Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies

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

Deliverable D3.1
Economic Traffic Management Systems Architecture Design (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: Economic Traffic Management Systems Architecture Design

Type: Public

Editor(s): Fabio Hecht, Peter Racz

E-mail: hecht@ifi.uzh.ch

Author(s): Fabio Hecht, Peter Racz, Franziska Wirz, Martin Hochstrasser, Hasan, Burkhard Stiller, Zoran Despotovic, Wolfgang Kellerer, Maximilian Michel, Konstantin Pussep, Haris Neophytou, Sergey Kuleshov, Marcin Niemiec, Jan Derkacz, Jerzy Domzal, Rafal Stankiewicz, Krzysztof Wajda, Robert Wojcik, Zbyszek Dulinski, Maria Angeles Callejo Rodriguez, Juan Fernandez-Palacios, Osama Abboud, Michalis Makidis, Spiros Spirou

Doc ID: D3.1-v1.0.doc

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.1	December 7, 2007	Hasan, Burkhard Stiller	First version, providing template
V0.2	June 25, 2008	Fabio Hecht	Table of Contents
V0.3	June 30, 2008	Hasan, Peter Racz, Fabio Hecht, Burkhard Stiller	Table of Contents – Internal Revision
V0.4	July 7, 2008	Fabio Hecht	Gathered input from e-mails and phone conference
V0.5	July 9, 2007	Fabio Hecht, Peter Racz, Franziska Wirz	Minor corrections and new input
V0.6	August 18, 2008	Fabio Hecht, Michael Makidis, Spiros Spirou	Current contributions + UZH input
V0.8	August 28, 2008	Fabio Hecht, Peter Racz, Martin Hochstrasser, Konstantin Pussep, Osama Abboud, María Ángeles Callejo	New contributions
V0.9	September 1, 2008	Fabio Hecht, Marcin Niemiec	Inserted AGH contribution on Security section
V0.10	September 4, 2008	Peter Racz	Update of structure according to phone conference
V0.11	September 17, 2008	Peter Racz, Fabio Hecht, María Ángeles Callejo, Sergey Kuleshov, Michalis Makidis, Zoran Despotovic	New contributions and updates Internal review version
V1.0	October 12, 2008	Peter Racz, María Ángeles Callejo, Michalis Makidis, Marcin Niemiec, Sergey Kuleshov, Zoran Despotovic	Update according to review comments from Spiros Spirou, Tobias Hoßfeld, Simon Oechsner, Sergios Soursos, Nicolas Liebau, Hasan

Table of Contents

1 Executive Summary

The aim of this deliverable is to define the initial version of the SmoothIT architecture that will support Economic Traffic Management (ETM) mechanisms in order to manage and optimize overlay application traffic in the network of Internet Service Providers (ISP) and telecommunication operators. This approach will enable cost reduction for operators and better service quality for end-users.

This deliverable is the first result of Task 3.2 “System Architecture Design” that is developing the SmoothIT architecture supporting economic traffic management for overlay application traffic and it runs from September 2008 (M7) to August 2009 (M18). This deliverable also confirms the achievement of milestone M3.2 “System Architecture Definition (Draft)”.

The main objectives of this deliverable are to present the design space by discussing possible approaches for ETM of overlay traffic, to define main architectural components and their basic interactions, and to provide an initial version of the protocol between overlay applications and the SmoothIT architecture. Each of those steps may determine an important aspect for the Future Internet and its management, particularly respective control planes and protocols for Next Generation Networks (NGN).

Therefore, main results of this deliverable include:

- The overview and evaluation of related approaches focusing on the optimization of overlay traffic and on possible interactions between overlay application and network provider (Section 4).
- The description of a scenario for the SmoothIT architecture (Section 5) that underlines key requirements and the refinement of functional and non-functional requirements (Section 6).
- The description of possible ETM approaches between overlay applications and ISPs and the evaluation of these approaches highlighting their advantages and disadvantages (Section 7). The “Honey Pot” approach aims at attracting intra-domain peers by deploying a highly-preferred peer with significant amount of resources (*e.g.*, content and bandwidth) in the network of an ISP. The “Control Freak” approach introduces a new service in the network of an ISP in order to assist peer selection of overlay applications and to achieve a more efficient overlay traffic flow within and across ISP networks. The “Block Party” approach focuses on interactions between network providers to achieve a more informed decision in the overlay traffic management. Finally, the “Optimal Anarchy” approach describes a fully distributed solution deployed on the routers of an ISP. All four approaches have been evaluated and their advantages and disadvantages have been discussed.
- The specification of the initial SmoothIT architecture (Section 8). The initial architecture specifies main components and their functionality. It introduces the SmoothIT Information Service (SIS) that is a central element of the architecture. The SIS server (or server farm) communicates with overlay applications, assists the peer selection process of the overlay application according to operator policies (preferences) and application requirements, and is responsible for overlay traffic management. Additionally, the architecture includes a metering component that is responsible for collecting any information from the network relevant for ETM. The

QoS manager of the architecture controls and enforces QoS policies in the network. The architecture also considers security aspects and defines security-related components.

- The initial design of the protocol between the SIS server and overlay applications (Section 8). The protocol specifies a request-reply interaction between the SIS server and SIS clients in the overlay application and supports the retrieval of preference information from the ISP. The protocol also supports the flexible extension of request and reply messages with additional, new attributes.

Results contained in this deliverable will serve as the basis for future work in Task 3.2 “System Architecture Design”, as well as for the implementation of the SmoothIT architecture in Task 3.3 “Development of Implementation Architecture” and Task 3.4 “Implementation of Economic Traffic Management Mechanisms”. Additionally, this deliverable will also serve as input for future work in Work Package 2 “Theory and Modeling”.

2 Introduction

Managing the broadband-capable infrastructure of tomorrow's Internet requires suitable technology and mechanisms as well as valid and viable economic means. Overlay applications, such as BitTorrent [BT], PPLive [PPLive], and Skype [Skype], are very popular on the Internet today. According to published global Internet traffic measurements [CS08] by Cisco Systems, peer-to-peer file sharing applications created monthly Internet traffic of 1747 Peta byte in 2007, an increase of 26% compared to 2006. Peer-to-peer applications create a high share of total global Internet traffic (around 52% in 2007) and their traffic is increasing [CS08]. Video streaming traffic is steadily growing, making up a higher traffic share each year (around 20% in 2007) [CS08]. Video streams may be distributed either in a client-server fashion or in a peer-to-peer overlay network.

Network providers, such as Internet Service Providers (ISP), typically have to pay for the traffic that is crossing their network boundaries – that is, inter-domain traffic. Random overlay connections and resource selection leads to a high share of overlay traffic crossing the network provider's network domain boundaries, resulting in higher traffic cost. Increasing use of overlay applications by the network customers leads therefore to higher cost for the network provider. If overlay applications better adapt their overlay topology and their resource distribution to network locality, the amount of inter-domain traffic can be reduced and the cost for the network provider lowered. Details on the optimization potential of overlay applications can be found in [D1.1], while the self-organization mechanisms to adapt the overlay topology are described in [D2.1].

As the popular peer-to-peer overlays span globally, they share resources over long distances, creating traffic that is transported over long connections and multiple networks. Network traffic measurements in a large scale ISP [GHN+03] have shown that the majority of overlay applications do not show a locality-aware behavior that optimizes overlay traffic by choosing local over remote resources. This leads to a high share of inter-domain traffic.

Although overlay applications vary greatly in functionality and architecture, they share concepts that make it possible to develop solutions to optimize their performance in a generic way. One of these concepts, shared among many overlay applications, is the peer selection process. Each overlay application has its mechanism to find which peers offer a particular resource. To use file-sharing as an example, BitTorrent use a centralized or decentralized tracker to maintain a list of peers offering a certain file, while eMule uses index servers or Kademlia a Distributed Hash Table (DHT). After obtaining the list of peers offering a certain resource, a peer must select which peers to request the resource from. This is the peer selection process.

A simple approach to peer selection is random selection. Peers are selected at random and discarded if they do not fulfill the minimum expectation [C03]. Although this technique is robust and may approximate an optimal peer selection, it requires a long time to reach it. In order to reach optimality quickly, a peer may actively perform measurements on the network, as suggested in [ECR+03]. Those mechanisms are able to perform better than random selection, and may use a variety of metrics to decide which peers should be preferred. Examples of metrics are hop count, latency, Internet Protocol (IP) prefix, Autonomous System (AS) identifier, or available upload bandwidth of a peer to disseminate contents faster. Those methods, however, have shortcomings, since performing measurements are time-consuming and always accurate enough.

To address those problems for the Future Internet, peer selection can be influenced by the ISP, since it has more information about its underlay network than it can be measured directly by the overlay applications. Nodes of the overlay application can use the preference information provided by the ISP-owned information service when selecting the connections to other overlay nodes to pro-actively choose nodes that are closer in terms of network locality and less expensive in terms of cost to the network provider. As a side effect, the utilization of network locality may result in a better Quality-of-Experience (QoE) of the end-user, *e.g.*, due to shorter delays for video streaming.

The main objectives of SmoothIT include the design and prototypical implementation of a mechanism for economic traffic management (for the definition see Section3, additional details on ETM can be found in [D2.1] and [OSP+08]) that allows for the restructuring of overlay application topologies to reduce costly network traffic and gain quality benefits for application users in the Future Internet. The design of such a network management mechanism is aimed to be based on economic principles, focusing on stake holder incentives. To achieve economic traffic management of overlay application traffic, the SmoothIT Information Service (SIS) is proposed that is deployed in the network of an ISP and enables the interaction between overlay application and the underlying network. The SIS conveys information between the overlay and the network, providing, *e.g.*, locality information to the overlay and assisting the peer selection process. The envisioned impact of SmoothIT are cost savings for ISPs, lower prices for end users and better quality of service for the overlay applications.

2.1 Purpose of Document D3.1

Deliverable D3.1 defines a first architecture design for the SmoothIT Economic Traffic Management System. The purpose of this document is to discuss possible ETM approaches between overlay applications and ISPs and to evaluate these approaches highlighting their advantages and disadvantages. Additionally, its purpose is to present the initial SmoothIT architecture, to define main components and their functionality, and to specify an initial version of the protocol between overlay applications and the SmoothIT architecture. The document serves as the basis for future work in the architecture design and implementation.

2.2 Document Outline

The outline of this deliverable is the following. Section 3 provides the terminology. In Section 4 related work is presented and analyzed. Section 5 describes a scenario for the SmoothIT architecture. Based on this scenario and deliverable D1.1 [D1.1] Section 6 specifies key functional and non-functional requirements for the SmoothIT architecture. Section 7 presents the design space by discussing four main approaches of ETM for overlay traffic. The initial SmoothIT architecture is presented in Section 8, focusing on main architectural components and their interactions. Finally, Section 9 summarizes this deliverable and gives an outlook to future work. Additionally, Appendix A lists and discusses use cases of the SmoothIT architecture.

3 Terminology

This section covers the key terms as well as its explanations utilized for the SmoothIT architecture design.

Internet Service Provider (ISP)

ISPs are commercial providers that offer Internet connectivity to users and companies. Between different ISPs, agreements are in place to regulate how traffic is routed from one ISP to another and how it is charged. Peering and transit agreements are the most common settlements for ISPs.

Peering Agreement

Peering agreements are common between two ISPs that are approximately of the same size, have similar data volumes and are geographically located in the same region. This kind of agreement is a mutual understanding to forward the traffic from the partner ISP for free. No money is being exchanged in this kind of settlement.

Transit Agreement

Due to the different sizes of ISPs, the smaller ISPs rely on the services of larger ones to connect their customers to the Internet. Transit agreements are settled on financial basis and describe a customer relationship between a larger ISP and a smaller one. Prices are based, for example, on data volume or bandwidth consumption.

Intra-domain Traffic

Traffic that stays within the borders of one ISP domain is referred to as intra-domain traffic. It does not incur additional costs for the ISP.

