in two ways: each time some content has to be deleted from cache and on a regular basis to decide whether a cache should reassign its upload bandwidth among swarms. This can be done in a scale of seconds. Each cache can communicate with three parties:

1. Neighbor caches to be informed about their state and content popularity estimations.
2. Tracker to obtain information about the swarm popularity. This would require the modification of trackers and is, therefore, optional.
3. SIS to find new caches in its swarm.

Note that the mechanism does not require any communication among single trackers or SIS components of single ISPs.

The relevant parameters are: How big should the cache be? How many swarms should a cache serve in parallel? What is the target seed-ratio for a swarm?

This approach has some similarity with the ISP-owned peer and ISP caches, since there is a relevant difference that the content is served by peers.

There seem to be no legal issues with this approach, unless an ISP starts to provide information to the overlay about locality and swarm popularity. But even then the ISP's involvement appears uncritical.

Regarding security the question arises whether some peers, tracker or even the ISP could provide false reports and how these can be recognized. Regarding the privacy, peers exchange the information about the content they already consumed. The privacy can be increased by exchanging hashes of files instead of file names.

The most similar architecture alternative is the Honey Pot.

9 Comparative Assessment of ETM Approaches

In the previous sections we have seen a number of ETM approaches presented. Some of them can stand alone, providing some basic functionality; others can be seen as extensions which address specific issues that the basic functionality does not handle. Some of them are based on a centralized architecture; others assume distributed decisions leading to a global objective.

The goal of this section is to summarize what has been presented by categorizing the various approaches according to certain criteria and by mapping the functionalities presented on the basic architectural components that have been described in D3.1. Moreover, we investigate the properties of such approaches regarding their applicability to peer-to-peer Video-on-Demand (VoD) overlays, which are the target application of SmoothIT, as specified in Section 7.3 of D1.1.

9.1 Classification of ETM Approaches

We start by classifying the proposed approaches according to their inherent characteristics. Three major properties characterize a mechanism: whether it defines an ***Information Exchange*** mechanism, whether it includes certain ***Traffic Management*** techniques and whether it introduces a new ***Architectural Component or Interface*** or a new/enhanced ***Overlay Entity***.

An *Information Exchange* mechanism defines the way overlay and underlay entities exchange the information required by the ETM mechanism's "intelligence" module. In the target application scenario, i.e. a BitTorrent-like VoD, the overlay system already operates one or more trackers capable of providing the requesting peers with a list of (remote) peers that share the required content. In other cases, peers can also directly exchange lists of peers among themselves. If not inherently provided by the overlay, such a communication must be implemented. However, some approaches apply to cases where no tracker (or other central overlay entity) exists, i.e. when the approach is distributed. In this case, we assume a trackerless system. Further classification could be provided if there was introduced a distinction between approaches resulting in information exchange and those resulting in information utilization. This distinction though would not be clear, since many approaches introduce both functionalities. Only if some approaches are considered as extensions of others, this distinction could be more clearly applicable. As will be seen later in this section, this view in terms of functionality is possible; hence, such a distinction should be kept in mind.

Regarding *Traffic Management* techniques, these may include traffic shaping, throttling, traffic differentiation and prioritization and other QoS techniques. Approaches belonging to this category specify how traffic is handled according to the optimization goal that each mechanism seeks. The approaches typically cooperate with an information exchange mechanism in order to perform the traffic management operation efficiently.

As far as new *Architectural Components and Interfaces* are concerned, the respective category defines whether the proposed ETM approach introduces a new component in the basic architectural scheme defined in Section 8 of D3.1. More specifically, the SIS component and sub-components are taken for granted, and only new additions are considered here. Additionally, any new interfaces of existing components are considered here.

Finally, we have mechanisms that introduce new entities in the overlay. Although such entities may be transparent to the standard overlay entities, it is an important feature to have in mind for the classification of approaches. Furthermore, in the same category fall the approaches that provide enhanced functionality to existing entities, also transparently to the rest of the entities.

Having defined the characterization parameters, Table 9.1 summarizes the properties of each mechanism:

Table 9.1: Classification of ETM approaches.

ETM approach	Abbreviation	Information Exchange Mechanism	Traffic Management Mechanism	New Architectural Component and Interface	New Overlay Entity
BGP-based Locality promotion	BGP-Loc	Yes	No	No	No
Centralized SIS and Dynamic Locality	Dyn-Loc	Yes	No	No	No
Tree-based Dynamic Locality-aware Neighbor Selection	Tree-Loc	Yes	No	New interface for tracker-SIS communication	Enhanced tracker
Locality-aware Tit-4-Tat/Unchoking	T4T-Loc	Yes	No	Modified Client Application	No
Insertion of ISP-owned Peers	IoP	No	No	No	IoP
QoS Incentive for Overlay Service Providers and End-Users	E2E-QoS	No	Yes	No	No
Locality-based Traffic Shaping	TS-Loc	Yes	Yes	QoS aware routers	No
VPN-assisted Overlays	QoE-VPN	No	Yes	VPN components	Gateway
QoE-aware Feedback Mechanism	QoE-feed	Yes	No	No	No
Distributed Exchange of Peer Lists	DistrXch	Yes	No	Modified Client Application & new interface for the IoP-SIS communication	IoP
Content Promotion	ContProm	Yes	No	New interface between tracker and SIS	No
Private and shared history-based incentive mechanism	PSH	Yes	No	Modified Client Application & interfaces	No
Locality-aware Overlay Caching	CacheLoc	Yes	No	Modified Client Application	No

It is obvious that some of the proposed approaches share common characteristics while others don't. Of course, some approaches, if considered as extensions to others, introduce little overhead and can be classified into a single category. Having this in mind, we proceed with the classification of each mechanism according to its functionality.

9.2 ETM Approaches' Roadmap

Next, based on the previous observations, we group related approaches and try to identify possible combinations and increments, based on the functionality offered. At the same time, we provide a first insight on how to construct a fully functional ETM framework, combining approaches with different scopes. Figure 9.1 serves both purposes, as it provides a categorization of the approaches according to their functionality and at the same time it provides a roadmap for a complete solution.

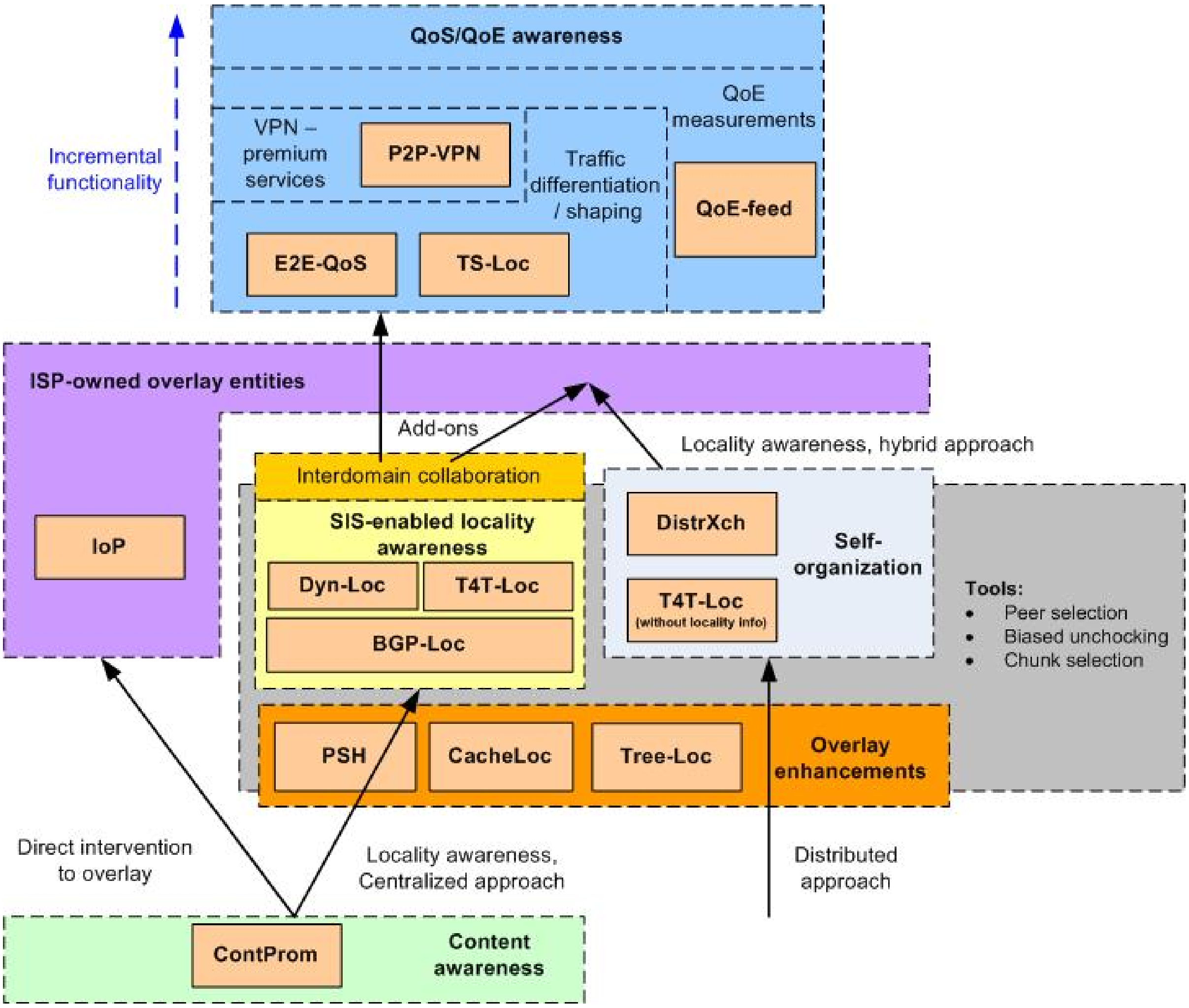


Figure 9.1: Roadmap of proposed ETM approaches.

Summarizing what is depicted in Figure 9.1, we have three fundamental classes of approaches: the "ISP-owned overlay entities", the "SIS-enabled locality awareness" and the "Self-organization". These can be seen either as alternatives, although there is an overlap among them, or can be combined to produce new approaches. Such a case could be to use *IoP* as an add-on to *SIS-enabled locality awareness* approaches or *Self-organization* approaches, in a different way for each case.

Regarding "Content awareness", if applied, it can be seen as a first step before the deployment of any other mechanism necessary to specify the content (and swarms) that the SIS or ISP-owned peers should consider. The same objective but through different means is achieved by the *CacheLoc* approach, which can also be combined with "Content awareness". The difference between *CacheLoc* and *ContProm* is that, in the latter, different active swarms that offer similar content are combined while, in the former, information about past activity (i.e., swarms that no longer exist) is propagated to new swarms.

The "Overlay Enhancements" can be considered as separate and independent approaches, although some of them are depending on information provided by the ISP. Such an approach is *Tree-Loc*, which can be considered as a "locality enhancement" for the overlay.

Finally, the "QoS/QoE awareness" is an extension of the fundamental approaches, on top of which quality metrics are considered and quality optimizations are deployed. In this block, the proposed approaches are also presented in an incremental way, with basic QoS functionality residing at the lower level and QoE enhancements appearing at the top level. Note however, that *QoE-feed*, as an information exchange approach, can be also considered as a separate mechanism, not requiring underlying QoS approaches. In fact, QoE can provide any of the underlying layers with user-centric information about the perceived quality. Such information can affect the decisions made in the overlay, in the same way network measurements or economic criteria are considered.

9.3 ETM Approaches and Architectural Implications

Through this analysis, it became necessary to make a distinction between functionality and architecture related to each approach proposed. While in the previous subsection we provided a functionality-based grouping and positioning of the various ETM approaches, in this subsection we depict them in an incremental manner in such a way that it becomes clear how each proposal relates with the rest from the architectural point of view.

According to the previous sections, it is clear that the majority of proposed approaches are based on the SIS architecture, with some of them providing the basic functionalities and others offering extensions and enhancements that strengthen the notion of ETM. Based on this observation, Figure 9.2 depicts how different approaches map to the specific SIS architecture.

In Figure 9.2 the grey boxes on the background depict the current architectural design of the SIS, as described in D3.1. Note that we have included one more component on the upper part of the diagram, the "Other Overlay Entities" box, which represents all the other entities that a peer-to-peer network would include (apart from the peers), so that we make sure that if there is also a need to "communicate" with such entities this is also taken into account.