Inter-domain Traffic

Inter-domain traffic is the counterpart to intra-domain traffic and describes traffic that leaves the domain of one ISP. Inter-domain traffic may incur additional costs to the ISP, depending on the kind of agreement that is in place with the neighboring ISP that receives the traffic.

Overlay Networks

An overlay network is a logical network built by applications to use alternative routing mechanisms independent from IP routing and achieve an abstraction from the network, *e.g.*, new services and features not deployed in the Internet like multicast can be emulated on application layer over virtual overlay networks. Overlay applications include, but are not restricted to, peer-to-peer networks.

Peer-to-Peer Network

A peer-to-peer network is an overlay network in which peers – normally home computers – both provide and consume resources. Examples of services offered by peer-to-peer networks are file sharing, Voice-over-IP (VoIP) and video streaming.

Economic Traffic Management (ETM)

SmoothIT proposes a new traffic management mechanism termed Economic Traffic Management (ETM), which provides for incentive-compatibility in interactions between overlay applications and the underlying ISP networks in order to gain the following measurable impacts:

1. Cost saving for ISPs: lower operation costs, due to ETM-based traffic engineering, lower interconnection costs, since traffic can be kept inside an ISP's domain, and lower capacity extension cost, since capacity requirements can be forecasted with much higher accuracy.
2. Lower prices for end-users, due to competitive pricing by the ISP, which are enabled by new ETM mechanisms.
3. Better Quality-of-Service (QoS) for overlay-based applications across ISP domains, due to the usage of ETM-based traffic engineering. This leads to an improved media consumption experience for end users.

Incentive

An incentive determines a monetary or non-monetary factor which provides a motivation for a particular course of action, or counts as a reason for preferring one choice to another.

SmoothIT Information Service (SIS)

SIS is a service that conveys information between overlay application and network. It is accessed by overlay applications and provided by a network operator in order to achieve ETM of overlay application traffic.

SIS Client

An SIS Client is a client of the SmoothIT Information Service that is located in the overlay application and accesses the SIS server

SIS Server

An SIS Server is a server or a server farm that provides the SmoothIT Information Service and replies to requests from SIS clients.

4 Related Work

Many solutions have been proposed to deal with P2P (Peer-to-Peer) traffic. Most of them focus primarily on locality, as it is a simple way to achieve a win-win-win situation for ISPs, end users, and overlay providers. This chapter presents approaches that involve cooperation between ISPs and overlay applications. In cooperation-based traffic management approaches, overlay applications and network providers exchange information to optimize the overlay topology to the underlying network.

Four existing cooperation-based overlay traffic management and optimization approaches are presented. Their functionality and limitations are briefly described.

4.1 Biased Neighbor Selection in BitTorrent

BitTorrent [C08] is a popular file sharing protocol used on the Internet. In the regular BitTorrent protocol, a peer contacts a tracker server to retrieve a list of random peers to download a certain resource from. For each resource shared, there exists a separate overlay of peers to exchange file chunks.

[BCC+06] suggest a change of the BitTorrent protocol to include a biased neighbor selection mechanism to increase traffic locality. The suggested changes could either be made in the BitTorrent tracker application, for a cooperative traffic management approach, or could be transparently enforced by an ISP into the protocol through tracker response manipulation on edge routers. Instead of measuring peer locality through connection delay measurements, accurate network domain affiliation information offered by the network provider is used in the biased neighbor selection mechanism.

The proposed cooperation-based approach for biased neighbor selection requires the adaptation of the tracker server application to return a list that contains a minimum amount of peers that are located in the same network domain as the requesting peer. The tracker server requires a method to group the peers which it keeps track of, according to the network domain they belong to. Therefore, network domain information is required to assign a network domain identification to each peer.

Two possibilities are presented that are based on cooperation between the network provider and BitTorrent to supply the tracker with the required network domain information. Network providers can either publish the IP (Internet Protocol) ranges of their network domains, to allow tracker servers to map requesting peers to network domains. Alternatively, network providers can use HTTP (Hyper-text Transfer Protocol) proxies that forward requests from BitTorrent peers within their network to trackers. The HTTP proxies add an additional HTTP header field to the request, containing a network domain identifier.

Evaluations of biased neighbor selection have shown to increase the performance of BitTorrent. Download times of files are lower with biased neighbor selection than with randomly selected neighbor peers. Inter-domain connections are systematically avoided, leading to an effective reduction of inter-domain connections and traffic cost for network providers.

4.1.1 Limitations

The biased neighbor selection method is not applicable to overlay applications other than BitTorrent, as it is dependent on the tracker interaction protocol.

Even though trackers can be adapted to use network domain address range information, the paper does not state in which way or format network providers publish the address range information. There is also no information given on how tracker servers can find the network domain address range information. Using a central entity to provide all network domain information from different network providers limits scalability of biased neighbor selection. If the information is directly served by the network provider, every tracker needs to have the address of the information server. As a tracker can receive requests from any peer in the Internet, it would require to have or to obtain all the addresses of the network-domain information servers of all network providers in the Internet, limiting the scalability of this approach.

The topology information derived from the network domain identifiers is binary. Two peers either belong to the same network domain or not. Connections between peers of different network domains can cause different traffic cost to network provider due to different traffic pricing agreements. Biased neighbor selection does not distinguish differing cost of inter-domain connections. An active optimization of inter-domain overlay connections is not possible.

4.2 Oracle Service

[AFS07] propose an oracle service that is offered by network providers that allows overlay nodes to query locality information about potential neighbors. The oracle service allows the network provider to express its preference to the querying node. This preference is taken into account in the neighbor selection process, to increase traffic locality and reduce cost for the network provider.

The oracle service ranks a set of potential neighbors according to the following proposed distance metrics:

- inside or outside AS (Autonomous System, as defined in RFC 1930 [HB96]);
- number of AS hops according to the BGP (Border Gateway Protocol) path;
- distance to the edge of the AS according to the IGP (Interior Gateway Protocol) metric.

For hosts within the providers' network domain, additional metrics are proposed:

- geographical information;
- performance information (*e.g.*, delay or bandwidth);
- link congestion.

The oracle service has been evaluated with the Gnutella protocol in a network simulator. Using the unchanged Gnutella protocol, a peer randomly connects to peers from a cached list to join the overlay network. For the evaluation the Gnutella peers were changed to first send the complete cached list to the oracle. The oracle picks a peer within the same AS for the joining peer to connect to. If no local peer is available, a random peer is chosen. When a peer searches for a file to download and find multiple sources, the different sources are also sent to the oracle for comparison to pick an intra-AS source, if available. For the simulation, random cached peer lists were generated with either 100 or 1000 entries.

The evaluation results show that the oracle increases the share of intra-AS peer-to-peer connections of the overlay structure. With Gnutella's original neighbor selection, an average of 14.54% overlay connections is intra-AS. Using the oracle for the neighbor selection process the share is increased to 38.04% for a queried list of 100 and to 74.95% for a list with 1000 peer entries. The results show that the effectiveness of the oracle service approach depends on the size of the peer list that is sent to the oracle. The longer the list, the higher is the probability to contain an intra-AS peer.

If the peers consult the oracle service to choose between multiple file download sources, the share of intra-AS file transfers is increased from 6.5% to 40.57%. The use of the oracle service for the neighbor selection process significantly increases the share of intra-AS traffic caused by Gnutella, lowering the traffic cost for the network providers.

4.2.1 Limitations

The oracle service only uses network topology distance metrics (inside/outside AS, AS hop count and IGP distance) to differentiate hosts outside the own AS. However, hosts with similar network topology distance can show significantly different network traffic cost due to different network peering and traffic pricing agreements. To allow for a better optimization, the oracle service also has to consider traffic cost information in the preference assignment process.

Even though the oracle service is designed to use detailed topology and network performance information to sort intra-AS request list entries, this information is not propagated to the querying overlay node. Overlay applications could use the network performance information to choose between multiple intra-AS nodes according to special application requirements, such as high available bandwidth or low delay.

The oracle service only sorts a list of supplied IP addresses. Overlay nodes can not compare the preference of sorted addresses from different replies, as the preference semantic is only valid within each single sorted list. To compare new potential neighbors with a set of already sorted neighbors, the total set of all addresses has to be resent to the server. A caching mechanism of preference information on overlay application nodes is not possible.

4.3 Network Topology Information Desk Service (NTIDS)

The Network Topology Information Desk Service (NTIDS) presented by [B03] allows a network provider and overlay application cooperation similar to the oracle service. The system uses an information desk service that is offered by the network provider, that can be queried by nodes of overlay applications within the network provider’s domain to acquire network topology information.

The NTIDS server provides a sorting service that allows overlay nodes to send a list of IP addresses to be sorted according to network provider preferences. The NTIDS server consults routing tables from a database to determine the proximity of the address within the request. The basic design of NTIDS describes only the use of routing information to sort the request entries according to network distance. The sorting process does not include traffic cost information.

The request and response messages contain header and payload bodies and are encoded in a binary format. The messages are transported over TCP (Transmission Control Protocol) connections for data transport reliability.

4.3.1 Limitations

NTIDS shares the limitations of the oracle service. No traffic cost information is used to sort the request entries. Therefore the cost optimization potential is limited due to the inability to compare the traffic cost for hosts with similar distance.

As the information desk service only sorts the list of requested address entries, the preference information semantic is only valid within a single request. It is not possible to compare the preference between responses list entries from different reply messages.

4.4 P4P

P4P [XKS+07] stands for Proactive Network Provider Participation for P2P. P4P uses an information service entity offered by the network provider to allow cooperation between network provider and overlay applications. The information entity is called iTracker. The P4P system focuses on traffic optimization of overlays within a network provider's domain.

The iTracker offers three interfaces. The info interface provides topology information. It assigns AS identification, a locality cluster ID and geographic locality information for IP addresses in a query. The policy interface provides cluster-to-cluster traffic preference information based on traffic cost and network policy information. The capability interface allows overlays applications to request participation of dedicated peers offered by the network provider, to improve application performance for overlay applications.

An overlay node can acquire cluster IDs for a set of intra-domain neighbors, through the info interface of iTracker. Together with the cluster-to-cluster preference information provided by the policy interface, the overlay node can then establish connections to intra-domain neighbors that optimize traffic within the network-domain. For an overlay network that uses tracker servers to provide requesting nodes with a list of neighbors to connect to (such as BitTorrent), the process of optimal neighbor selection through querying the iTracker interfaces can be performed by the tracker server, to adapt the list replied to the peer according to network provider preference.

The iTracker keeps track of overlay nodes within the different intra-domain locality clusters to optimize cluster-to-cluster traffic by changing the preference information supplied to the overlay applications. Overlay nodes indicate their overlay affiliation to the iTracker (for example the BitTorrent swarm hash). As the iTracker can map the overlay node IP addresses to clusters, it can maintain an abstract topology map for each overlay and the nodes located in each cluster. The iTracker periodically calculates the best cluster peering preferences according to the topology maps of the different overlays, and takes into account current network traffic conditions and traffic cost information.

Evaluation of P4P with a tracker based file sharing application on a simulated ISP network have shown that download completion time is reduced by 45% and intra-domain link utilization is improved up to 70%. P4P optimizes overlay traffic to network provider preferences and improves application performance.

4.4.1 Limitations

P4P has a higher complexity than the other presented cooperation-based approaches, because of the tracking of overlay node states in cluster topologies. To optimize multiple overlays, cluster topologies have to be maintained for each one. Therefore P4P requires higher amounts of hardware resources to handle a large number of overlays, than simpler cooperative traffic management approaches that do not require maintaining overlay state information. Considering that tracker based file sharing applications such as BitTorrent create a separate swarm overlay for each shared resource, P4P has to track the state of potentially thousands of separate overlays, to manage all BitTorrent traffic.