An important note is that the "floating" objects only depict which components are necessary to implement the respective mechanisms or which components are needed to realize a specific extension. The color of each floating box corresponds to the color of each of the main boxes in Figure 9.1, apart from the "Basic Functionality", i.e. the BGP-Loc approach, which is marked in red. As far as the existing overlay components are concerned (peers and other overlay entities), it is not yet depicted which of these and how they should be "altered" (by means of software add-ons) in order for the mechanism to be fully functional.

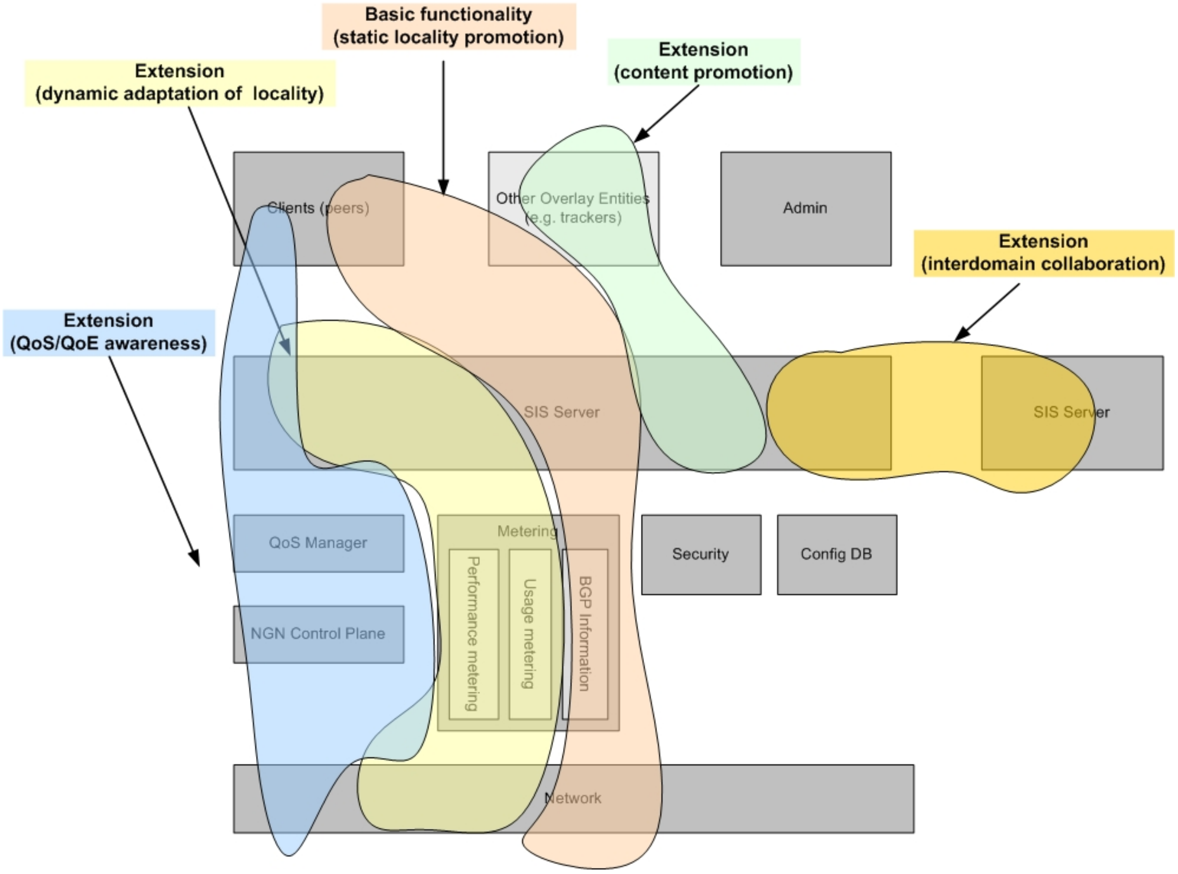


Figure 9.2: Mapping of centralized ETM mechanisms to architecture.

We start with what we consider as **Basic Functionality**, which includes the interaction between peers, trackers, the SIS server (with some basic ETM intelligence) and the metering component (in fact, the BGP information subcomponent). This basic functionality is that of the *BGP-Loc* mechanism, since it provides a way to sort a list of peers based on BGP information. The sorted list can be used in the peer selection procedure (either the standard procedure followed by the overlay application or by a modified version of the software) or in other procedures like the biased unchocking, as described in *T4T-Loc*. Therefore, we take the *BGP-Loc* as the first prototype of an ETM approach to be implemented. This is mainly due to the fact that few components are required and the functionality offered is basic for the rest of the approaches. Note however that the ranking algorithm is rather basic and takes into account "economic" information only indirectly, to the extent that such information in incorporated in the BGP attributes. In fact, it is the "local preference" value of BGP that hides all the complex business relationships between ISPs and implies that some sort of "economic" information has been considered in order to define those values. Hence, the basic mechanism cannot stand alone as an ETM approach but rather as a locality promotion mechanism that can be proven economically sound as well.

Having defined the core functionality, we can extend this approach by adding new features and enhancing the ETM intelligence. The first extension would be to provide **Dynamic**

Adaptation of Locality, as described in the mechanisms *Dyn-Loc* and *T4T-Loc*. Here locality is promoted only whenever it is required, i.e. in an adaptive manner. Performance and monetary triggers specify when locality should be promoted. Although these mechanisms fall under the same category, their approaches are quite different and this will hold for their implementation.

Additionally, we can incrementally deploy **QoS/QoE Awareness** as described in *E2E-QoS*, *TS-Loc*, *QoE-VPN* and *QoE-feed*. In Figure 9.2, it is made apparent which parts of the architecture are involved in order to provide the extended functionality. Note however that *P2P-VPN* introduces some special functionality that may fall outside the current scope of the “QoS Manager” and “NGN Control Plane” components and that is currently not depicted in the above diagram. In this case, additional components may be needed. Moreover, we see that *QoE-feed* expands the client as well. This is done since user-centric measurements are required in other to define the levels of QoE a user perceives and be able to react accordingly.

In each of the aforementioned incremental steps, **Content Promotion** can be also deployed, which implies an additional coordination between the SIS and the overlay entities, such as the trackers, as described in *ContProm* and *CacheLoc*.

At any point we should not forget that a level of **inter-domain collaboration** between SISes should take place. This functionality is beneficial for all the mechanisms, because it will enhance their effectiveness, though it can be seen as an extension. Although we first have to finalize the intra-domain functionality, the inter-domain case should always be considered. For example, *Dyn-Loc* considers the case of a single AS domain and tries to optimize incoming and outgoing traffic. But, given that ISPs might collaborate, we can subsequently consider the case of ISPs exchanging messages in order to fine-tune and coordinate their actions.

Related to the issue of inter-domain collaboration, is the case of the *Tree-Loc* approach. *Tree-Loc* does not consider inter-domain collaboration, it just affects overlay traffic travelling over multiple domains, considering locality information provided by the SIS. Of course, instead of having a single entity collecting all these information, we could use a hierarchical structure of collaborating SISes (based on the same tree structure of the Internet), that allows the exchange of such information. The same approach can be used for the SIS-SIS interaction if included in the *Dyn-Loc* approach.

So far we have studied only the centralized approaches described previously. We now will consider the decentralized and hybrid approaches, summarized in Figure 9.3.

We start with the insertion of *IoPs*, which assumes that content is stored (downloaded either progressively or from the beginning) by an ISP-owned entity, which indirectly affects the content distribution within an AS by participating in the overlay. This functionality can be enhanced by **Content Promotion** as described in *ContProm* and it corresponds to the involvement of an SIS module that communicates with the tracker and can advice the IoP which swarms to participate in, based on information like popularity.

Another functionality that should not be considered as an addition to the previous one, but includes almost the same architectural components, is the **Distributed ETM with SIS-enhanced IoPs**. Here a change in the overlay protocol as ran by the clients might be necessary in the case that peers do not originally share overlay information (such as lists of peers). In order for the ISP to be able to affect the decisions made, a simple implementation of the SIS is necessary. This is represented in Figure 9.1, as the point where two arrows meet, one coming from the “SIS-enabled locality awareness” box and the other from the “Self-organization” box respectively. More details can be found in Subsection 8.1, where *DistrXch* is introduced. Similar functionality can be attained by altering/adapting the approach proposed in *T4T-Loc*.

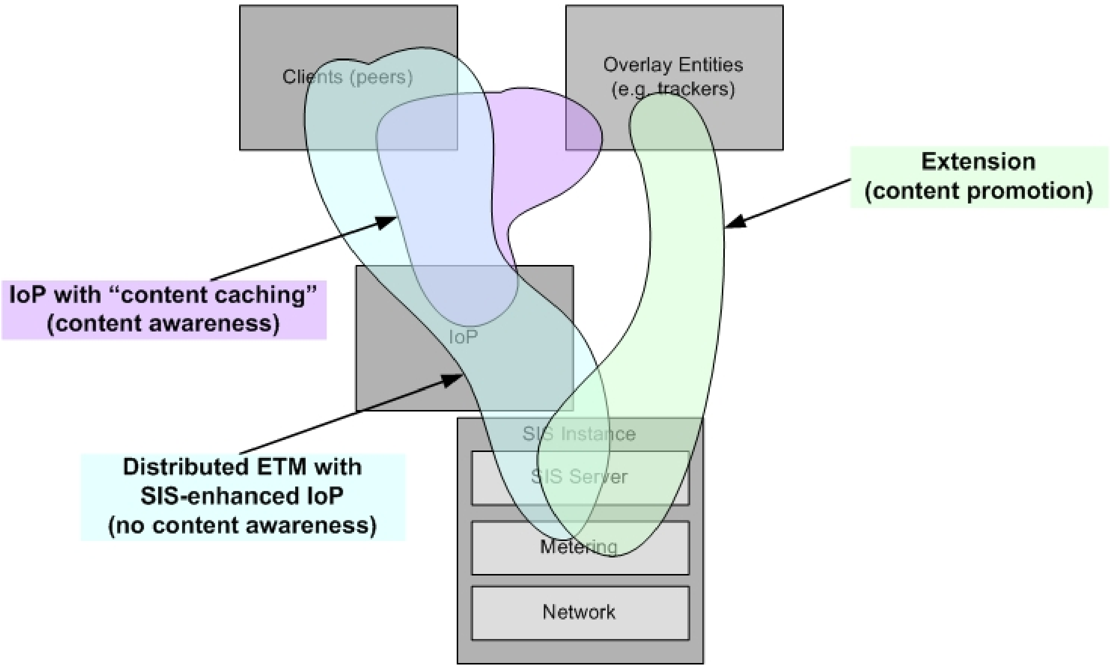


Figure 9.3: Mapping of distributed ETM mechanisms to architecture.

There are cases though and versions of client software that already implement exchange of such overlay information between peers [PEX], [BTPEC]. Here, we need to come up with heuristics that define how this information is parsed by the peers, without requiring the constant intervention of the SIS to re-rank the available list, even when one or two new (remote) peers are discovered by the peer.

9.4 Important Considerations

So far, we have seen that the realization of the various ETM approaches in general involves altering the peer selection algorithm or the unchocking algorithm, although there are exceptions such as the insertion of the IoP. Although this is a desirable effect, it mainly pertains the case of file-sharing overlay applications. Changing the way peers are selected or unchoked alters the downloading procedure of the file. Different peers offer different file chunks but the ultimate outcome is not affected. Indeed, at the end, the entire file is downloaded, probably quicker than when using the standard procedures.

Overlay video streaming applications introduce a new restriction that should be considered by the ETM approaches. In particular, play-out buffer should always be full while playing of the video should continuously be advancing, in order to minimize jitter and the possibility for bad QoE. This fact is very important for applications of this category. Existing chuck selection strategies like the “least shared chunk” selection and the “give-to-get” approach should not be ignored but might need to be improved when live or streaming content is considered. Thus, in the cases of video streaming peer-to-peer applications, QoE violation is less acceptable than in the case of file-sharing. Therefore, for VoD, QoE objectives become more important than monetary objectives (associated with, *e.g.,* 95th percentile charging rule). For example, there might be the case that thresholds related to monetary savings are violated in order to assure the smooth playback of a video content.