Even though P4P optimizes overlay traffic within a network domain, there are limits to the usability for optimization on a larger scale. To be able to optimize traffic in a multi network domain environment with cooperation of multiple network providers, the different iTrackers would have to cooperatively maintain large scale cluster topology maps for all overlays that should be optimized. Scaling P4P to work over a large number of different network provider domains is challenging due to its complexity. This limits P4P's usability for optimization of traffic according to different inter-domain preferences.

4.5 Comparison

Table 4-1 shows a comparative overview of the cooperation-based traffic management and optimization approaches for overlay applications presented in this chapter. All allow for overlay application performance benefits and cost reduction for network providers due to reduced inter-domain traffic and increased traffic locality.

Biased neighbor selection, the oracle service and NTIDS only reduce network traffic cost for network providers through the higher traffic locality in the overlays. Systematic cost reduction according to traffic pricing information is not possible, as these three approaches do not consider such information.

Table 4-1: Comparison of Cooperation-based Overlay Traffic Management and Optimization Approaches

	Biased Neighbor Selection	**Oracle Service**	**NTDIS**	**P4P**
Information processed on network provider side	None	Topology, QoS metrics	Topology (routing tables)	Topology, cost, traffic policy, QoS metrics
Information provided to overlay applications	Domain Range	Sorted list	Sorted list	Topology (AS ID, cluster ID, locality), preference
Inter-domain connection share reduction	Yes	Yes	Yes	Yes
Inter-domain connection differentiation	No	Distance, QoS metrics	Distance	Distance, cost, traffic policy
Intra-domain connection differentiation	No	Distance, QoS metrics	Distance	No

These four approaches show differences in the differentiation possibilities of intra-domain and inter-domain connections. Biased Neighbor Selection does not allow for the comparison of different inter-domain or intra-domain connections. NTIDS can only differentiate connections based on network topology distance, as it only considers routing information. The oracle service has improved connection differentiation capabilities as it also allows comparing intra-domain connections according to QoS metrics, if QoS differences are considered during the list sorting process. P4P has the best differentiation capabilities for inter-domain connections of the four approaches presented in this section. Besides static topology information, network capabilities and cost information, it also considers real time network performance and overlay node distribution information. The tracking of the overlay nodes makes it the most complex approach of the four described.

The SmoothIT project has similarities with the approaches discussed and will take some of their ideas, like the interaction between overlay applications and the underlying network in the form of a generic information service, the provisioning of locality information, and the reduction of inter-domain traffic. But the SmoothIT project will further investigate incentive-compatibility to achieve a win-win-win situation for ISPs, users, and overlay providers. It considers not only locality information but additional approaches as well, like applying differentiated pricing, or providing QoS-enabled services. Additionally, SmoothIT aims at developing a solution applicable for different overlay application types, like file sharing and video streaming, and will therefore consider different application characteristics. Finally, SmoothIT also aims at investigating inter-domain interactions in more detail.

5 Scenario

This section presents a scenario showing possible interactions between overlay applications and ISPs. This scenario (see Figure 5.1) is used to illustrate benefits from a cooperation for users and ISPs and to derive requirements and possible design approaches for the SmoothIT architecture.

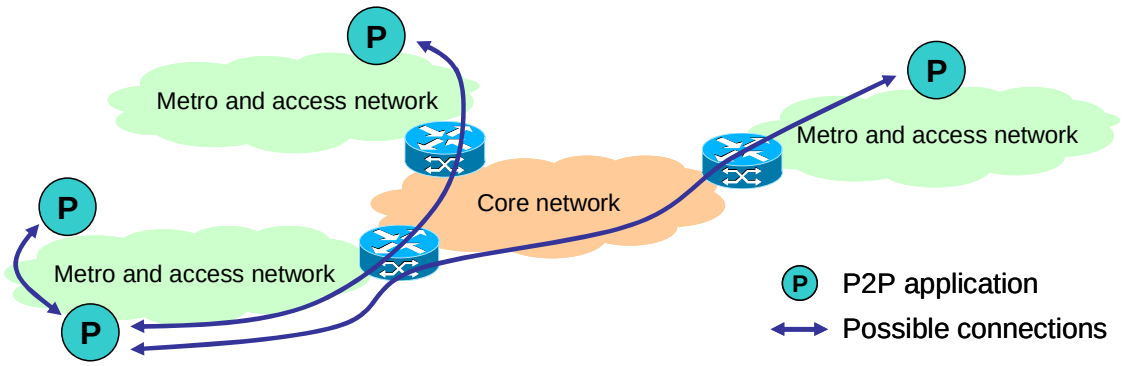

Figure 5.1 – Scenario

Consider a P2P video streaming application. For simplicity, assume that all peers belong to a single ISP. Any peer is interested in receiving the stream at the highest possible quality. To achieve this goal, the peer tries to retrieve the stream from peers with the following connection properties: low delay, high throughput, and long time-to-fail. The peer has two possibilities: one is to keep track of the behavior of other peers and make appropriate decisions based on own measurements. The other possibility is to ask for the help of its ISP, which can do the needed measurements more accurately (and, in fact, does most of them already as part of its daily work).

Consider now the other player in this scenario, i.e., the ISP. Among all possible strategies to connect peers, there might be some that, apart from providing the best quality of service to the peers, provide benefits to the ISP as well. Such strategies might bring uniform distribution of the load in the ISP's network, lower load in the ISP's backbone and so on.

This simple scenario shows that it makes sense for the ISP to provide a service to overlay applications to guide the overlay formation. The key is to see that both the ISP and the overlay application (i.e., the end users) can benefit from such a service (see [D1.1] for further details on incentives). In the rest of this document it is assumed that this service is provided through an architecture component (typically a server that might be distributed) that is called SIS.

In order to obtain relevant information from the SIS, overlay needs to discover SIS components or they have to be manually configured. This function can be either internal to the overlay or an external service, *e.g.,* Domain Name System (DNS) extension or a public web service.

There are a number of problems with a straightforward application of the approach. Most notably, since the ISP is effectively forming the overlay can peers trust the ISP? Will the ISP act for increasing the peers' quality of service or will simply optimize its cost reduction regardless of peers' utilities? Algorithms are needed to motivate the ISP to make choices that benefit both the peers and the ISP. Or, it must be possible for the peers to form the overlay without the ISP's intervention and select the one which brings them higher benefits.

6 Requirements

The SmoothIT project has defined both functional and non-functional requirements. They were defined in Deliverable D1.1 [D1.1] and further refined during the process of designing and detailing the system architecture.

6.1 Functional Requirements

Table 6-1 shows the main functional requirements identified for the SmoothIT architecture.

Table 6-1: Functional Requirements

ID	Requirement	Core req.	Add. req.
R.1.	Improving P2P application performance while reducing the network traffic: SmoothIT will create a win-win-win situation for the involved players: end-users, ISPs and, possibly, overlay providers. As an example, end-users may benefit through selecting local peers to connect to, while ISPs can benefit by network status gathering and subsequent traffic optimization and shaping. SmoothIT will come up with overlay operation strategies that improve on the current practice by employing these two techniques simultaneously.	X	
R.2.	Incentive-compatibility: The solutions provided by SmoothIT will be incentive compatible in the sense that it will be in the best interest of all involved players to behave according to the rules of the proposed SmoothIT protocols.	X	
R.3.	Support of different overlay applications: The SIS shall provide an open service that is accessible by different P2P applications.	X	
R.4.	Interface supporting various optimization schemes: The interface between the SIS and the overlay application shall provide means to specify the application scenario and the respective parameters. Due to the various incentives of ISPs, overlay providers, and end-users, the SIS shall provide several services (*e.g.*, “Throughput Optimization”, “QoS enhancement”) that could be classified into *free* and *premium* (charged) network services.	X	
R.5.	QoS support: The SIS shall support QoS for network services and it shall be able to configure network resources.	X	
R.6.	Different mode of operation: The SIS shall be able to operate in two different modes: user anonymity mode for free services and user aware mode for premium services.		X

R.7.	Inter-domain support: The SIS deployed in different ISPs shall be able to interact with each other. SIS elements in different ASs may communicate with each other in order to get the overall view of a communication in respect of the optimization parameters specified.		X
R.8.	OAM (Operation and Management) support: The SIS shall be able to interact with the OAM processes of the ISP.		X
R.9.	Mobile network support: The above requirements should also be valid in the context of a cellular network operator, which is characterized by the following key properties: node mobility, heterogeneity of nodes and link capacities, and presence of shared medium.		X

6.2 Non-functional Requirements

Table 6-2 shows the non-functional requirements identified for the SmoothIT architecture.

Table 6-2: Non-Functional Requirements

ID	Requirement	Core req.	Add. req.
R.10.	Easy deployment: It shall be easy to extend existing overlay applications with the functionality of the SIS and it shall be easy for ISPs to deploy the SIS in their networks.	X	
R.11.	Extensibility: The SIS shall be extendible to support new overlay applications, new optimization attributes, and new metrics (both application-driven and provider-driven).	X	
R.12.	Scalability: The SIS shall be scalable to support a large end-user population.	X	
R.13.	Efficiency: The operation of SIS shall be efficient in terms of communication (bandwidth) overhead, storage consumption, and processing requirement.	X	
R.14.	Robustness: The SIS shall be robust against malicious behavior and against dynamic behavior (churn of peers). It shall be also fault tolerant.	X	
R.15.	Security: Secure communication between SIS entities and between SIS and overlay application shall be supported, providing message origin authentication, data integrity, and data confidentiality. Any data storage in the system shall provide data integrity, confidentiality, and authentication.	X	
R.16.	Standard compliance: The SIS shall use and based on standard protocols where applicable.	X	

R.17.	Transparency: The SIS shall not apply Deep Packet Inspection (DPI).	X	
R.18.	Data privacy and legislation/regulation: The SmoothIT architecture needs to provide interfaces for regulation aspects, such as data retention, and it has to address data privacy concerns, which are determined by the European Directives on Security.		X

7 Design Space

In this section, the range of architectural options for SmoothIT in general and the SIS in particular is explored. This section does not recommend a specific architectural option; it lists and evaluates possible options only. Also, the elements of the options presented here can and should be freely combined to create a more concrete architecture fulfilling as many requirements as possible.

A central component of all architecture approaches is the ETM module, which contains the ETM algorithms. This module usually takes as an input some economic "signals" from the end user (like the requested content or type of QoS) and outputs the type of service that the user will receive (this might include parameters like QoS, pricing and others). When the component with the ETM algorithms resides in a central server, it is referred to as SIS (SmoothIT Information Service).

7.1 The Honey Pot: Attract Peers

This architectural option follows the main idea of attracting peers' resource requirements to be fulfilled by a super peer.

7.1.1 Overview

In this example a possible architecture of SmoothIT (Figure 7.1) is described for attracting normal P2P peers to an ISP-managed over-provisioned P2P peer so that inter-domain traffic is reduced. The concept borrows from insect management with an exposed pot full of honey. Emphasis is on reducing inter-domain traffic without resorting to DPI. The performance of the end user's P2P application is also increased, as it would get the content from a local peer [AAF08] that has low delay and high bandwidth.

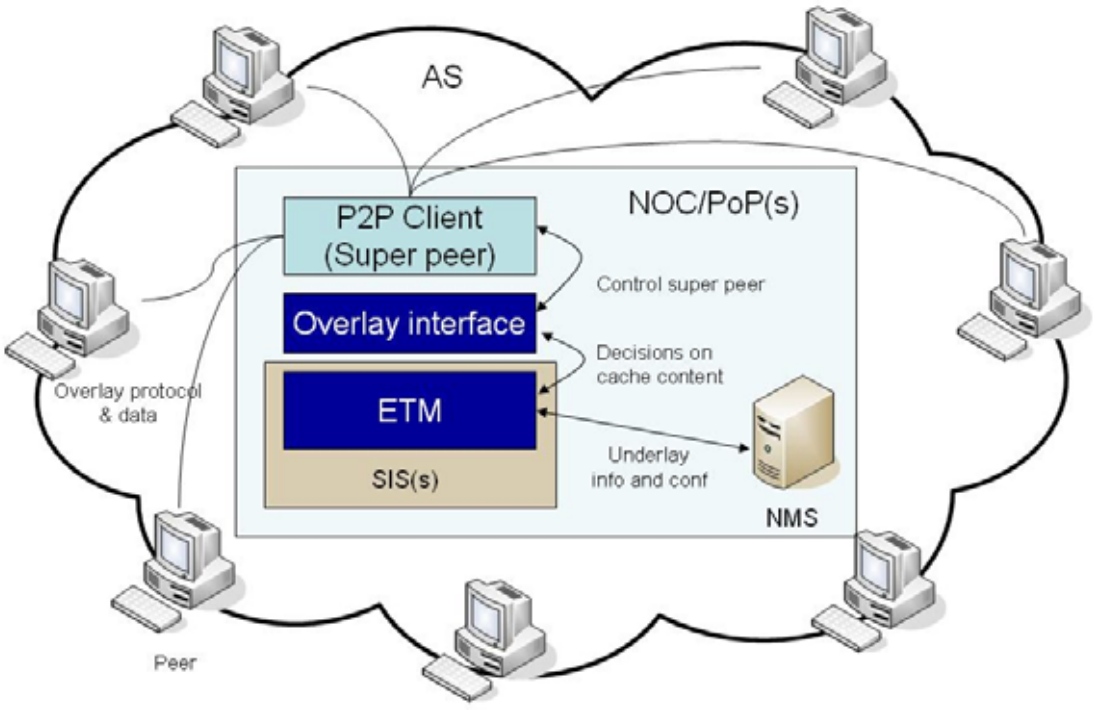

Figure 7.1 – The "Honey Pot" Concept

In this option, the ISP could add a new element to its network, the SIS. Here, the SIS functions as an intelligent super-peer, i.e. it participates in the overlay of an application and provides a lot of resources (such as bandwidth). This makes it more attractive to be selected as a peer by intra-domain overlay peers. It is assumed that the overlay already has a mechanism to discover potential peers with a lot of resources and that it will prefer to use them.

7.1.2 Structure

The possible structure encompasses components and interfaces to be described.

7.1.2.1 Components

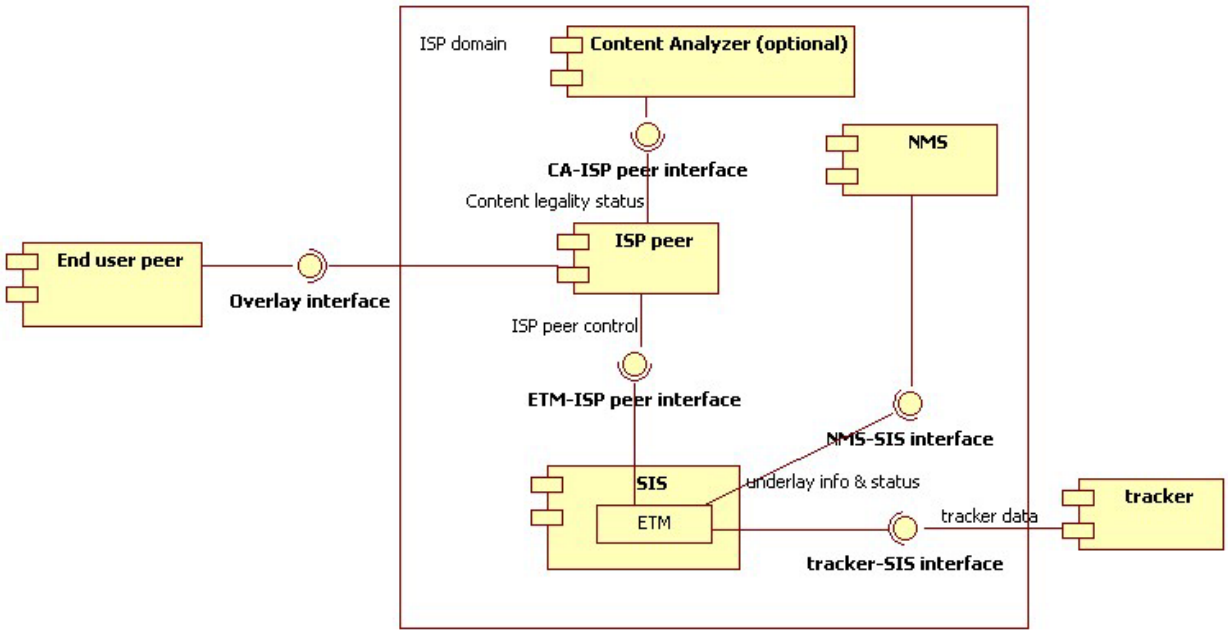

Figure 7.2 – The “Honey Pot” Component Diagram[1]

The components (Figure 7.2) of this architecture are:

End user peer: The P2P client software of the end user is used to connect the user with the overlay. This software is unmodified.

SIS: This component encapsulates the SmoothIT ETM algorithms at the ISP’s premises as well as the interfaces to the other components.

ISP peer: The ISP P2P client software functions as a normal peer that downloads and shares P2P content. A different client software is required for each overlay that the SIS needs to participate in. This client caches the content that the SIS instructs it to do so.

ETM: Used in order to decide on the content that the ISP client must cache. It gathers data from the NMS (Network Management System) in order to discover information concerning the network status (such as intra-domain and inter-domain links nearing congestion or expensive inter-domain links). It also makes decisions as which content

[1] In this section, UML 1.x notation is used.

to cache and which peers it should be presented to as an attractive super-peer. It also receives any policy set from the NMS. Furthermore, the ETM component gathers data from trackers or other overlay elements in order to make reasonable decisions regarding caching. This is similar to how iTrackers gather overlay information in the P4P project [XYK+08].

Content analyzer (optional): Used in order to analyze if the cached content can be redistributed. The aim of this component is to check if any cached content has no copyright restrictions or the content copyright allows ISP distribution. This component can be deployed physically on the same hardware as other SIS subcomponents or a separate machine. The content analyzer component can be treated as optional in the first stage of the implementation.

NMS: The Network Management System provides information such as the current inter-domain and intra-domain links of the ISP, the link usage (congestion) or the link cost (rate).

Tracker: The tracker (or other special-purpose element of an overlay) provides aggregate data regarding the current status of the overlay and the popularity and freshness of content.

7.1.2.2 Interfaces

The communication interfaces of this architecture are the following:

ISP peer-End user peer: The ISP peer software uses its existing communication interfaces and protocols in order to connect to other peers and participate in the P2P overlay. The peers use their existing P2P software with no modification. They discover, communicate with and treat the ISP peer as another peer, using the existing overlay protocol.

SIS-ISP peer: This interface is used by the ETM to interact with the clients (and influence the overlay by downloading and caching content). The ETM also uses this interface in order to gather information for the overlay via the ISP peer.

Content analyzer - ISP peer: This interface is used by ISP peer to access services which allow inspecting the copyright of the cached content.

NMS-SIS: This interface is used by the ETM in order to connect to a specific NMS and retrieve network status, underlay information etc. This interface is NMS-specific.

Tracker-SIS: This interface is used by the ETM component of the SIS in order to collect additional data (such as popularity of content etc.) from an overlay. This interface is specific to the type of tracker (*e.g.,* BitTorrent tracker). The tracker may need to be modified to support this.

7.1.3 Behavior

This architectural option's behavior is determined by an operation and message definition.

7.1.3.1 Operation

Based on components and interfaces as described above, the SmoothIT architecture in this scenario operates as follows: The SIS participates in overlay networks via the ISP peer, gathering data from peers and other overlay elements such as trackers and monitors

the requests for content coming from its domain, by means of the ISP peer(s) interface(s). The ETM component is able to make decisions on the content that needs to be cached locally so as to limit the usage of expensive inter-domain links. It makes choices such as which content to download to the cache and which content to remove from the cache. If the Content analyzer is implemented, copyright-protected content is filtered and not downloaded into the ISP's cache. The ISP peer uses its extensive resources (bandwidth, disk space, etc.) to pose as an attractive peer to the intra-domain overlay. The peers of the AS choose to get their content from the ISP peer, and as a result they get better performance. The SIS can also monitor trackers or other elements of the overlay and get aggregate data regarding the overlay. In this way, it can act proactively and cache content before it is requested by intra-domain users.

The SIS and ISP peers are not necessarily a single machine, nor are they essentially a single physical element of the network. In order to provide scalable service with high availability, the SIS and/or ISP peer could be a cluster positioned in the Network Operation Center (NOC) or a distributed system among the Point of Presences (PoP) of the ISP.

7.1.3.2 Messages

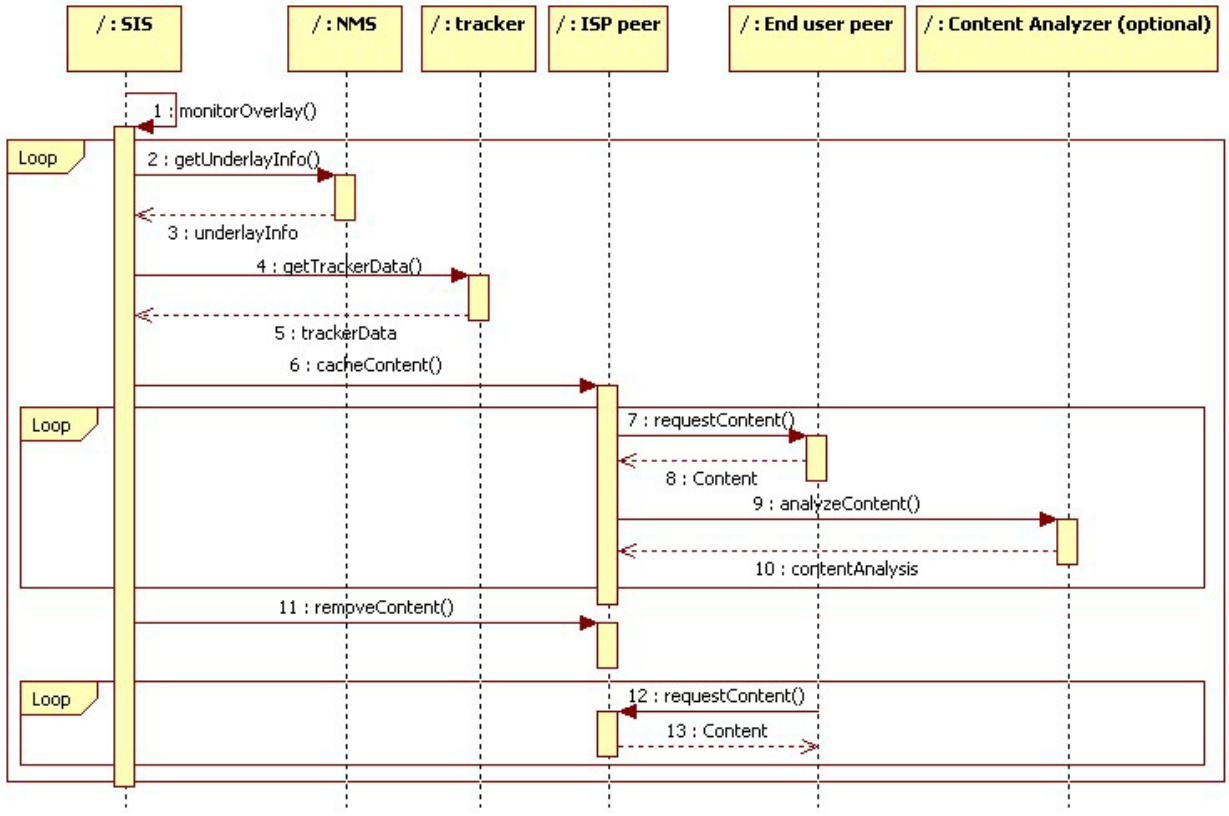

Figure 7.3 – The "Honey Pot" Sequence Diagram

The sequence (Figure 7.3) that the SIS performs in order to offer the above services is the following

1. The SIS is activated, it initializes in order to start monitoring the overlay. The overlay monitoring and content provision procedures are performed concurrently.
2. The SIS requests the current condition of the underlay from the NMS.
3. The NMS returns the current condition of the underlay, i.e. congestion on inter-domain and intra-domain links, link rates etc.
4. The SIS requests overlay information from trackers.
5. The trackers return information such as popularity of content to the SIS.
6. The SIS computes the content it needs to cache using ETM mechanisms and informs the ISP client.
7. The ISP client requests the content from a peer (End user client) which possesses it. This client may or may not be in the ISP domain.
8. The peer sends the requested content via the overlay's standard downloading procedure.
9. The ISP client asks the Content Analyzer whether the content is copyright-protected.
10. The Content Analyzer returns the copyright status of the content.
11. The SIS computes the content it needs to delete from the cache. The computation is based on the copyright analysis and ETM indications. Note that this is not a trivial problem. For example, previous attempts at P2P cache management can be found at [WLR+04].
12. The SIS client receives requests for content from some P2P clients (end user clients). This request is asynchronous and can appear in any sequence of the main loop.
13. The SIS client returns the cached content to the end user peer.