In this sense, considering the framework of approaches that were described in the previous subsections, being completely content agnostic might not be desired for this type of content. More specifically, peer and chunk selection strategies, as well as unchoking thresholds might need to be adapted as the video plays. One step towards this direction could be to provide the "Content awareness" layer of SIS with some intelligence that can advise the peer which (remote) peers he should connect to, based on the current content of the play-out buffer. Such approaches though may turn out to be very hard to design and implement; this issue can be considered in the future.

Another, equally important issue is *timescales*. Each mechanism performs an action at different timescales and the overlay entities exchange information at different timescales. The first point refers to how can different mechanisms be combined and extended. It is important to know on which timescale each mechanism works so as to be able to compose a more complex mechanism from simple functionalities. The second point refers to the fact that the timing of certain mechanisms is not irrespective of the time certain overlay procedures take place.

9.5 Conclusions and Open Issues

This section had as objective to present the correlations, possible similarities, and combinations of the previously presented ETM mechanisms. Briefly, we have provided the following:

- A categorization of mechanisms according to some inherent characteristics, like the information exchange and traffic management characteristics, architectural implications and overlay additions.
- A classification of mechanisms according to their functionalities and a roadmap for a complete solution.
- The implications each class of mechanisms introduces on the architectural design as well as which mechanisms can be considered and implemented as extensions of others.
- Some other important considerations that have to do with the nature of the peer-to-peer video streaming applications as well as the timescales in which specific actions should be made.

The next step is the qualitative assessment of the proposed approaches. We need to provide the scenarios and the conditions that should apply in order for the mechanisms to be effective and result in a TripleWin situation. By TripleWin, we describe a situation where all the stakeholders, i.e. the end-users, the ISP and the overlay provider, are in a win (or at least in a non-lose) situation, as compared to the outcome without employing the proposed mechanisms; see also D2.1, Section 6. For example, locality promotion is useful when the population of peers inside a domain is above a critical mass. Otherwise, the local peer cannot do better than fetching chunks from peers belonging to another AS. In the same sense, the *IoP* approach may be appropriate for a wide range of swarm sizes (see also Section 11), but not bring the desired results when the swarm sizes are too small or too large. In the case of too small swarms, there might be limited performance improvement for the peers but from the ISP’s point of view such an action would not justify the costs incurred since the incoming traffic would remain the same (more or less) while the outgoing traffic would be increased; at the same time the ISP would incur the cost for the resources of the IoP. In the case of too large swarms, the IoPs may be proved not to be scalable enough to handle large swarms. Certain scenarios may have to do with the network conditions as well, as in the case of QoS/QoE mechanisms or *Dyn-Loc*, which will point out the benefits of such approaches

10 A Markov Model for the Evaluation of ETM Mechanisms on BitTorrent Swarms

The Markov Model presented in this section models a BitTorrent swarm, which is an overlay network per file. The purpose of the Markov Model is the analysis of certain properties of BitTorrent, such as scalability of performance, and evaluation of optimization approaches, such as insertion of ISP-owned Peer (IoP) in a BitTorrent-like network. By means of probabilistic analysis, the model estimates the distribution of the number of chunks downloaded by each given peer and other performance measures such as the upper tail of the distribution of the time require for a peer to complete downloading a file. For the purpose of analytical tractability, the model employs certain simplifications of BitTorrent; thus, it is expected that its outcomes will constitute bounds for the corresponding metrics of the actual BitTorrent.

In particular, the model's objective is to estimate the above performance measures for BitTorrent peers completion time in:

1. A pure BitTorrent network
2. A BitTorrent network where ISP-owned Peers are inserted; see Section 6
3. A BitTorrent network where some other optimization approach such as some form of locality awareness is employed.

Therefore, the model can be employed for studying a wide variety of questions, such as:

- Comparison of completion times in a pure BitTorrent network for different numbers of regular peers; *e.g.,* does there apply any monotonicity property, i.e. does performance improve with the number of peers? Or with the number of leechers?
- Comparison of completion times of regular peers when an IoPs is inserted; this study can show, *e.g.*, the extent of the performance improvement attained by employing an IoP or the tradeoff with respect to the associated upload bandwidth.
- Comparison of completion times of regular peers when multiple IoPs are inserted for different values of upload bandwidth (see below); this study can show whether and to what extent the addition of extra IoPs improves performance.

In following subsections, we present the basic assumptions, the rationale and preliminary equations that describe the evolution of the Markov Model. Analytical results, numerical calculations and conclusions derived are left as future study.

10.1 Basic Assumptions

Below, the main assumptions of the Markov Model are stated. The Markov model is a discrete time model; time is slotted: step 1, step 2, ..., step n, …Originally, we consider N+1 peers in the swarm; namely, N downloaders with initially 0 chunks and one seed which has all K chunks, that is the complete file. Moreover, for simplicity we assume that after a downloader finishes its downloading then it turns into a seed, namely it does not leave the swarm. (By introducing minor modifications to the formulation, the model can accommodate other assumptions at this point.) The population of peers remains constant. The complete state of the system would be specified completely by the set of chunks each peer possesses at step n, or respectively the number of peers out of N that has 1, 2, ..., or

K chunks at step n. However, due to symmetry among chunks in their initial distribution, it is sufficient to specify the number of file chunks that each peer has acquired until the end of each 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, *i.e.,*

$$\binom{N+K}{K} = \frac{(N+K)!}{K!N!}.$$

Due to the prohibitively large state space we resort to an approximation, which is motivated by the fact that due to symmetry the evolution of any given peer is same as that of each other peer. Let D be a tagged peer out of the set of the N downloaders. The state of D (as well as that of each other peer) belongs to {0, 1, ..., K}, where K is the total number of file chunks. We study the evolution of the tagged peer D. Due to symmetry; the equations derived for D can actually characterize each of the other downloaders. That is, at each time step n, the marginal distribution of the state of each peer is the same as that of D. The objective of the model is to derive this marginal distribution by exploiting this symmetry and introducing some approximations so that dealing with the limited state space of peer D is enough.

Next, we present our assumptions on the modelling of BitTorrent. First, 'tit-for-tat' is ignored. That is, it is assumed that each peer has the possibility for C unchokes. C is a parameter, the default value of which is 5, i.e. each peer can unchoke up to 5 other peers; these peers are randomly selected, which is similar to having 5 optimistic unchokes. However, we assume that at every step n, only 0 or 1 chunk can be downloaded by each peer, regardless of how many others have unchoked this peer. Additionally, no particular chunk selection method (*e.g.,* rarest first replication) is considered. All chunks that D is missing are considered useful and assumed to be sought simultaneously. If a peer that unchokes D has more than one useful chunk for D, and D decides to download from him, then D selects randomly and uniformly one of these useful chunks.

Due to the above assumptions, this Markov Model corresponds to a version of BitTorrent where all decisions are made randomly, and therefore is expected to have inferior performance compared to the original BitTorrent. Consequently, the results obtained by this model are expected to constitute bounds of the actual performance of the BitTorrent protocol.

10.2 Markov Model

First, we consider the pure BitTorrent scenario. In Figure 10.1, the evolution of the Markov Chain for a regular peer is depicted.

The transient marginal distribution of the state of a regular peer at step n is denoted $P(n) = [P_n(0), P_n(1), ..., P_n(K)]$, where $P_n(k) = \Pr[X_n = k]$. The transient distribution at step n+1 is calculated as follows:

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

where $P_{n+1}(1;k-1) = \Pr[unchoked\_AND\_useful \mid k-1]$ is a transition probability described as follows: peer D is unchoked and finds useful chunks in at least one other peer at step n+1, given that peer D has k-1 chunks at the end of step n, whereas $P_{n+1}(0;k) = \Pr[choked\_OR\_no\_useful \mid k]$ is a transition probability described as: peer D is choked by all peers or does not find a useful chunk by any peer that unchokes it, given that it has k chinks at the end of step n.

$$\begin{tikzcd}
X_{n-1}=k-1 \arrow[r] \arrow[dr] & X_n=k-1 \arrow[r] \arrow[dr] & X_{n+1}=k-1 \\
 & X_n=k \arrow[r] \arrow[dr] & X_{n+1}=k \\
 & & X_{n+1}=k+1
\end{tikzcd}$$

Figure 10.1: Evolution of Markov Chain of a regular peer.

It is plausible that in the setting we consider, there is a deterministic upper bound for the completion times of all peers (i.e. makespan), given that the population size is fixed. Due to the fact that the completion time of the last peers might be quite large, this upper bound is expected to be loose. Thus, we choose to assume as the proxy for completion time to be numerically estimated the time (step) *n** when a large portion of the peers' (say 95%) population should have finished downloading, that is they have K chunks. Using the transient marginal distribution of the state of D, it follows that *n** can be specified as follows: $n^* = \min\{n : P_n(K) > 0.95\}$[1].

Below, some basic approximations of the Markov Model are presented. The following sets of downloaders are non-overlapping. As a result, the random variables are taken to be independent.

Distribution of number $N_s(n)$ of downloaders that have K chunks at step n:

$$P(N_s(n) = x) = \binom{N-1}{x} (P_n(K))^x (1 - P_n(K))^{N-1-x}$$

Distribution of number $N_c(n)$ of downloaders that have 0<k<K chunks at step n:

$$P(N_c(n) = y) = \binom{N-1}{y} (1 - P_n(0) - P_n(K))^y (P_n(0) + P_n(K))^{N-1-y}$$

Distribution of number $N_e(n)$ of downloaders that have 0 chunks at step n:

$$P(N_e(n) = z) = \binom{N-1}{z} (P_n(0))^z (1 - P_n(0))^{N-1-z}$$

The evolution of the Markov Chain for a regular peer in a pure BitTorrent network is characterized by the following equations:

[1] Download completion times calculated by the discrete time Markov Model are expressed in units of "steps". In order to convert the completion times derived by the model to actual completion times expressed in msecs, we can multiply the number of steps by an average downloading time per chunk, which would equal the size of a chunk divided by the average end-to-end transmission rate within a swarm.

Step 0: $P(0)=[1,0,...,0]$

Step 1:

k=0 D is choked by the seed: $P_1(0)=P_0(0)P_1(0;0)=P_0(0)\left(1-\frac{C}{N}\right)$.

k=1 D is unchoked by the seed: $P_1(1)=P_0(0)P_1(1;0)=P_0(0)\frac{C}{N}$.

k≥2 $P_1(2)=P_1(3)=...=P_1(K)=0$

Step 2:

k=0 D is choked by the seed and the C peers that were unchoked in step 1:

$$P_2(0)=P_1(0)P_2(0;0)=P_1(0)\left(1-\frac{C}{N}\right)^{C+1}$$

k=1 D gets one chunk either at step 0, **or** at step 1:

$$P_2(1)=P_1(0)P_2(1;0)+P_1(1)P_2(0;1)=P_1(0)\frac{C}{N}\left(1-\frac{C}{N}\right)^{C}+P_1(1)\left(1-\frac{C}{N}\right)^{C}$$

k=2 D gets one chunk at step 0 **and** one at step 1, where it should be noted that if unchoked by another downloader the chunk thereof is useful to the D with probability (K-1)/K: $P_2(2)=P_1(1)P_2(1;1)=P_1(1)\left(\frac{C}{N}\left(1-\frac{C}{N}\right)^{C-1}+(C-1)\frac{C}{N}\left(\frac{K-1}{K}\right)\left(1-\frac{C}{N}\right)^{C-1}\right)$.

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. Note here that the number $N_s(n)$ of downloaders that become seeds is taken into account, because it influences the content among the remaining downloaders. Recall also that we have assumed that $N_e(n)$ and $N_s(n)$ are independent. It is important to distinguish cases here, since if n≥K then possibly some of the downloaders have already finished downloading and have started serving as seeds, *e.g.,* $N_s(n)\geq 0$. Of course, if n<K, then there are still N downloaders and only one (the original) seed in the swarm, *e.g.,* $N_s(n)=0$.