7.1.4 Potential

This option's potential is briefly discussed by addressing advantages and drawbacks.

7.1.4.1 Advantages

This topology has the advantage that it is easily deployable (R.10) and can be easily extended (R.11) by adding different ISP clients. There is no need to change the peers' software in the P2P overlay, the software of the routers or other network elements. Only a new set of self-contained elements need to be inserted, the SIS, ISP client (and optionally, the Content Analyzer). Moreover, the topology is not intrusive. Peers might not be interested in cooperating with the SIS and they may connect to any other peer they wish without consequence. This architecture also offers incentive-compatibility (R.2) for all parties involved, it supports different overlay applications (R.3), it performs network traffic optimization (R.1) and supports OAM integration (R.8). It also does not use DPI (R.17).

7.1.4.2 Disadvantages

Open issues for this topology are that it is difficult to bootstrap the SIS, as in some cases (for example, BitTorrent) you must be an active part of the P2P overlay in order to gather

data for this overlay. Therefore, it needs a new interface to collect information about a P2P overlay without joining it and it needs to provide incentive to trackers to support this interface. The ISP must also be very careful with the content it caches, so as to not cache copyrighted content. This would potentially limit the applicability of this topology to legal P2P content. Moreover, there are no well known automated solutions for filtering copyright-protected content that the ISP could deploy. The ETM methods for deciding which content to cache and when to remove content from the cache would also merit investigation. Finally, network monitoring by the NMS may be either real-time or offline, which requires a different approach by the SIS.

7.2 The Control Freak: Reward/Punish Peers

This architectural option follows the main idea of rewarding or punishing peers upon their behavior within the given overlay network.

7.2.1 Overview

Here an option for the SmoothIT system is described that monitors the ISP's network for P2P traffic, provides incentives to the peers and might optionally manipulate their network link to enforce them (Figure 7.4). The ISP sets the target behavior for the peers and fully controls reward and punishment based on their actual behavior. The peers inform the ISP about their price/performance ratio preference (*e.g.,* in the form of a slider) for each piece of content they need to consume. The SmoothIT architecture can also monitor the peers' traffic and adjust it (by rewarding or punishing some flows) so that it provides an incentive to the users to follow the suggested guidelines. By doing so, it is attempted to reduce the inter-domain traffic for the ISP and increase the performance of the end user application.

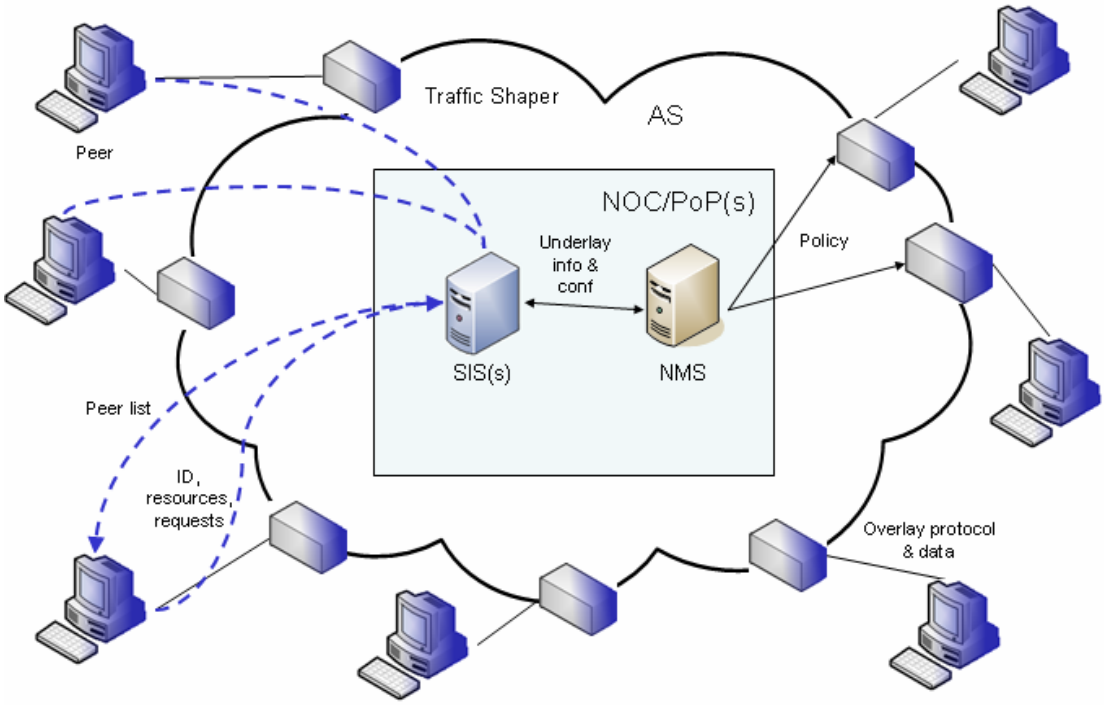

Figure 7.4 – The "Control Freak" Concept

The ISP has again added the SIS as a new element of its network. In this scenario, the SIS functions as an ETM-enhanced oracle [AFS07] that gathers the status of the ISP

network from the NMS (Network Management System). The difference here is that the SIS also ranks peers, computes suggested peers and calculates policies for incentives to the intra-domain peers. This can be combined with previous attempts at enhancing the locality of P2P applications (for example, for BitTorrent see [BCC+06]).

This scenario may use pricing in its ETM only for premium services, while standard services only get different performance.

7.2.2 Structure

The possible structure encompasses components and interfaces to be described.

7.2.2.1 Components

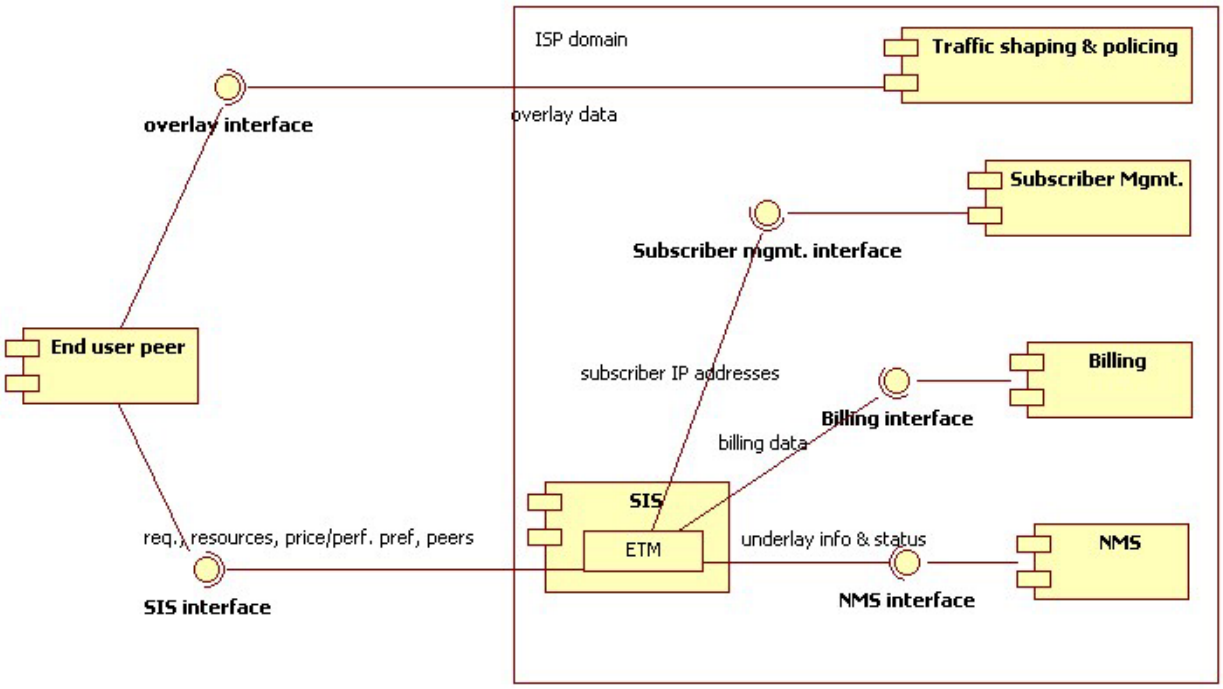

Figure 7.5 – The “Control Freak” Component Diagram

The components (Figure 7.5) of this SmoothIT architecture are:

End user peer: This software connects the end user to the overlay. It also sends the end user’s requests, resources and cost/performance ratio to the SIS and receives the suggested peer list. This list might contain a ranking of the peers provided by the End user peer and/or suggestions from the ISP.

SIS: This component contains the SmoothIT components at the ISP’s premises. It encapsulates the ETM mechanisms.

ETM: This component is part of the SIS and it receives the resources, requirements and pricing/performance preference from end user peers and underlay configuration and costs from the NMS and computes the optimal overlay configuration so as to minimize the ISP cost (by suggesting intra-domain peers) and increasing the peer performance. This component also computes billing information based on the

performance provided to users (probably in case of premium services only). It also sends suggested peers to the end user. It also uses the subscriber management component in order to identify peers, so that it can also employ reputation-based techniques (for premium services only).

NMS: The Network Management System provides information such as the current inter-domain links of the ISP, the link usage (congestion), the link cost (rate) etc.

Traffic Shaping & Policing: This is installed on each link to a local peer host. It monitors the P2P flows and applies the policy (for incentives) that has been calculated by the ETM.

Billing: This component is used in order to charge the end users based on the performance they receive.

Subscriber management: This component is used in order to map the user's IP addresses to user IDs (Identifier) so as to perform accounting based on the flows of each user (monitored by the traffic shaping and policing equipment). It can also be used to identify users, as the ETM algorithms might use reputation-based mechanisms.

7.2.2.2 Interfaces

The following interfaces are used in this SmoothIT architecture:

End user peer-overlay: The existing interface between the overlay and the P2P peers is maintained in order to participate in the P2P overlay.

End user peer-SIS: This interface is used by the end user peer application in order to announce its requests, resources and cost/performance preference to the SIS. The same interface is used by the SIS to return a suggested peers list.

ETM-NMS: This interface is the interface used by the ETM in order to connect to the NMS. This interface is NMS-specific. It is used in order to get the status of the network as well as communicate the incentive policy to the NMS

ETM-Billing: The billing interface is used in order to communicate the billing information (charge) of the end user to the billing system.

ETM-Subscriber Mgmt: The subscriber mgmt. interface is used in order to get the current IP addresses of end users.

7.2.3 Behavior

This architectural option's behavior is determined by an operation and message definition.

7.2.3.1 Operation

Based on the components and interfaces described above, the SmoothIT architecture in this scenario operates as follows. The SIS communicates with the NMS in order to get the current status of the network, such as the links that are nearing congestion and the flows (types and endpoints) that consume most of the resources. Using this data and the ETM mechanisms, the SIS computes a set of policies based on the incentives that it wants to present to the peers. It is capable of performing cross-request optimizations based on the network status and all the requests it receives. It communicates these policies to NMS and

the NMS applies them to the traffic shaping equipment. These policies would optimize the network resources usage and provide enhanced performance to the users and reduced costs for the ISP. The ISP can also manipulate the users' connection characteristics (by using traffic shaping and policing devices) in order to align their usage with the policies that have been computed by the SIS. The NMS presumably already has an interface to traffic shaping equipment for configuration options and monitoring of links. The actual changes that would happen on the connection characteristics (bandwidth, delay, jitter, etc.) of each flow depend entirely on the output of the ETM mechanisms. These mechanisms would also influence the charging of each user.

In order to present these incentives to the user, the SIS would also suggest a peer list to each peer of the overlay application in order to get maximum performance with a reduced cost for the ISP. Alternatively, the end user peer might send the list of peers it has selected, and have the SIS rank them according to the ISP's point of view [AFS07]. The end user peer software is modified in order to provide the SIS with information that is required for its ETM-based suggestions. Each peer communicates economic parameters that can be used by the ETM mechanisms. Such parameters can be its resources, ID, cost/performance preference and content requests to the SIS. The SIS collects this data from the entire AS and it can therefore have a clear view of the most requested and bandwidth-intensive content on the AS. As a result, it can calculate appropriate pricing, incentives and peer lists.

Again, the SIS is not necessarily a single machine, but could as well be a cluster positioned in the NOC or each PoP of the ISP.

7.2.3.2 Messages

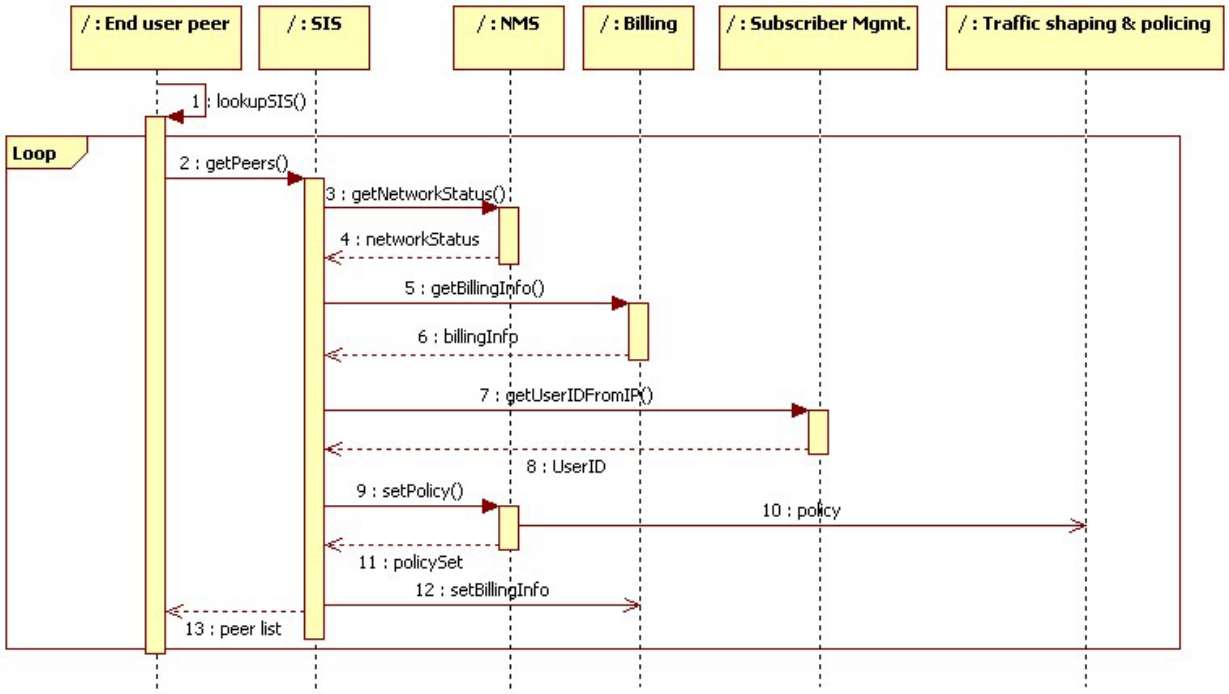

Figure 7.6 – The "Control Freak" Sequence Diagram

The sequence (Figure 7.6) for a peer to connect to a P2P overlay in a SmoothIT-enhanced ISP is the following:

1. The end user peer looks up the SIS IP address. This can either be a standardized, hard-coded DNS name that corresponds to a different IP address on each ISP or a manual setting. In the latter case, the user would enter the SIS name or IP address of her ISP in the same way that she would manually enter the DNS server IP for the ISP. This happens only the first time the peer tries to connect to the SIS.
2. The end user peer contacts the ISP requesting a suggested peer list. It sends the user requests, resources and price/performance ratio preference set by the user.
3. The SIS requests the current network status from the NMS.
4. The SIS gets the current status of the ISP network from the NMS. This includes types of links, congestion levels, rates for links and other economic data.
5. The SIS requests the current billing data for the end user from the Billing system to use as input for the ETM algorithms.
6. The Billing system returns the billing data to the SIS.
7. The SIS requests the user's ID from the IP address from the Subscriber Management System.
8. The SIS gets the user ID of the user from the Subscriber Management System. It uses this to set the billing information for the user.
9. The SIS computes and sets the traffic policy (including throughput, delay etc.) for this user and informs the NMS.
10. The NMS applies this policy to the traffic shaping equipment. This equipment then monitors and shapes each flow coming from the user.
11. The NMS notifies the SIS that the policy has been applied.
12. The SIS computes and sets the billing data for this user and informs the Billing system.
13. Finally, the SIS returns the suggested peer list to the end-user peer.

7.2.4 Potential

This option's potential is briefly discussed by addressing advantages and drawbacks.

7.2.4.1 Advantages

The advantages of this topology is that it can provide a wide range of incentives to the overlay peers, including traffic characteristics and pricing (R.2) without restricting itself to a specific application (R.3). Using the traffic shaping equipment the ISP can monitor the connections and have a detailed view of the traffic flows (R.1) and offer various optimization schemes to the client (R.4, R.6), based on his/her price-performance preference. This does not require DPI (R.17) if the traffic shaper only monitors the flow (i.e. the endpoints and port). It can also easily influence it by increasing or decreasing the available bandwidth and delay. Since the ISP can have detailed information for flows, it can also provide different pricing options. The ISP also does not need to cache anything (which could easily pose legal problems), as it influences traffic flows only. This architecture also features QoS support (R.5) and OAM integration (R.8).

7.2.4.2 Disadvantages

The open issues for this topology are that it is difficult to deploy (R.10), as it requires changes to the overlay peers, the introduction of traffic shaping equipment as well as the SIS. Moreover, since it requires the P2P applications to include support for SmoothIT, it needs to convince the developers to make this change by providing appropriate incentive (usually the end-user satisfaction) and convince the users to upgrade their clients. Finally, this approach can be misused by the ISP and produce policies that are only to the ISP's advantage (essentially reverting to current traffic-throttling mechanisms).

7.3 The Block Party: Inter-SIS Communication

This architectural option follows the main idea of introducing an information service, which resides between the overlay and the underlay network and which is enabled to exchange relevant information between neighboring Autonomous Systems.

7.3.1 Overview

An SIS can potentially make better decisions for optimizing the P2P traffic in its domain (and reduce ISP costs while providing better performance), if it can get underlay and overlay status data from neighboring ASes that also employ an SIS. SISes cooperate in pairs (Figure 7.7), but through transitivity can form groups of optimized ASes. For example, ISPs can create an Internet Coordinate system for the delay of peers belonging to various ASes [AFK07].

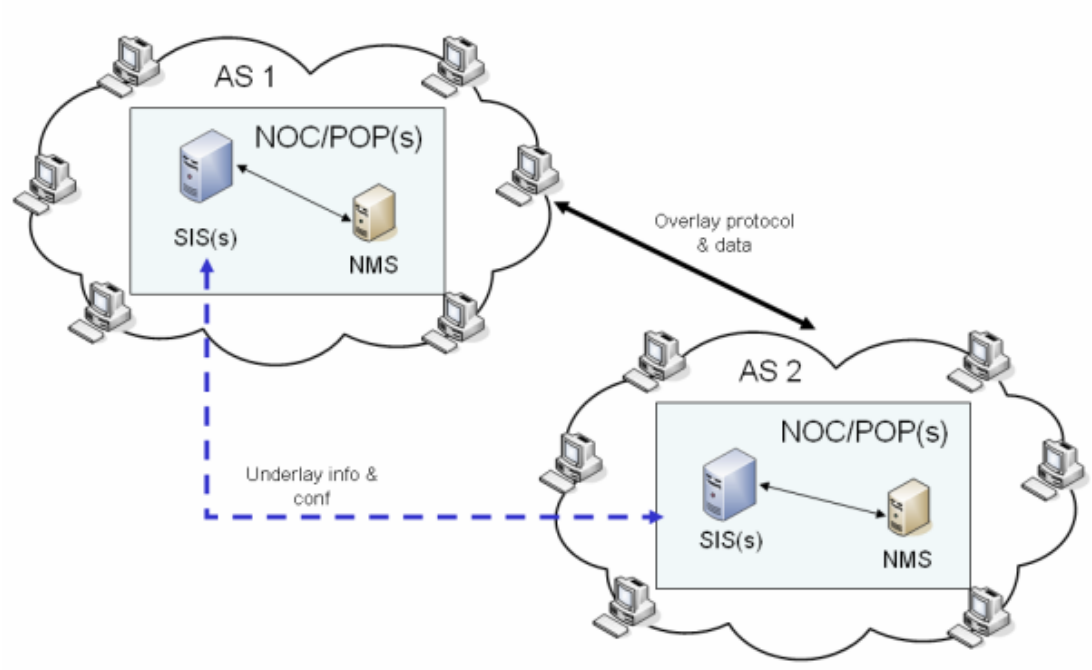

Figure 7.7 – The "Block Party" Concept

7.3.2 Structure

The required component of the inter-AS SmoothIT architecture is the extra inter-SIS interface component inside a SIS element. This component implements the inter-SIS interface (Figure 7.8).

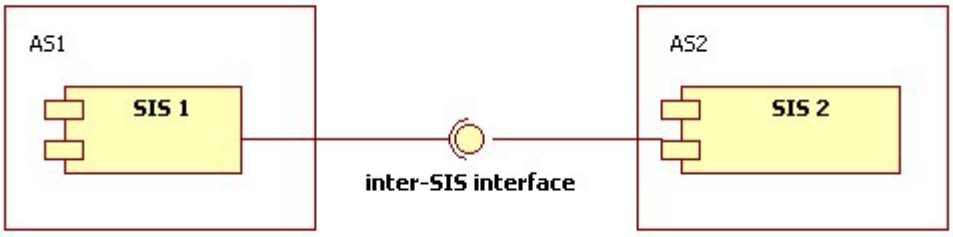

Figure 7.8 – The “block party” component diagram

7.3.3 Behavior

This architectural option’s behavior is determined by an operation and message definition.

7.3.3.1 Operation

In this scenario, both ASes have an SIS element as the core of the SmoothIT architecture. The SIS is integrated in some way with its ISP’s network. It may use one of the topologies presented before (such as the Honey Pot or the Control Freak) or some other way. In order for the SIS to come up with better decisions with its ETM algorithms, it would need a more clear view of the traffic status around its AS. If it is assumed that a neighboring AS also has the SmoothIT architecture installed (and consequently, an SIS), it would be beneficial for both of them to exchange traffic data and the status of their underlay. Therefore, in this example, the actual overlay data is exchanged between the ASes in the usual way, but a separate channel for SmoothIT control data between the SISes would be used.