Step n+1:

k=0 D got no chunks at step n+1 given that it had no chunks until step n:

$P_{n+1}(0) = P_n(0)P_{n+1}(0;0) = P_n(0)\mathrm{E}_{N_e(n)N_s(n)}\left[\left(1-\frac{C}{N-N_s(n)}\right)^{N-1-N_e(n)}\right]$, where we make use of the distribution of $N_e(n)$ and the assumption that $N_e(n)$ and $N_s(n)$ are *independent*. Thus, probability $P_{n+1}(0;0)$ is written as:

$$\begin{aligned}
P_{n+1}(0;0) &= \mathrm{E}_{N_s(n)}\left[\sum_{z=0}^{N-1} P(N_e(n)=z)\left(1-\frac{C}{N-N_s(n)}\right)^{N-1-z}\right] \\
&= \mathrm{E}_{N_s(n)}\left[\sum_{z=0}^{N-1}\binom{N-1}{z}P_n(0)^z(1-P_n(0))^{N-1-z}\left(1-\frac{C}{N-N_s(n)}\right)^{N-1-z}\right] \\
&= \mathrm{E}_{N_s(n)}\left[\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N-N_s(n)}\right)\right)^{N-1}\right]
\end{aligned}$$

If n<K, then: $P_{n+1}(0;0) = \left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N}\right)\right)^{N-1}$, because $N_s(n)=0$ with probability 1,

otherwise: $P_{n+1}(0;1) = \sum_{x=0}^{N}\binom{N}{x}P_n(K)^x(1-P_n(K))^{N-x}\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N-x}\left(1-\frac{C}{N}\right)\right)\right)^{N-1}$.

k=1 D either got one chunk at step n+1 given that it had zero chunks **or** it got no chunks given that it had exactly one chunk:

$P_{n+1}(1) = P_n(0)P_{n+1}(1;0) + P_n(1)P_{n+1}(0;1) = P_n(0)(1-P_{n+1}(0;0)) + P_n(1)P_{n+1}(0;1)$.

The probability $P_{n+1}(0;0)$ is known due to prior calculations; that is, we make recursive use of prior calculated probabilities; thus, we only need to calculate $P_{n+1}(0;1)$. Therefore:

$$P_{n+1}(0;1) = \mathrm{E}_{N_e(n),N_s(n)}\left[\left(1-\frac{C}{N-N_s(n)}P_{n+1}(useful\_chunk;1)\right)^{N-1-N_e(n)}\right]$$

The probability $P_{n+1}(useful\_chunk;1)$ expresses the probability that D' finds a useful chunk given that it has 1 chunk and can be written as:

$$P_{n+1}(useful\_chunk;1) = P_n(1)\left(1-\frac{1}{K}\right)+\sum_{l=2}^{K}P_n(l)$$

where the latest term means that if another peer D' has two or more chunks, then D will find a useful chunk to download from D' with probability 1. Thus,

$$P_{n+1}(0;1) = \mathrm{E}_{N_s(n)}\left[\sum_{z=0}^{N-1}\binom{N-1}{z}P_n(0)^z(1-P_n(0))^{N-1-z}\left(1-\frac{C}{N-N_s(n)}\left(P_n(1)\left(1-\frac{1}{K}\right)+\sum_{l=2}^{K}P_n(l)\right)\right)^{N-1-z}\right]$$

$$= \mathrm{E}_{N_s(n)}\left[\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N-N_s(n)}\left(P_n(1)\left(1-\frac{1}{K}\right)+\sum_{l=2}^{K}P_n(l)\right)\right)\right)^{N-1}\right].$$

If n<K, then: $P_{n+1}(0;1)=\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N}\left(P_n(1)\left(1-\frac{1}{K}\right)+\sum_{l=2}^{K}P_n(l)\right)\right)\right)^{N-1}$, because $N_s(n)=0$ with probability 1, otherwise:

$$P_{n+1}(0;1)=\sum_{x=0}^{N}\binom{N}{x}P_n(K)^x(1-P_n(K))^{N-x}\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N-x}\left(P_n(1)\left(1-\frac{1}{K}\right)+\sum_{l=2}^{K}P_n(l)\right)\right)\right)^{N-1}$$

k=2,…,K D either got one chunk at step n+1 given that it had m-1 chunks **or** it got no chunks given that it had m chunks:

$$P_{n+1}(k)=P_n(k-1)P_{n+1}(1,k-1)+P_n(k)P_{n+1}(0,k)=P_n(k-1)(1-P_{n+1}(0,k-1))+P_n(k)P_{n+1}(0;k).$$

The probability $P_{n+1}(0;k-1)$ is known again due to prior calculations for value k-1; thus, we only need to calculate $P_{n+1}(0;k)$. Therefore:

$$P_{n+1}(0;k)=\mathrm{E}_{N_e(n),N_s(n)}\left[\left(1-\frac{C}{N-N_s(n)}P_{n+1}(useful\_chunk;k)\right)^{N-1-N_e(n)}\right].$$

The probability $P_{n+1}(useful\_chunk;k)$ expresses the probability that D' finds a useful chunk given that it has k chunks and can be written as:

$$\begin{aligned}P_{n+1}(useful\_chunk;k)&=P_n(1)\left(1-\frac{1}{K}\right)+...+P_n(k)\left(1-\frac{k!}{(K-k+1)!}\right)+\sum_{l=k+1}^{K}P_n(l)\\&=\sum_{m=1}^{k}P_n(k)\left(1-\frac{m!}{(K-m+1)!}\right)+\sum_{l=k+1}^{K}P_n(l),\end{aligned}$$

Particularly, the term $P_n(k)\left(1-\frac{k!}{(K-k+1)!}\right)$ expresses the probability to find a useful chunk from another peer with n chunks, n≤k, which is the probability of that peer being in that state multiplied by the probability to find a chunk different from the k chunks that the tagged peer D already has. This expression is also used in [QS04]. In the latest equation, the last term means 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. Thus,

$$P_{n+1}(0;k)=\mathrm{E}_{N_s(n)}\left[\sum_{z=0}^{N-1}\binom{N-1}{z}P_n(0)^z(1-P_n(0))^{N-1-z}\left(1-\frac{C}{N-N_s(n)}\left(\sum_{m=1}^{k}P_n(k)\left(1-\frac{m!}{(K-m+1)!}\right)+\sum_{l=k+1}^{K}P_n(l)\right)\right)^{N-1-}\right.$$

$$=\mathrm{E}_{N_s(n)}\left[\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N-N_s(n)}\left(\sum_{m=1}^{k}P_n(k)\left(1-\frac{m!}{(K-m+1)!}\right)+\sum_{l=k+1}^{K}P_n(l)\right)\right)\right)^{N-1}\right].$$

If n<K, then: $P_{n+1}(0;k)=\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N}\left(\sum_{m=1}^{k}P_n(k)\left(1-\frac{m!}{(K-m+1)!}\right)+\sum_{l=k+1}^{K}P_n(l)\right)\right)\right)^{N-1}$,

because $N_s(n)=0$ with probability 1, otherwise:

$$P_{n+1}(0;k)=\sum_{x=0}^{N}\binom{N}{x}P_n(K)^x(1-P_n(K))^{N-x}\left(P_n(0)+(1-P_n(0))\left(1-\frac{C}{N-x}\left(\sum_{m=1}^{k}P_n(k)\left(1-\frac{m!}{(K-m+1)!}\right)+\sum_{l=k+1}^{K}P_n(l)\right)\right)\right)$$

Figure 10.2 shows the rationale of the iterative process of the calculation of the transient probabilities:

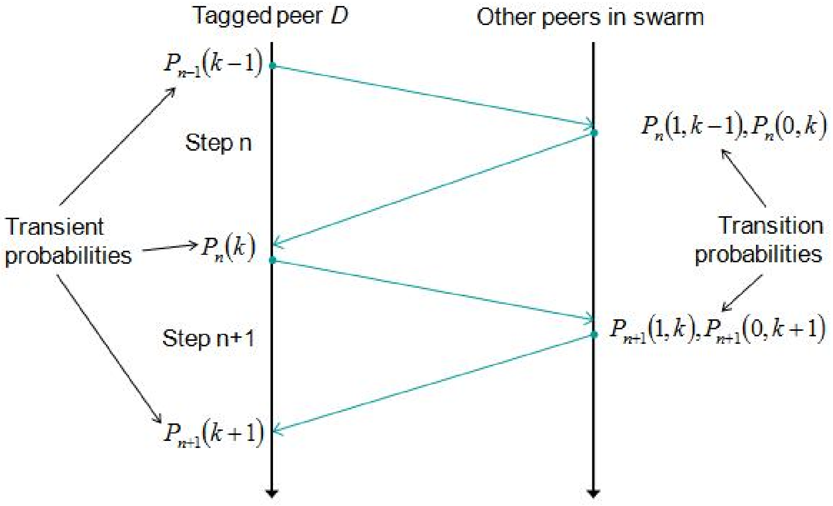


Figure 10.2: Iteration.

10.3 Future Work

Work on this model will continue with the calculation of measures related to completion times based on the Markov Model. These calculations will be performed in Matlab. Possible scenarios to be evaluated in terms of performance are:

- N regular downloaders and 1 seed
- N+1 regular downloaders and 1 seed
- N+M regular downloaders and 1 seed

After assessing performance for the case of a pure BitTorrent network, we consider the insertion of one IoP in the specific system that has already been analyzed. The IoP is not an intervening cache but runs the overlay protocol in parallel with regular peers (see Section 6). The IoP is assumed to be equipped with high download/upload capacity, *e.g.,* C’≥C, D’≥D. Since ‘tit-for-tat’ does not apply for seeds, unchoking is based on the highest downloading rate criterion. This happens because the seeds are not interested in downloading any new chunks – they have all the file chunks; as a result they unchoke the downloaders with the highest downloading rates in order to assure the fastest dissemination of the file. Thus, we can assume that the IoP will always be unchoked by the seed due to its large download bandwidth. Additionally, even though ‘tit-for-tat’ is not assumed in our Markov Model even for regular peers, the IoP is assumed to be unchoked due to its large upload capacity. Besides extra bandwidth, extra number of unchokes is also considered here. Therefore, evolution of the Markov Chain for the IoP is quite simple. The IoP may reach up to D’ chunks at every step, until step (K-1)/D’, when it also becomes a seed. Afterwards, it remains active in the swarm serving other peers. The trade-off here lies between the fact that the IoP initially consumes network resources, (and thus for the tagged peer C → C-1 for the seed, while N → N+1 in the contention for chunks of other peers), and the fact that after the IoP completes its downloading, it serves other peers boosting overall performance in terms of completion times.

Furthermore, calculation of measures related to completion times based on the Markov Model are also planned for the following scenarios to evaluate the IoP:

- N regular downloaders and 1 IoP with C’≥C and 1 seed
- N regular downloaders and M IoP with a total of C’≥C and 1 seed
- N regular downloaders and M IoP with C’≥C each and 1 seed

Comparing completion times for the above cases can help derive conclusions on how completion time is affected by:

- The number of regular downloaders
- The number of IoPs
- Different values of the unchoking capacity C’ of the IoP(s) and the resources totally dedicated to IoPs. For example, it is preferable to dedicate all resources to a single IoP, or divide them among several such entities?

11 Measurement Analysis of BitTorrent Swarms

The main objective of this study is to investigate typical sizes of BitTorrent swarms, in particular the number of leechers and seeds per swarm. This analysis has a direct influence of the applicability and efficiency of ETM mechanism, which will be illustrated by the example of locality promotion and ISP-owned peers (IoP).

In addition, we want to describe the temporal evolution of the swarm sizes which can be used furthermore to model file request arrivals or user arrivals in swarms. The dynamics of the swarms, i.e. the variation of the population sizes over time, also gives a hint if it is sufficient to use average values or if we need more complex models. This is especially useful for theoretical investigations when we need to abstract real traffic patterns or user behavior to get tractable mathematical models.

11.1 Measurement Description

In a BitTorrent-like system, each separate piece of content (*e.g.,* movie, archive or document) has its own overlay, the so-called swarm, where it is distributed. To estimate the effectiveness of ETM measures in one such swarm, its size has to be known. Therefore, we present here a study of typical swarm sizes of BitTorrent swarms, in particular the number of lechers and seeds per swarm. In total, we measured 63,867 BitTorrent swarms offering video contents, movies, TV series, or documentary. As we sequentially measured the number of seeders and leechers of all swarms, it took about 23 minutes to get data for each swarm. This means, every 23 minutes we obtain one measurement sample per swarm. The measurements were conducted for the duration of three days. In order to increase the granularity of our measurement results, especially to describe the dynamics of a swarm, we traced the most popular movies individually. This allows us to obtain the swarm size every 10 seconds for the large swarms.