Specifically for the Honey Pot architecture, the SIS of ISP 1 contacts the SIS of ISP 2 in order to see if the SIS 2 has cached content that the SIS 1 needs. This could be evaluated by the ETM mechanisms of the SIS 1 in order to see if it is in its interest to download content from the SIS 2 (and not from the overlay). More complex scenarios could be explored if the two SISes might want to share their cache in some way in order to serve peers from both ISPs. However, the cooperation and competition incentive issues need to be addressed for this to work.

Cooperation between the SISes in a Control Freak scenario can be achieved as follows. Once the SIS 1 gets a peer list request from a peer of ISP 1, it could also request a peer list from the SIS 2. The SIS 2 would compute a peer list and send it to SIS 1, which would process that list (using its ETM mechanisms) and add any peers it finds suitable to the list that it would return to the peer. Since the SIS 1 may not trust the SIS 2, it needs feedback as to the information the SIS 2 provided. Therefore, the end user peer rates each peer in the peer list and sends this information to the SIS 1. Based on the rating, the SIS 1 can select a future policy for cooperation with the SIS 2. The SIS 1 would need to provide an incentive to the end user peer so that the peer offers truthful and complete feedback.

7.3.3.2 Messages

Potential messages for this architecture include:

- SIS 1-SIS 2: Ask for cached content.
- SIS 2-SIS 1: Negative response for content, or the content itself.

- SIS 1-SIS 2: Request peer list for specific content.
- SIS 2-SIS 1: Return peer list.
- End user peer-SIS 1: Ratings for peers in peer list.

It should be noted that regardless of the actual intra-domain SIS architecture, the messages exchanged between SISes would not contain pure information regarding each other's network, as there is no incentive to do so (in fact, quite the opposite). Network status information could be extrapolated only through indirect ways (such as the suggested peer list that the one SIS offers to the other).

7.3.4 Potential

This option's potential is briefly discussed by addressing advantages and drawbacks.

7.3.4.1 Advantages

The main advantage of this architecture is that it provides inter-domain support (R.7), i.e. the ETM module of an ISP can get more data so that it can make more informed decisions regarding incentives and prices. It is also simple to deploy, as it only requires the SIS, which is presumably already installed to offer intra-domain services.

7.3.4.2 Disadvantages

An important aspect in this scenario is the type and amount of data that the SISes exchange. If it is assumed that all SISes exchange information regarding the networks and overlays around them, it makes the ISPs almost fully aware of the neighboring networks and provides them with a lot of power when making decisions.

Although this could lead to the most informed ETM decisions for optimization, it is conceivable that the different SIS could abuse this wealth of information and come to decisions that are more to the benefit of the ISPs and less to the benefit of users, especially if the ISPs have a symbiotic relationship. Therefore, it is essential that the SISes only exchange the amount and type of data that is necessary for P2P optimization by introducing appropriate methods and incentives to the data-exchange game.

However, if the ISPs are competitors – for example, if they are ISPs of the same level targeting the same geographical locations – this architecture would be difficult to implement, as those ISPs would not have incentives to cooperate.

Finally, another open issue is the kind of reciprocal actions that an ISP could offer another ISP to get it to cooperate, this would strongly depend on the type of interconnection agreement they have; therefore, in case this option would be implemented, the impact on the interconnection agreement will have to be further evaluated.

7.4 The Optimal Anarchy: Distributed ETM

This architectural option follows the main idea of rewarding and punishing peers in a fully distributed manner, where ETM mechanisms operate between peers and routers in a decentralized manner.

7.4.1 Overview

The Optimal Anarchy concept is essentially the Control Freak scenario but with distributed ETM algorithms. As a result, there is *no* centralized component that runs the ETM algorithms (the SIS, in previous scenarios). The characteristic of this architecture is the distribution of ETM among the end-user peers and ISP routers (or other ISP elements) so that traffic optimization emerges from their cooperated, but anarchic, operation. This architecture monitors the usage of links, like the Control Freak, and provides incentives to users of the overlay based on their price/performance ratio preference for each piece of content they need to consume. This architecture tries to improve congestion control for intra-domain traffic as well as lower the costs of the ISP for inter-domain traffic while preserving or improving the QoE of the end user.

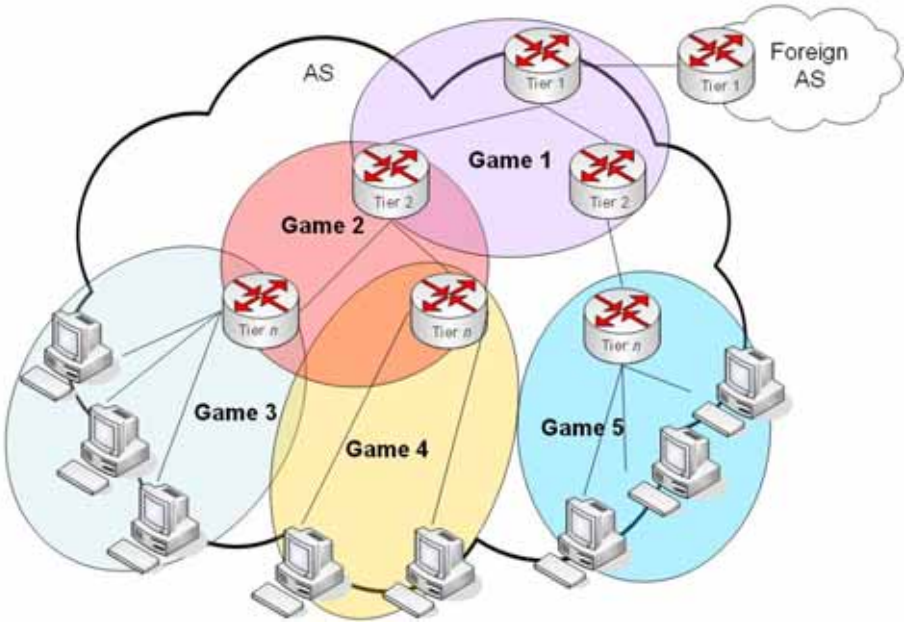

Figure 7.9 – The “Optimal Anarchy” Concept

In this scenario, the ISP has augmented its routers (or PoPs) with ETM capabilities. As a result, these routers are able to make ETM decisions regarding the QoS (and pricing) of flows at a local level. These decisions about flow priorities are taken at every level of routing, starting from the end user peer’s default gateway all the way up to the inter-domain router. In each level, the ETM decision encapsulates the decisions that were made in the lower levels. The inter-domain edge router also makes decisions for the pricing of the services provided to that user. This forms a *hierarchy* of ETM decisions, from the intra-domain edge router all the way to the inter-domain edge router. Each router makes local decisions regarding QoS for flows by using local data (and the output of the ETM) in the same way that it computes its routing table without having the full view of the network. Moreover, the end user peer has been augmented to be SmoothIT aware in order to inform the ISP about its price/performance preference. The external router of the foreign AS may or may not be SmoothIT aware. Here it is assumed that the router is not SmoothIT aware, and the other case is explained in the “block party” architecture.

Due to the fact that the ETM decision making process is distributed, there is no need for any SmoothIT component to interface with the NMS infrastructure of the ISP. This architecture is more scalable than the previous, centralized options. Moreover, the way that this option provides no strict guarantees on QoS, but propagates QoS types from

router to router, makes this scenario similar to the DiffServ architecture [CS05], but with ETM enhancements.

7.4.2 Structure

The possible structure encompasses components and interfaces to be described.

7.4.2.1 Components

The components (Figure 7.10) of this SmoothIT architecture are:

End user peer: This software connects the end user to the overlay. It also sends the end user's requests, resources and cost/performance ratio to the intra-domain edge router and receives the suggested peer list.

Intra-domain Edge router: This router receives the price/performance preference of the end users and identifies the peers that this peer wants to connect to (by checking the endpoints of the packet flows). Based on its congestion level and the requests from users, it sets priorities on the flows that it forwards by using ETM mechanisms.

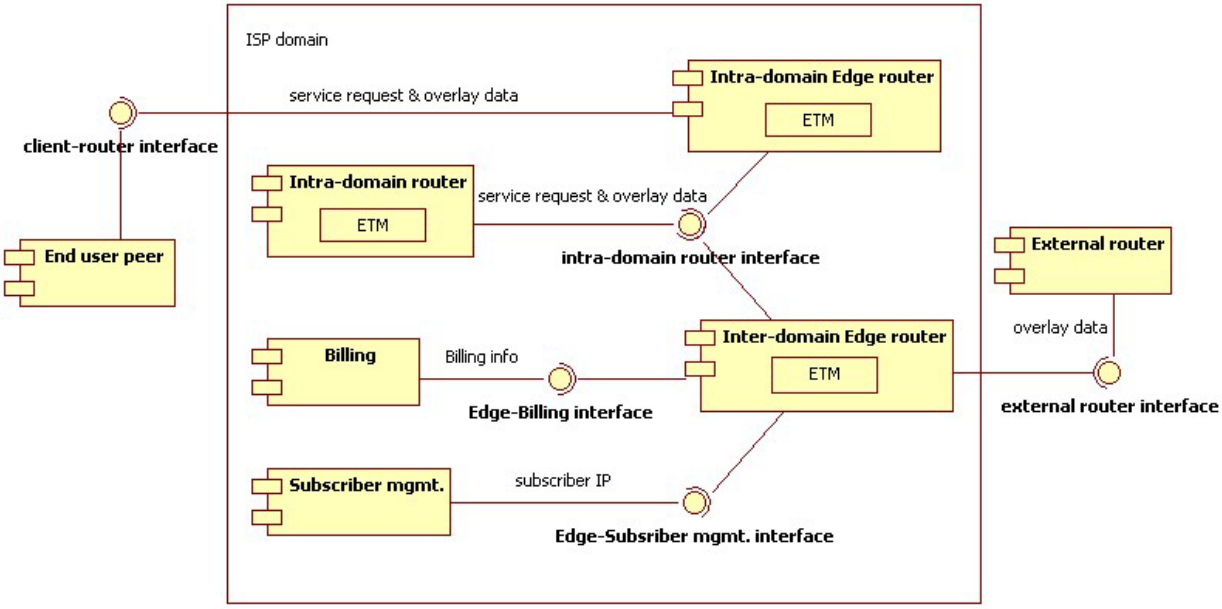

Figure 7.10 – The "Optimal Anarchy" Component Diagram

Intra-domain router: This router receives flows from other routers and based on its congestion level and flow endpoints, it alters the priority of the flows it forwards based on ETM mechanisms.

Inter-domain Edge router: This router connects the ISP's AS with a foreign AS. This router examines the endpoints, priority and other QoS characteristics of each flow and calculates the actual performance characteristics (delay, throughput etc.) that will be offered to each flow by using ETM mechanisms. Based on that, it can also set the billing information for the user that corresponds to each flow. To identify the user, it interfaces with the Subscriber Management System.

ETM: This component is part of the routers and it receives the QoS characteristics and priority and pricing/performance preference for each flow and computes the optimal overlay configuration so as to minimize the ISP cost (by suggesting intra-domain peers) and increasing the peer performance. This component on the inter-domain edge routers also computes billing information based on the performance provided to users.

Billing: This component is used in order to charge the end users based on the performance they receive.

Subscriber management: This component is used in order to map the user's IP addresses to user IDs so as to perform accounting based on the flows of each user. It can also be used to identify users, as the ETM algorithms might use reputation-based mechanisms.

7.4.2.2 Interfaces

The following interfaces are used in this SmoothIT architecture:

End user peer-intra-domain edge router: Using this interface, the end user peer transfers the overlay data as well as its price/performance preference, QoS requirements and potentially the peer list that it will connect to.

Intra-domain router-Intra-domain router: This interface is used by intra-domain routers in order to exchange overlay data as well as QoS requirements, price/performance preference and priority for each flow.

Inter-domain Edge router-External router: This interface is used in order to transfer the overlay data to and from the router of a foreign AS.

Inter-domain Edge router-Billing: The billing interface is used in order to communicate the billing information (charge) of the end user to the billing system.