11.2 Measurement Results for BitTorrent Swarm Sizes

Figure 111.1 shows results from this study. On the x-axis, the x% of the largest swarms are given, i.e. the swarms are sorted according to their population size, while the y-axis shows the cumulated percentage of total peers which belong to the x% of largest swarms. It shows that a Pareto-principle [Jur54] governs the total peer distribution: a large share of the peers can be found in a small number of swarms. This has several consequences for the ETM mechanisms employed in such a system. Depending on the swarm size an IoP is participating in, its optimal position and allotted resources vary. The same holds for the selection of highly active peers (HAPs). Moreover, the decision in which swarms to participate is important for the mechanism. From the view point of an ISP, the best would be to participate in swarms with large number of local peers, which allows to cache popular contents, to serve many local peers, and thus to achieve a reduction of inter-domain traffic. On the other hand, the increase of upload capacity by adding one or a small number of IoPs to a large swarm can possibly be negligible. Within the SmoothIT project, a detailed traffic analysis of (a) the size of swarms, (b) the volume size of the content offered, and (c) the emerging traffic amount will be provided in Task T1.3 “Traffic Analysis of Overlay Applications”. Interesting questions that have to be addressed are: (i) Do larger files generate larger swarms due to longer download times? Are there any correlations? (ii) What is the actual ratio between the amount of inter-domain and intra-domain traffic within a swarm? What is the correlation between the swarm size and this ratio? (iii) What is the ratio between seeders and leechers in a swarm?

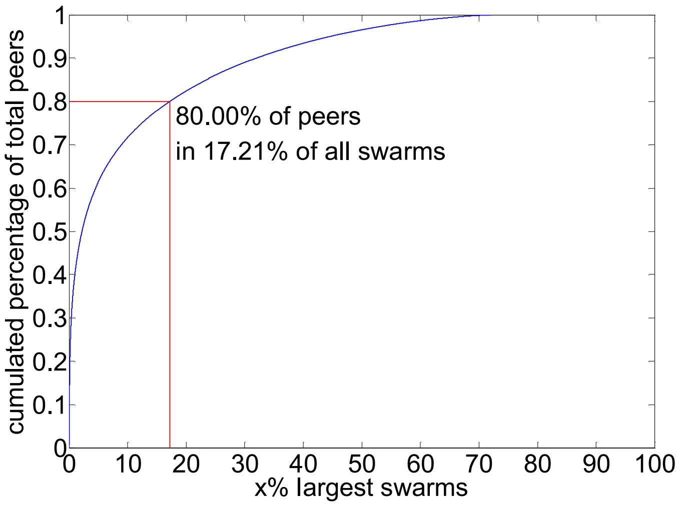
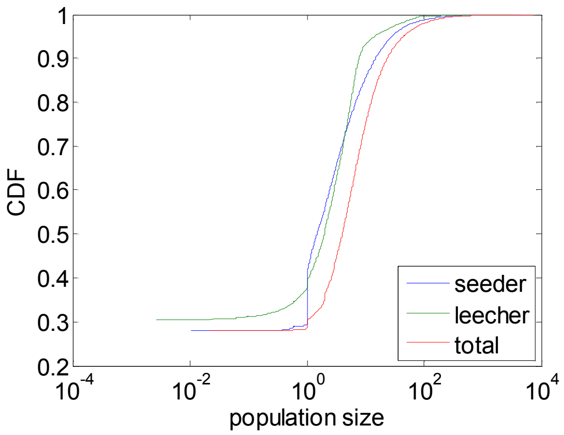


Figure 111.1: BitTorrent Swarm Sizes.

This information about content popularity and swarm sizes can be collected by an ISP without cooperating directly with the overlay. One way to do this would be to randomly join swarms with the IoP, meter the effect achieved by adding the IoP to this swarm, and leave again if it is not large enough. Still, the overlay can contribute greatly by offering this information to the decision ISP, allowing for a better strategy on which swarms to join before actually doing so.

For locality promotion, the impact of these results is different. For this mechanism, it is important that swarms are large in order to provide local alternatives for remote neighbors. If a swarm is too small, only few peers are actually in the same network, prohibiting an overlay restructuring due to a lack of choices.



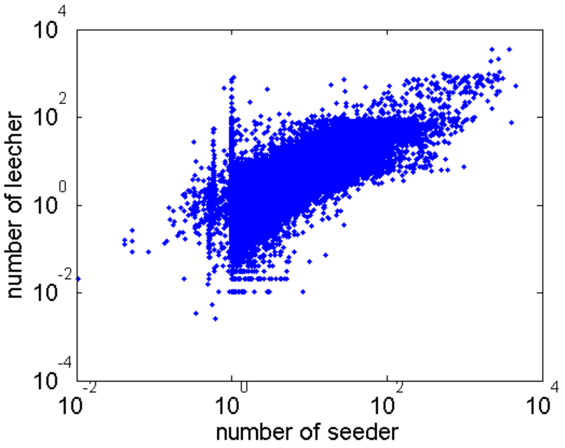


Figure 111.2: Cumulative distribution function of the leecher and seeder population of BitTorrent swarms.

Figure 111.3: Scatter plot of the number of seeder and leecher of a BitTorrent swarm.

Figures 11.2 and 11.3 show more results on the measurement of the BitTorrent swarm sizes. Since we observe the individual BitTorrent swarm size over time for a measurement period of two days, we use the average value of the measured time series. For this reason, we may obtain real-valued population sizes. In Figure 111., the cumulative distribution function of the leecher and seeder population, as well as the total population which is the sum of leechers and seeders is given. It has to be noted that the CDF of the total population corresponds to the curve in Figure 11.1. The results show that there are a lot of small swarms, which have a total population *X* below 100 peers. The probability, that the population of any of the measured swarms is below 100 peers, is close to one, i.e. *P(X<100)~1*. While the average population size is 15.53 peers, the median of the population size is 4 peers. In addition, 17861≡ 27.97% of the swarms don't have any seeders. Thus, none of the peers has a complete copy of the file. In this case, an ETM approach like IoP would be beneficial to increase the availability of the content for such swarms. Furthermore, 19357 swarms do not have any leechers which means that the peers are (a) either no more interested in that content or (b) are not able to download the file in this swarm and give it up.

However, it has to be noted, since the x-axis is logarithmically scaled, that there are some large swarms. From the largest 95 swarms, we observe that in 47 swarms the number of leechers is limited to [990;1000], which is likely caused by the used tracker implementation. Only 14 swarms are larger than 1,000 peers. However, more measurement studies have to be performed in this direction to get statistical significant data. It has to be noted that we have focused here only on video contents.

Figure 111. shows a scatter plot of the number of seeders and leechers in a BitTorrent swarm. Obviously, there is a strong correlation between the number of seeders and leechers per swarm. This reflects the popularity of the offered contents and the user behaviour. If content is popular, i.e. there are many leechers, there are also many peers willing to share this, possibly because they have an attitude of contributing to the overall community of peers. This might also be caused by letting the BitTorrent client run in the background. The coefficient of correlation between the number of seeder and leecher computes as *COR(#seeder,#leecher) = 0.7067*. This allows to estimate the number of leechers per swarm when the number of seeders is known for example. Regarding our ETM mechanisms, this means that we only need to monitor the amount of seeders. This can be done for example by monitoring and requesting the tracker about the number of seeders, or by estimating the number of leechers by measuring the arrival rate of new requests to specified peers (*e.g.,* to an ISP-owned peer).

Figures 11.4 and 11.5 focus on the temporal evolution of the BitTorrent swarms. Figure 111. shows the evolution of population sizes over time for some top swarms. However, the temporal evolution strongly depends on the actual swarm. For the green and the red curve, a daily behavior can be observed. In the afternoon, the swarm size increases which might be either caused by people starting their BitTorrent client, reflected by an increased number of leechers and/or seeders, or by finished downloads reflected by an increased number of seeders. However, the current ratio of leechers to seeders might strongly vary, which can be seen from the red and the blue curve for example. Figure 111. shows the standard deviation of the swarm size, obtained from the 23 minutes measurement samples per swarm over the three days. On the x-axis, the swarms are sorted by the total population size in decreasing order. It can be seen that there is a clear decreasing relationship between the standard deviation and the population size which can be taken into account to model swarm sizes.

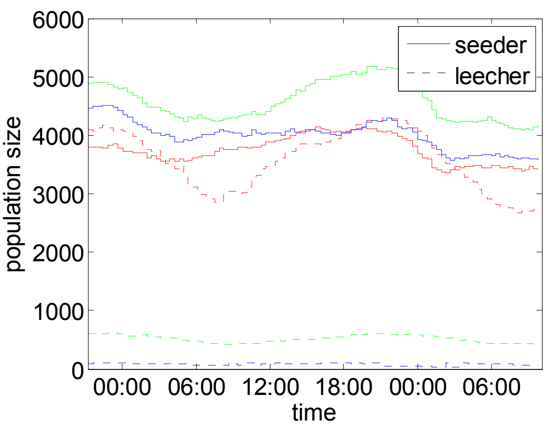


Figure 111.4: Evolution of population sizes over time.

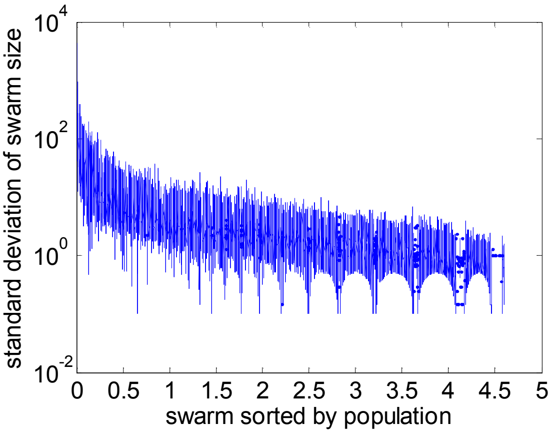


Figure 111.5: Standard deviation of swarm size sorted by population size.

As a result of this section, we have seen that there are many small BitTorrent and few very large BitTorrent swarms. As a consequence, BitTorrent swarms of all sizes contribute significantly to the overall BitTorrent traffic, and for an ETM to work efficiently it has to tackle swarms of – if possible – all sizes. Locality promotion potentially reduces the inter-domain traffic for large swarms. However, there are a critical number of peers per swarm required within an ISP's network in order to successfully use locality promotion for achieving a substantial reduction in inter-domain traffic without simultaneously decreasing the overlay's performance and thus violating the win-win-win maxim of ETM.

11.3 Conclusions and Future Work

The key idea of an IoP is to serve many local peers to achieve inter-domain traffic reduction and thus to save costs. However, the IoP provides only a viable solution for small to medium swarm sizes. In that case, the available uplink capacities can be notably increased by inserting an IoP into a small swarm. For larger swarms, the additional capacities offered by an IoP may only have a minor effect.

However, the number of swarms an IoP can support simultaneously is limited mainly by the available storage. Supporting many very small swarms addresses many content files and peers. This means that a large amount of storage capacity is required, which leads to an increased amount of CAPEX. As a consequence, the achieved inter-domain traffic reduction per required storage capacity i.e. per CAPEX may shrink dramatically with the swarm size. Therefore, the application of IoP may be useful for swarm to medium sized swarms only.

Now, the charging model comes into play when evaluating the usefulness of the ETMs. Our exemplary charging scheme, the 95th percentile rule, is sensitive to asymmetric changes only while symmetric changes have no direct impact on the costs. Locality promotion does not affect this difference while the IoP clearly shifts the traffic difference in the favor of A deploying the ETMs When interpreting A as a Tier 3-ISP and B as a Tier 2-ISP the use of an IoP is beneficial for the Tier 3-ISP. Whether there is an actual monetary benefit, on the one hand depends on the OPEX and CAPEX for the ETM and on the other hand on the exact charging scheme and the achieved traffic reduction which again depends on the swarm sizes.

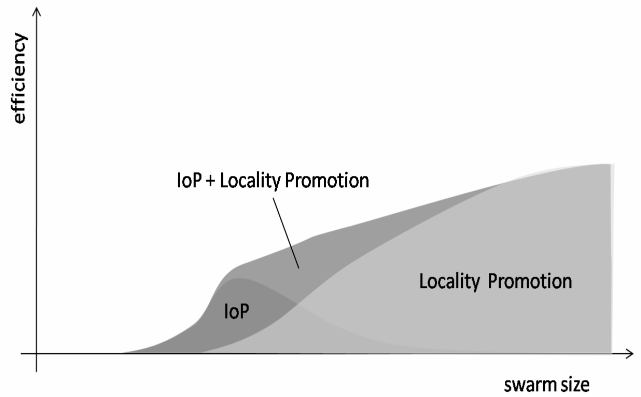


Figure 111.6: Qualitative illustration of inter-play of Locality Promotion and IoP.

For an efficient operation of ETMs we propose a combined and purposeful usage of ETM mechanisms depending on the present overlay application and its characteristics. As an example, we have already explained that the IoP is expected to be more effective with medium-size swarms. Its efficiency can be improved by a more directed use which is enabled by a SIS like architecture that resolves the information asymmetry. Figure 111. gives a qualitative impression on the regions of operation for the different ETM mechanisms. A quantitative statement on the efficiency of IoP and locality interplay is still subject to further studies.

As additional future work, we have to derive the actual traffic in these swarms. For typical swarms, we can perform active measurements by joining the swarm to get the IP addresses of the participating peers. Then we can derive the location of the peers in the swarms a) to evaluate whether it is possible to keep traffic locally and b) to evaluate possible cost savings by reducing inter-domain traffic, while considering also the impact on the download performance. The outcome of this study is the effect of an ETM approach to a range of cases. Since these evaluations cannot cover all possible cases, its results may not lead to a complete and detailed model of the effect of ETM on all possible swarms. However, it will provide a general overview on these effects and some basic dependencies to be taken into account when selecting which ETM approach to employ or fine tuning its parameters.

12 Simulation Framework for the Analysis of ETM Mechanisms

In order to be able to conduct a comparative performance evaluation of the different theoretical approaches described in this document, a common simulation framework is being created. In this section, we describe the planned usage of this simulator, the simulation model used as a basis for its implementation, and the first scenario that will be implemented and tested. This scenario merely provides a common basis on which the more sophisticated approaches can be built upon in the following phases of SmoothIT.

Therefore, the described version of the simulator should be seen as a start to begin covering all of the scenarios and approaches described in this document. These scenarios, strategies (*e.g.,* for peer selection), overlay components and models may be added to the simulator later on. It is built with extendibility in mind; however, we concentrate on its current status here. In general, any modification to these scenarios is possible, with different degrees of additional overhead depending on how far the modification differs from the common framework.

The simulator developed in this project is based on the abstract simulation framework Protopeer [PRO08]. It was selected because it offers a built-in logical distinction between the overlay logic and the network model, cf. Figure 12.1. This allows the exchange of any of these layers without modifying the other. This goes as far as allowing the usage of a real network instead of a network simulation or emulation. In any case, it offers a useful interface for message exchange between peers, which are organized into peerlets according to their functionality.

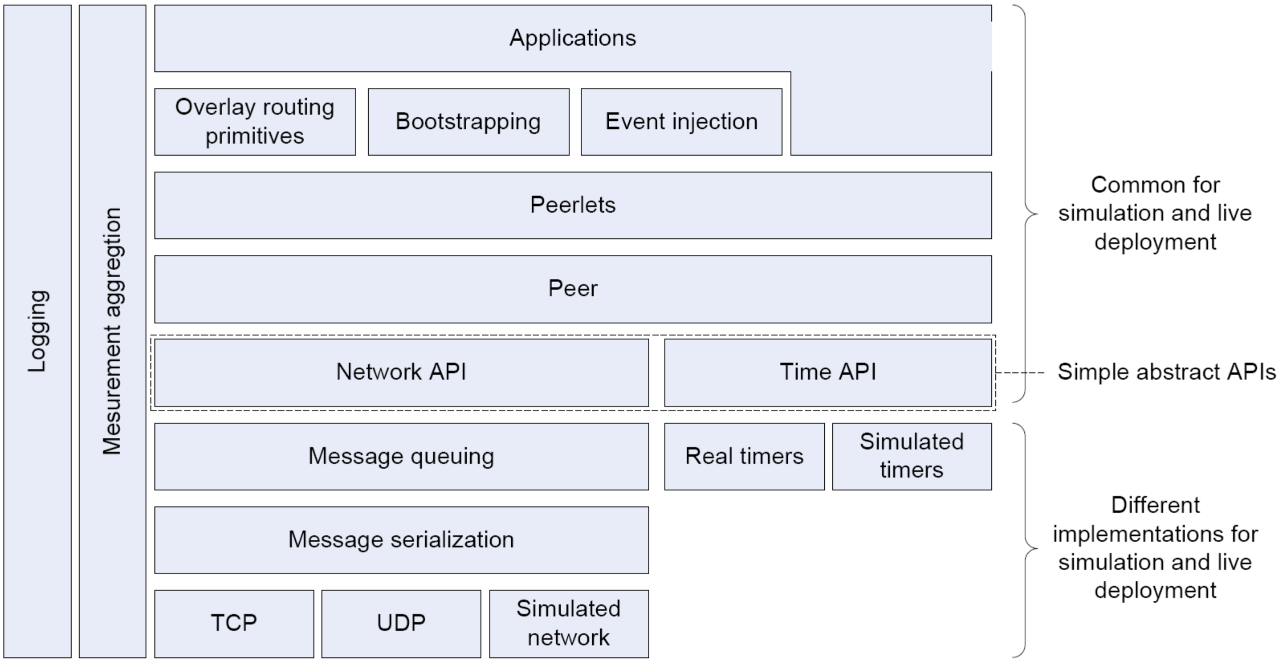


Figure 12.1: Protopeer framework structure (Source: [PRO08]).

Apart from that, Protopeer offers measurement infrastructure, saving the effort of implementing that from scratch. The same measurement functionality can be used in all the different network models including real deployment in, *e.g.,* Planetlab.

12.1 Objectives

The aim of developing this simulator for the project is to be able to evaluate the currently described and future ETM approaches by means of simulation experiments.

Additionally, parameter studies for the architectures developed in SmoothIT can be conducted, allowing also for a better understanding of the specific architecture chosen to be implemented in the trials. While this deployed system allows for measurements, the number of possible scenarios that can be included in simulations is much larger and it is easier to quickly adapt mechanisms or parameters.

Apart from that, simulation allows for an evaluation of larger systems than it would be possible in a trial setup. While the external trial might lead to overlays in the same order of magnitude as simulated, the gathering of measurement data and their evaluation is much more efficient in a simulation environment. Moreover, these results should be available much earlier, thus allowing for fine-tuning of the approaches to be implemented in the trials.

Since WP2 does not only limit itself to studying the ETM approach/architecture option that is implemented, but also aims to generate evaluation results about other approaches, the simulator should provide the opportunity to do so, with all the advantages mentioned above.

12.2 Simulation Model

In this section, we describe the theoretical assumptions and models that are realized by the simulator. They show which abstractions are made in simulation scenarios and which mechanisms are simulated in detail.

12.2.1 Network Model

The main goal of the network-layer model is to simulate the general behavior of the underlay. To be able to simulate larger swarms, the network model will not include packet level details. Instead, traffic flows in the physical network will be modeled as flows characterized by delay and available bandwidth. The time it takes for a message to reach its destination therefore depends on the network delay as well as on the size of the message.

The bandwidth available for a flow is computed from the capacity of the end-to-end connection and the other flows of the peers in question, depending on the assumption of the network bottleneck (bandwidth sharing). Two of the approaches to this problem are described in [PBC06] and [GB02].

The topology used to derive the delay and bottleneck for an end-to-end connection is abstracted from a real network by only explicitly regarding the access links of the peers and the inter-AS or transit links. The assumption here is that intra-AS links do not experience bottlenecks and interesting measurement data is primarily gathered for the two types of links mentioned above. No routers or network equipment in general is included in detail in the model. As a result, the basic unit in any topology simulated is an AS with attached POPs, where in turn the peers are located. This also means that no queueing delay is simulated. It may however be modeled as part of the network delay in general.

The routing between different AS is currently assumed to be static. No routing decisions have to be made during the simulation; changes in the routing may however be introduced manually. The addition of an automated routing procedure at a later point is possible.

12.2.2 Overlay

The overlay structure and functionality is currently modeled after the Tribler peer-to-peer VoD application that was chosen as the test case of SmoothIT. Therefore, a tracker is included in the model. Additionally, an SIS server is part of the scenarios that employ the according SmoothIT solution.

The peers partaking in the overlay are placed in the underlay on POP level, i.e., by the access network they use. They have a certain up- and download bandwidth as well as an access type. Currently, the simplest type of access, i.e., a wired network like DSL, is included. Other access types may be added as necessary.

To exchange data with other peers, the tracker or an SIS server, peers use messages that are transported by the underlying network model. These messages include signaling traffic as well as data transfer. The communication in the overlay is modeled in detail insofar as the information exchanged is concerned. Protocols or message formats are not modeled, the focus lies on the semantics of messages, as explained below. This means that all messages that are currently part of the overlay communication and the modeled overlay-SIS communication are implemented with the information they contain. An example would be the bitmaps exchanged between peers to signal the chunks they are able to share, as well as the update messages exchanged whenever this status changes.

Peers can join and leave the network during the simulation, allowing for the inclusion of churn and its effect into the evaluation.

The user data that is distributed via the overlay, i.e., the video, is also abstracted. No real video data is exchanged. The frames that are the smallest unit in a video and that make up the payload of data packets are characterized by their size, type (i.e., I-frames, P-frames and B-frames) and their position in the video (their timestamp). This data is generated from real videos.

This video data is partitioned into chunks and blocks according to the mechanisms of Tribler. Each peer has a list of the blocks and chunks it has stored locally in order to simulate the overlay data distribution process in detail. The peer and chunk selection mechanisms are modular in order to allow for the evaluation of different scenarios, currently the according algorithms from Tribler are implemented.

As a consequence, also the playing out of the video has to be modeled, since the chunk selection of, *e.g.*, Tribler uses the current play-out position of the local peer in its selection criteria. Therefore, this playout position can be accessed per peer at any time in the simulation, also depending on the play-out strategy used; *e.g.,* stalling when no data is currently available locally.

While Tribler is the primary application chosen for evaluation, the simulator is flexible enough to support other systems, *e.g.*, BitTorrent, by adapting the according peer behavior. Since the peer functionality is separated from the network simulation, completely different overlay are also possible, however with an accordingly higher additional implementation effort.

12.2.3 Simulation Control

The simulations will be parameterized by a number of input files in XML-format. These are used to initialize the network topology, the peer population (including tracker and SIS) and the video file that is shared. Additionally, the measurement objects are created. The current parameters that can be influenced by these input files are listed in the next section.

12.3 Scenarios and Experiment Design

The first SmoothIT ETM approach that will be implemented in the simulator is the concept of a central SIS server that is contacted by the peers in its own AS to receive underlay information about their neighbors. This is in line with the architectural description contained in [D3.1].

The initial version of the algorithm applied by the SIS basically makes a binary decision: a neighbor on the list received from the querying peer is either in the same AS as this peer or it is not. It is assumed that the information about the AS affiliation of a peer can be directly derived from its IP address; therefore this decision can be made locally, without contacting other entities. The implementation of this algorithm will be flexible enough to allow for an exchange with more sophisticated methods later.

This SIS server will be placed in a reference topology to be able to compare other approaches under the same network circumstances. The current definition of this topology is depicted in Figure 12.2. Note that the number of POPs and peers is not reflecting the actual size of the overlay, while the shown access networks are just for illustration. The simulator is flexible enough to allow other topologies; a common scenario to reference nevertheless provides the grounds for direct comparison. Thus, results generated for different mechanisms should include one scenario with this topology if they should be evaluated in comparison to others.

This scenario includes a multi-homed AS that will be in the focus of the evaluations (‘Local Tier 2 AS’). Apart from forwarding traffic via Tier 1 ISPs, it also has a connection with another Tier 2 AS which can be used to test scenarios where a peering agreement between two Tier 2 AS. Finally, end users in remote AS can be reached via the Tier 1 ISPs. This should cover the most important aspects for the different ETM approaches.

Finally, we give a list of currently configurable parameters of the simulation below. The parameters in italics currently have no effect. Apart from the values given here, the topology of the network is also specified by the structure of the input file.

Network topology:

- Number of AS
- Number of Inter-AS links
- Number of POPs
- Number of different Access Types
- AS
    - Size
    - Delay

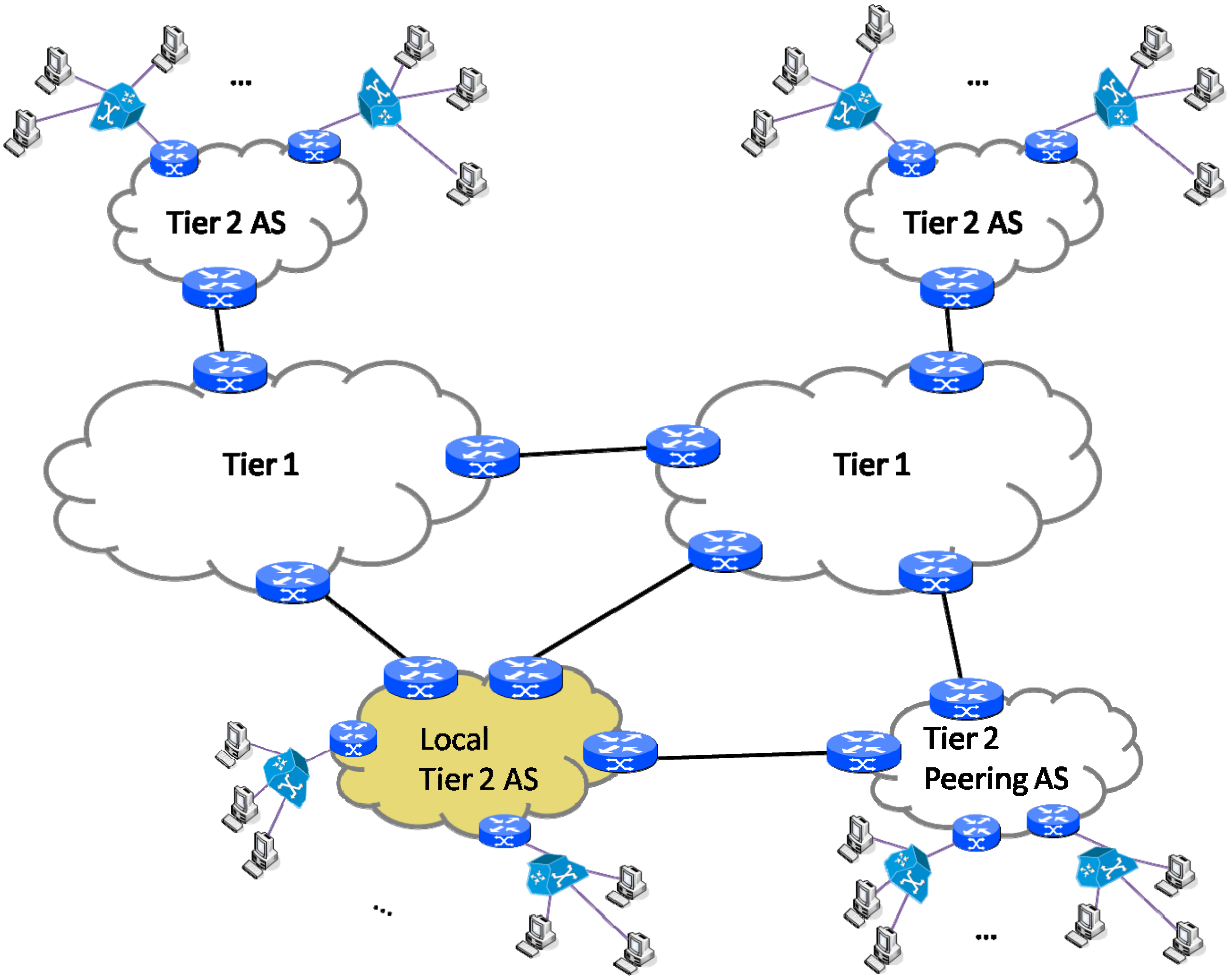


Figure 12.2: Reference topology.

- Inter-AS link
  - Bandwidth
  - Delay
  - *SLA*
  - *Background traffic*
- POP
- Access Type
  - Delay
  - Downlink capacity
  - Uplink capacity
  - Mode

Overlay parameters:

- SIS
  - POP

  - Algorithm
- Tracker
  - POP
  - Algorithm
- Peers
  - AS
  - Number
  - Class
- Peer Class
  - Application
  - Chunk selection algorithm
  - Peer selection algorithm
  - SIS usage
  - Online time distribution
  - Interarrival time distribution
  - Default Number of Neighbors
  - Seeder flag
- Data
  - Chunk size
  - Block size

Video:

- Frame rate
- *Resolution*
- *Number of Layers*
- Layers
  - Frames
    - Size
    - Type

12.4 Conclusions and Future Work

The simulator described in this Section will provide a tool to conduct performance evaluations of the different ETM mechanisms in a range of different scenarios. It allows to compare the effectiveness of these mechanisms under varying circumstances, e.g., the overlay size, the share of peers participating in the ETM scheme, the underlay topology, etc .To this end, the currently identified important parameters and concepts are part of the simulation model and implementation. There are already now a large number of configurable parameters, which is expected to be expanded over time. The expansion of the simulator with new functionality and supported simulated technologies is an ongoing effort. As new mechanisms and architectures are generated in the course of the project, they may be included in the range of scenarios that can be configured for simulations.

The ETM mechanisms may be evaluated with respect to the traffic they cause on inter-AS transit links or other underlay measures, as well as user-perceived quality like stall times. Also, the effect of ETM parameters on these performance indicators can be studied.

13 Summary and Conclusions

This deliverable presents a selected variety of innovative ETM approaches aiming at the optimization of the traffic impact of overlay applications on underlay networks, while offering monetary and/or performance-related incentives to *all* players involved. For each approach associated components and required intelligence are described. Based on a detailed classification of ETM approaches proposed, a study of their relations is provided, both at the conceptual and the architectural level, having as a yardstick the SmoothIT architecture of [D3.1], and a roadmap for the future development of the ETM approaches.

Three major properties characterize an ETM approach and determine how it is classified: whether it defines an *Information Exchange* mechanism, whether it includes certain *Traffic Management* techniques, and whether it introduces a new *Architectural Component or Interface* or a new/enhanced *Overlay Entity* (such as the ISP-owned peer, i.e. IoP). The majority of those ETM approaches proposed are based on the SmoothIT Information Service (SIS) architecture, with some of them providing basic functionalities (such as locality promotion) and others offering extensions and enhancements that strengthen the notion of ETM. The rest of these approaches are decentralized and show hybrid characteristics. Therefore, the classification and roadmap of these ETM approaches have also led to the identification of most prominent combinations, namely: SIS-based locality promotion (or an enhanced version thereof), possibly combined with a self-organization mechanism in the overlay and/or the IoP for achieving more active intervention of the ISP in ETM, and with a QoS/QoE-related mechanism for enforcing performance objectives.

The effectiveness of an ETM approach is influenced by several factors, such as the overlay application and the type of content (*e.g.*, files or video) shared thereby, the size of the application-level swarm, the charging scheme mainly for inter-ISP traffic, which is a major source of costs for an ISP. To this end, the distribution of swarm sizes and the charging schemes associated with agreements among ISPs have been studied. The measurement-based investigation of the sizes of actual BitTorrent swarms in the Internet lead to the main conclusion that a Pareto principle governs the total peer distribution: a large share of peers can be found in a small number of swarms, while there are many small swarms. As with tariffs for inter-ISP traffic, the focus is laid on an instantiation of the commonly used 95th percentile rule employed under transit agreements, which had been investigated with respect to its sensitivity to different parameters.

Those solutions to be chosen for suitable ETM will have to take all of the aforementioned parameters into consideration. For instance, in case of a symmetric traffic reduction, which might be a result of a locality promotion mechanism, this may result in minor reduction of the charges. Moreover, a locality promotion is best suited for large swarms, as they provide sufficient possibilities to change the overlay topology so as to reduce inter-domain traffic; this effect can be enforced by combining locality promotion with content promotion, the effect of which is to create and maintain large swarms. For small to medium swarms, the IoP is a promising approach; its efficiency can be improved by a more directed use enabled by the SIS, so as to resolve the information asymmetry. Another important dimension to influence the selection of ETM approaches is the appropriateness of each such approach for video-on-demand (VoD), which has already been selected by SmoothIT as the application for both internal and external trials, and/or for file-sharing, which determines a fallback solution for the external trial. Overlay video streaming applications introduce much stricter QoE requirements than file-sharing ones do. In particular, the play-out buffer should always be full, while playing the video should continuously be advancing, in order to minimize jitter and the possibility for unacceptable QoE. All ETM approaches based on QoS/QoE currently involve the provision of strict QoS guarantees; whether service differentiation would be sufficient for them is a matter that could be studied in the future. Moreover, “Content awareness” may prove to be helpful to deal with the aforementioned characteristics of VoD, by affecting peer and chunk selection based on the current content of the play-out buffer.

The work documented in this deliverable has considerably advanced the project’s understanding of Economic Traffic Management principles and led to a rich design space of those ETM approaches and to a study of their relations. Nevertheless, for the assessment of the proposed approaches (such as effectiveness or complexity), there is important future work to be accomplished. In particular, it is necessary that SmoothIT provides those scenarios and conditions, which should apply in order for each such approach to be effective. This will result in a situation, where it is beneficial for all players involved, taking into account the complex interaction between the overlay application, the charging scheme, and the ETM approach itself. Furthermore, future work will also be pursued in several other directions, the conclusions of which will be taken into account for the assessment of these and possibly other ETM approaches. In particular, future work in the direction of interconnection charging should focus on these conditions and parameters that affect mostly the 95th percentile rule, and validate all results through analysis of real traffic traces. Furthermore, SmoothIT has to investigate overall costs, when keeping traffic locally; on one hand, there is a reduction of inter-domain traffic charges, but on the other hand, there may be an increase in the number of “internal” hops or of resources (*e.g.*, IoPs or SIS) and their cost. These studies will have a close interplay with investigations on actual swarms, which in the future will focus on the actual traffic they generate. The derivation of the location of peers in these swarms will follow to evaluate, whether it is possible to keep traffic locally and to evaluate possible cost savings by reducing inter-domain traffic, while considering also the impact on the download performance.

Note, however, that an integral part of the assessment of ETM approaches is the analysis by means of theoretical models and an experimental study by means of simulations, both of which are also already in progress. Regarding analytical studies, this deliverable presented an innovative Markov model of a BitTorrent swarm, to enable an analysis and evaluation of optimization approaches, such as insertion of ISP-owned peers in a BitTorrent-like network. Since the state-space of the system is prohibitive, the model resorts to certain approximations and is numerically tractable. This work will be continued with the evaluation of the accuracy of the model and the derivation of performance-related conclusions and guidelines for ETM. Moreover, as already mentioned, ETM approaches will also be validated by means of simulations. The simulator developed in SmoothIT is based on the abstract simulation framework Protopeer. The simulation model for a Tribler peer-to-peer VoD overlay and the associated network model are presented in this deliverable as well, together with a reference topology proposed, representative scenarios for experiments targeted at, and a list of currently configurable parameters of these simulations. This reference topology is rich enough to study all important effects applicable to an ISP employing ETM. The first ETM approach that will be implemented in the simulator is a centralized SIS, which is contacted by peers in its own Autonomous System to receive underlay information about their neighbors. Future work will focus on relevant experimental results and the evaluation of this approach on this basis, as well as on extending the simulation model and experiments’ specification to other ETM approaches.

Another promising direction for future work is that of inter-domain collaboration for the purposes of ETM. To this end, we envision a communication between SISes deployed by different ISPs. The objective is to allow the exchange of additional information, internal to the SIS' functionality (other than the BGP information, which is already available), that could help in coordinating the ISPs' decisions and lead to a further optimized situation. Optimization in this case will most likely be expressed in terms of monetary gains for the ISPs, without excluding better network performance. Possible situations that could enable such interaction include information exchange between peering ISPs, between a multi-homing ISP and its higher tier ISPs etc. Of course, due to the fact that the complete information to be exchanged could include sensitive information that would affect the ISPs' business relationships, a certain level of filtering and abstraction is considered necessary. Furthermore, the information should not be application and/or swarm-aware in order for the solution to be realistic and scalable. Work on SIS-SIS collaboration will be included in future SmoothIT deliverables.

References

[A99] RFC 2676, G. Apostolopoulos, R. Guerin, S. Kamat, A. Orda, T. Przygienda, D. Williams, “QoS Routing Mechanisms and OSPF Extensions”, Experimental, August 1999.

[ABJ02] A. Asgari, S. Berghe, C. Jacet, P. Trimintzios, R. Egan, D. Goderis, L. Georgiadis, E. Mykoniati, P. Georgatsos, D. Griffin, “ A Monitoring and Measurement Architecture for Traffic Engineered IP Networks”, 2002

[AH00] E. Adar, B. A. Huberman, “Free Riding on Gnutella”, First Monday, Internet Journal, 5(10), October 2000.

[BB98] RFC-2475, S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", Informational, December 1998.

[BCC+06] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, A. Zhang, “Improving Traffic Locality in BitTorrent via Biased Neighbor Selection,” In Proceedings of the 26th IEEE international Conference on Distributed Computing Systems (July 04 - 07, 2006).

[BGP1] RFC 1771, “A Border Gateway Protocol (BGP-4)”

[BGP2] RFC 1772, “Application of the Border Gateway Protocol in the Internet”

[BGP3] RFC 1966, “BGP Route Reflection: An Alternative to Full-Mesh I-BGP”

[BGP4] RFC 1997, “BGP Communities Attribute”

[BGP5] RFC 2270, “Using a Dedicated AS for Sites Homed to a Single Provider”

[BGP6] RFC 2385, “Protection of BGP Sessions through the TCP MD5 Signature Option”

[BGP7] RFC 2439, “BGP Route Flap Damping”

[BGP8] RFC 2842, “Capabilities Advertisement with BGP-4”

[BGP9] RFC 2858, “Multiprotocol Extensions for BGP-4”

[BGP10] RFC 2918, “Route Refresh Capability for BGP-4”

[BGP11] RFC 3065, “AS Confederations for BGP”

[BKH+08] T. Bocek, W. Kun, F. V. Hecht, D. Hausheer, B. Stiller. "PSH: A private and shared history-based incentive mechanism”, 2nd International Conference on Autonomous Infrastructure, Management and Security Resilient Networks and Services (AIMS), Bremen, Germany, July 2008.

[BTPEC] BitTorrent Peer Exchange Conventions: http://wiki.theory.org/BitTorrentPeerExchangeConventions

[C03] B. Cohen, “Incentives Build Robustness in BitTorrent”, Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.

[CB08] D. R. Choffnes, F. E. Bustamante, "Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems”, *SIGCOMM Comput. Commun. Rev.* 38, 4 (Oct. 2008), 363-374

[CLM+08] A. Cuoto da Silva, E. Leonardi, M. Mellia, M. Meo, “A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems,” 8th International Conference on Peer-to-Peer Computing (P2P’08), Aachen, Germany, 2008

[CR05] M. Caesar, J. Rexford, “BGP routing Policies in ISP networks,” UC Berkeley Technical report, UCB/CSD-05-1377, March 2005.

[D02] J. R. Douceur, “The sybil attack”, in IPTPS ’01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, pages 251–260, London, UK, 2002. Springer-Verlag.

[D1.1] The SmoothIT consortium, Deliverable 1.1: “Requirements, Applications Classes and Traffic Characteristics (Initial Version)”

[D1.1A] The SmoothIT consortium, Deliverable 1.1 Annex: “SmoothIT Related Work”

[D2.1] The SmoothIT consortium, Deliverable 2.1: "Self-Organization Mechanisms for Economic Traffic Management"

[D3.1] The SmoothIT consortium, Deliverable 3.1: "Economic Traffic Management System Architecture Design"

[EHB+07] K. Eger, T. Hoßfeld, A. Binzenhöfer, G. Kunzmann, "Efficient Simulation of Large-Scale P2P Networks: Packet-level vs. Flow-level Simulations," 2nd Workshop on the Use of P2P, GRID and Agents for the Development of Content Networks (UPGRADE-CN'07) in conjunction with IEEE HPDC, Monterey Bay, USA, June 2007

[EuQoS] EU FP6 Integrated Project "EuQoS", 2004-2008, http://www.euqos.eu

[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 (November 12 - 15, 2006). ICNP. IEEE Computer Society, Washington, DC, 239-248

[FLS+04] M. Feldman, K. Lai, I. Stoica, J. Chuang, "Robust Incentive Techniques for Peer-to-Peer Networks", EC '04: Proceedings of the 5th ACM conference on Electronic commerce, pages 102–111, New York, NY, USA, 2004. ACM Press.

[G00] TEQUILA Deliverable D1.1, D. Goderis (ed.), "Functional Architecture and Top Level Design", September 2000, http://www.isttequila.org/.

[GB02] T. Giuli, M. Baker, "Narses: A Scalable Flow-Based Network Simulator", CoRR, 2002, http://arxiv.org/abs/cs.PF/0211024

[GFJ+03] Z. Ge, D.R. Figueiredo, J. Sharad, J. Kurose, D. Towsley, "Modeling peer-peer file sharing systems", INFOCOM 2003, 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, vol.3, no., pp. 2188-2198 vol.3, 30 March-3 April 2003

[GQX+04] D.K. Goldenberg, L. Qiuy, H. Xie, Y.R. Yang, Y. Zhang, "Optimizing cost and performance for multihoming", SIGCOMM Comput. Commun. Rev. 34, 4 (Aug. 2004), 79-92.

[GS05] P. Ganesan, M. Seshadri, "On Cooperative Content Distribution and the Price of Barter", International Conference on Distributed Computing Systems, 2005

[ITCSS18] 18th ITC Specialist Seminar on "Quality of Experience", Karlskrona, Sweden, May 29-30, 2008

[ITUY2001] ITU-T Rec. Y.2001, „General Overview of NGN", 2004

[Jur54] J.M. Juran, "Universals in management planning and controlling", Management Review, 43(11):748-761 (1954)

[KR06] R. Kumar, K.W. Ross, "Peer-Assisted File Distribution: The Minimum Distribution Time", *Hot Topics in Web Systems and Technologies, 2006. HOTWEB '06. 1st IEEE Workshop on* , vol., no., pp.1-11, 13-14 Nov. 2006

[KRP05] T. Karagiannis, P. Rodriguez, K. Papagiannaki, "Should Internet Service Providers Fear Peer-Assisted Content Distribution?", Internet Measurement Conference 2005: 63-76

[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

[LZG+05] Y. Liu, H. Zhang, W. Gong, D. Towsley, "On the Interaction Between Overlay and Underlay Routing", Proc. IEEE INFOCOM 2005

[MM] "MaxMind: Geolocation and Online Fraud Prevention", URL: http://www.maxmind.com/

[MPD+07] B. Mitchell, P. Paterson, M. Dodd, P. Reynolds, P. Waters, R. Nich, "Economic study on IP interworking", White paper, March 2007

[MPM+08] J.J.D. Mol, J.A. Pouwelse, M. Meulpolder, D.H.J. Epema, H.J. Sips, "Give-to-Get: An Algorithm for P2P Video-on-Demand", 2007

[MWW06] J. Mundinger, R. Weber, G. Weiss, "Analysis of peer-to-peer file dissemination", SIGMETRICS Perform. Eval. Rev. 34, 3 (Dec. 2006), 12-14.

[NCW05] S. J. Nielson, S. Crosby, D. S. Wallach, "A Taxonomy of Rational Attacks", 4th Annual International Workshop on Peer-To-Peer Systems (IPTPS 2005), Ithaca, NY, USA, February

2005. Springer Berlin / Heidelberg.

[P4P] P4P Framework, URL: http://www.openp4p.net/

[PBC06] F. Lo Piccolo, G. Bianchi, S. Cassella, "Efficient Simulation of Bandwidth Allocation Dynamics in P2P Networks", Proceedings of the Global Telecommunications Conference, 2006. GLOBECOM '06, San Francisco, CA, USA

[PEX] "Peer Exchange (PEX)", URL: http://www.azureuswiki.com/index.php/Peer_Exchange

[PM08] F. Picconi, L. Massoulié, "Is there a future for mesh-based live video streaming?", 8th International Conference on Peer-to-Peer Computing (P2P'08), Aachen, Germany, 2008

[PRO08] W. Galuba, J. Herzen, J. Respen, "ProtoPeer Peer-to-Peer Systems Prototyping Toolkit", URL : http://protopeer.epfl.ch, 2008

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

[RFC1633] RFC 1633, R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", 1994

[RFC2475] RFC 2475, S. Blake et al., "An Architecture for Differentiated Services", 1998.

[RFC2386] RFC 2386, E. Crawley et al., "A Framework for QoS-based Routing in the Internet", 1998

[RVC01] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", RFC-3031, January 2001.

[Shai99] A. Shaikh, J. Rexford, K.G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows", Proc. ACM SIGCOMM, Cambridge, MA, September, 1999

[SP03] J. Shneidman, D.C. Parkes, "Rationality and Self-Interest in Peer to Peer Networks", 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Berkeley, CA, USA, February 2003.

[SS06] S. Shakkotai, R. Srikant, "Economics of Network Pricing with Multiple ISPs", IEEE/ACM Transactions on Networking, Volume 14, Issue 6, pp. 1233-1245, Dec. 2006.

[TAP01a] P. Trimintzios, I. Andrikopoulos, G. Pavlou, C.F. Cavalcanti, D. Goderis, Y.T' Joens, P. Georgatsos, L. Georgiadis, D. Griffin, R. Egan, C. Jacquenet, C. Memenios, "An architectural Framework for Providing QoS in IP Differentiated Services Networks", IM2001, Seattle, WA USA, May 2001, http://www.isttequila. org/

[TAP01b] P. Trimintzios, I. Andrikopoulos, G. Pavlou, P. Flegkas, D. Griffin, P. Georgatsos, D. Goderis, Y.T'Joens, L. Georgiadis, C. Jacquenet, R. Egan,"A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks", IEEE Communications Magazine, vol. 39, No. 5, pp. 80-88, May 2001

[TEQ] "Traffic Engineering for Quality-of-Service in the Internet", Large Scale, IST project, URL: http://www.ist-tequila.org

[TK07] S. Tawari, L. Kleinrock, "Analytical model for BitTorrent-based live video streaming", Proceedings of IEEE NIME 2007 Workshop, Las Vegas, NV, January 2007

[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

[VY03] G. de Veciana, X. Yang, "Fairness, incentives and performance in peer-to-peer networks", In the Forty-first Annual Allerton Conference on Communication, Control and Computing, Monticello, IL,Oct. 2003.

[WCL08] J.H. Wang, D.M. Chiu, J.C.S. Lui, "A Game Theoretic Analysis of the Implications of Overlay Network Traffic on ISP Peering", Computer Networks, vol. 52, no. 15, pp. 2961-2974, 2008

[XK08] H. Xie, A. Krishnamurthy, A. Silberschatz, Y.R. Yang, "P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers", under submission, available online: http://www.dcia.info/documents/P4P_Overview.pdf

[XY08] H. Xie, Y.R. Yang, A. Krishnamurthy, Y. Liu, A. Silberschatz, “P4P: Provider Portal for Applications”, SIGCOMM Comput. Commun. Rev., ACM, 2008, 38, 351-362

[ZLG+04] H. Zhang, T. Liu, W. Gong, D. Towsley, “Understanding the Interaction Between Overlay Routing and Traffic Engineering”, 2004

Abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization, Accounting
AS	Autonomous System
BGP	Border Gateway Protocol
C/S	Client/Server
CoS	Class of Services
CAPEX	Capital Expenses
CP	Content Provider
DPI	Deep Packet Inspection
DRtM	Dynamic Route Management
DRsM	Dynamic Resource Management
EBGP	Exterior BGP
EGP	Exterior Gateway Protocol
ETM	Economic Traffic Management
GPS	Gradient Projection Search
HAP	Highly Active Peer
IAP	Internet Access Provider
IBGP	Interior BGP
IGP	Interior Gateway Protocol
IoP	ISP-owned Peer
ISP	Internet Service Provider
LSP	Label Switched Path
MBGP	Multiprotocol BGP
MED	Multi-Exit Descriptor
MPLS	Multi-protocol Label Switching
ND	Network Dimensioning
NEP	Nash Equilibrium Point
NGN	Next Generation Networks
NN	Network Neutrality
OP	Overlay Provider
OPEX	Operational Expenses
OR	Overlay Routing
P2P	Peer-to-peer

PHB	Per-hop Behavior
PoP	Point-of-Presence
PSH	Private and Shared History
QoE	Quality-of-Experience
QoS	Quality-of-Service
RFC	Request For Comments
RTT	Round Trip Time
SLA	Service Level Agreement
SIS	SmoothIT Information Service
SmoothIT	Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies
STREP	Specific Targeted Research Project
TE	Traffic Engineering
TFT	Tit-for-Tat
VoD	Video on Demand
VPN	Virtual Private Network

Acknowledgements

This deliverable was made possible due to the large and open help of the WP2 team of the SmoothIT team within this STREP. The authors would also like to thank especially the reviewers Burkhard Stiller, Zoran Despotovic and Nicolas Liebau for their valuable comments.