Inter-domain Edge router-Subscriber Mgmt.: The subscriber mgmt. interface is used in order to get the current user ID of the end user.

7.4.3 Behavior

This architectural option's behavior is determined by an operation and message definition.

7.4.3.1 Operation

In this scenario, the ETM mechanisms are applied in each routing level. This means that there is a game between all the hosts of a routing level and their immediate parent. The intra-domain routers use the ETM mechanisms to improve their congestion level without compromising QoS requirements while the inter-domain routers also use the ETM mechanisms to calculate the pricing for users based on the cost of the inter-domain links. Based on the behavior of the ETM-enhanced routers, the users are offered incentives to optimize their traffic, resulting on lower costs for the operator (as expensive links are avoided) and improved performance for the end user (as better connections with QoS characteristics are provided). The incentive for the application developer for the P2P overlay client to support the SmoothIT architecture is that the users of the application will enjoy improved performance.

7.4.3.2 Messages

The sequence (Figure 7.11) for a peer to connect to a P2P overlay in a SmoothIT-enhanced ISP is the following:

1. The end user peer communicates with the intra-domain edge router in order to communicate its requirements, such as price/performance preference and QoS characteristics. The peer can do so by marking the flows themselves – for example, by using the Type of Service (ToS) field of IPv4 or Traffic Class for IPv6 – or by using a separate channel for this type of signaling. In the former case, the router identifies the flows once they have started and this case is far more difficult to deploy, as the routers need to be modified to process specific protocol headers in a new way. In the latter case the user needs to disclose the peers that it will connect to beforehand, along with the QoS characteristics and price/performance ratio. In that case, the router does not need to be modified - it only needs to be configured to offer specific QoS to specific flows. This case is relatively easier to deploy as the component that performs this signaling and the ETM analysis might be a separate component next to the physical router but the interface between those components is known to be too slow.
2. The intra-domain edge router runs the ETM algorithms and based on its congestion level and the flow characteristics, it modifies the priority of each flow (essentially performing admission control). This process is internal – no messages are exchanged.
3. The intra-domain edge router starts forwarding the packets of the flow with the QoS requirements it has determined.
4. The intra-domain edge router communicates the characteristics of its flows to the next router.
5. The intra-domain router runs the ETM algorithms and based on its congestion level and the flow characteristics, it modifies the priority of each flow. This process is internal – no messages are exchanged.
6. The intra-domain router starts forwarding the packets of the flow with the QoS requirements it has determined. Only packet flows are exchanged.
7. The intra-domain router communicates the characteristics of its flow to the next router.
8. The inter-domain edge router runs the ETM algorithms and based on its congestion level and the flow characteristics, it modifies the priority of each flow. It also calculates the pricing for the user of each flow. This process is internal – no messages are exchanged.
9. The inter-domain edge router asks the subscriber management system for the user ID that corresponds to the flows (IP addresses) it processes based on the IP address.
10. The subscriber management system returns the user ID to the inter-domain edge router.
11. The inter-domain edge router requests the current billing information for that user.
12. The billing system responds to the inter-domain edge router.
13. The inter-domain edge router sets the new billing information for that user.

14. The inter-domain edge router starts forwarding the packets of the flow to the external router with the QoS requirements it has determined.
15. The inter-domain edge router starts forwarding the packets of the flow from the external router with the QoS requirements it has determined.

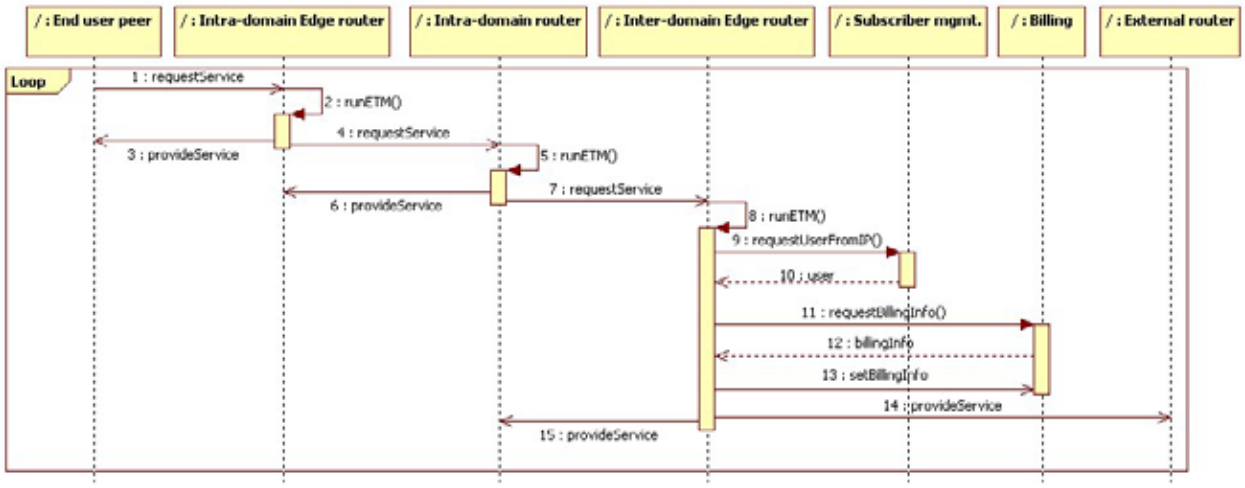

Figure 7.11 – The “Optimal Anarchy” Sequence Diagram

7.4.4 Potential

This option’s potential is briefly discussed by addressing advantages and drawbacks.

7.4.4.1 Advantages

This architecture provides incentive-compatibility to all parties (ISPs, overlay providers and end-users) (R.2). It is also more scalable as far as the ETM algorithms are concerned, since they are distributed (R.12). It supports different kinds of overlay applications (R.3) and provides different optimization schemes and modes of operation (R.6). It forgoes the need for network status gathering, as all such decisions are local and distributed without resorting to DPI (R.17), since only the endpoints of a flow are inspected. Finally, it supports limited QoS (R.5) and OAM integration (R.8).

7.4.4.2 Disadvantages

The major disadvantage of this architecture is that it is very difficult to deploy (R.10), as it requires the modification of all routers as well as the end user peer software. Moreover, it may prove to be less stable (R.14) than the centralized architectures, as the distributed nature of the algorithms might lead to decreased performance or even infinite loops and deadlock situations as well as invite Denial-of-Service (DoS) attacks (R.15). It may be slower to respond to requests as it needs to coordinate a large number of elements for each request. Finally, like the Control Freak, this approach could be misused by the ISP by using policies that are only to the ISP’s advantage (in the same way as current traffic-throttling mechanisms).

Taking into account that this solution aims to implement QoS by means of the direct interaction of the end user device with the transport plane of the ISP network, it seems very similar to IntServ architecture [BCS94], which was not widely implemented since the

information stored on each node strongly depends on the number of reservations [BBC+98], leading to scalability problems in the aggregation nodes. However, since there is no real end-to-end admission control or tracking of the flows (but only on a per-tier level), scalability should be better than IntServ and more akin to DiffServ. Of course, the disadvantage in this case is that there are no hard QoS guarantees.

Moreover, an important issue to be taken into account is the difficulty to develop an interface able to interact with the edge routers. As stated in [CE08], one of the major problems to deploy QoS capabilities in current network architectures is the lack of standardized interfaces in commercial network equipments; this means, that depending not only on the network technology but also on the vendor equipment, different interfaces would have to be developed in order to make possible the interaction between the end user application and the edge router.

8 Overall SmoothIT Architecture

This section presents the initial design of the SmoothIT architecture by specifying main components, their functionality, and their interactions. In order to take advantage of the different design solutions presented in the previous section, a comparative analysis of the proposed solutions has been performed in order to infer the best solution to the problem SmoothIT is addressing or at least to select the best combination of all the solutions.

The following attributes have been selected in order to perform the analysis of the solution:

- Legal issues are considered in order to evaluate the viability of the solution. If the solution is considered as non-legal, the solution will not be deployed and cannot be considered as a good option to be developed.
- The feasibility to deploy the solution is also considered and it evaluates how feasible it is to deploy a solution in current operational environments (both the network and overlay applications).
- Complexity and scalability of the proposed mechanisms: even though a solution could be considered as a good option due to its capabilities to improve the performance of the overlay and to support ISP in the efficient management of overlay traffic, if the solution is demonstrated as too complex (in terms of, *e.g.,* requiring too much computation time) or non-scalable (the solution is not able to meet a set of performance requirements), the solution will not be deployed.
- Optimization potential: whether the design proposal can lead or not to real improvement in overlay performance and a reduction of ISP costs.
- The innovation proposed in each design is also evaluated. Since SmoothIT is a research project, the final decision must be based on emerging issues and not on well-known and tested options.

According to the defined attributes, the following table shows the analysis of the solutions proposed in the previous section.

Table 8-1: Design Space comparison

	Legal issues	**Deployability**	**Complexity and Scalability**	**Optimization potential**	**Innovative**
Honey Pot	A clear drawback due to caching of non-legal content	Easy to deploy (no migration for end users) but a new interface to trackers could be required to get the information about the most popular content	Scalability concern depending on the amount of content to be cached and the number of users	Very high (a new peer with high BW working as seeder for multiple content is added)	Not really innovative. Commercial products are available but not deployed due to legal issues
Honey Pot + Content Analyzer	None	Easy to deploy (similar to current existing solutions). Concerns about its applicability	The detection of legal/non-legal content seems a complex task that could face performance	Very high (a new peer with high BW working as seeder for multiple content is added)	The content analyzer is a clear innovative challenge

		for multiple overlay applications	problems in large scale networks.		
Control Freak	None	It needs changes in the overlay application. Additionally, if QoS is required, it needs AAA, traffic shaping, and QoS enforcement.	NAT traversal (how to manage multiple NAT scenarios in the overlay-SIS interface). Scalability concerns can appear in case of too much centralization of the algorithms	High (multiple incentives can be provided)	Innovative (QoS and pricing are included)
Block Party	None whenever the exchanged information does not violate end users privacy If the domains exchanged cached contents, legal restrictions described in the first 2 options are also applicable	Difficult to achieve an inter-domain collaboration.	NAT traversal in inter-domain protocol Definition of parameters that do not provide details of each topology	High (but also depends on the used algorithm)	Innovative (interaction between different domains is always a technical challenge)
Optimal Anarchy	None	Difficult since it requires modifications in routers and the end user has to interface to a network router, it would be difficult to define a standardized interface valid for any vendor equipment	Provisioning of decentralized CAC is very complex. Problems of DoS (Denial-of-Service) attacks.	High	Very innovative

According to the analysis shown in Table 8-1, the following *learnt lessons* can be inferred:

- The usage of caching based solutions (*Honey Pot* scenario) will be only considered for legal content. Since the component to infer the legality of the content seems very complex and a possible bottleneck, this solution will be only considered when the ISP has a service agreement with a content provider.
- Even though the *Control Freak* based approach is complex due to QoS support and traffic shaping, it is the option that provides the capability to add new components and, therefore, incentives. Since this architectural approach provides

the capability to incorporate multiple mechanisms (to be developed in the framework of WP2), it can be considered the starting point to build an extensible SIS.

- Based on the *Optimal Anarchy* approach that aims to implement decentralized algorithms, the SIS should take advantage of decentralized algorithms in order to build the final solution to, *e.g.,* to enforce QoS policies in the network.
- The main deficiency of the *Optimal Anarchy* approach is that it possibly requires the modification of routers in a network, making its deployment very difficult. It needs a standardized interface implemented by different network equipment vendors, otherwise the solution is not portable from one network environment to another. Therefore, the SIS architecture should require only limited changes in the network infrastructure.
- Finally, following the *Block Party* approach, SIS will take the challenge of defining a SIS-SIS interface in order to allow the coordination between different domains.

8.1 Top-level Architecture

Figure 8.1 depicts the SmoothIT high level architecture. The architecture covers the options discussed in Section 7.

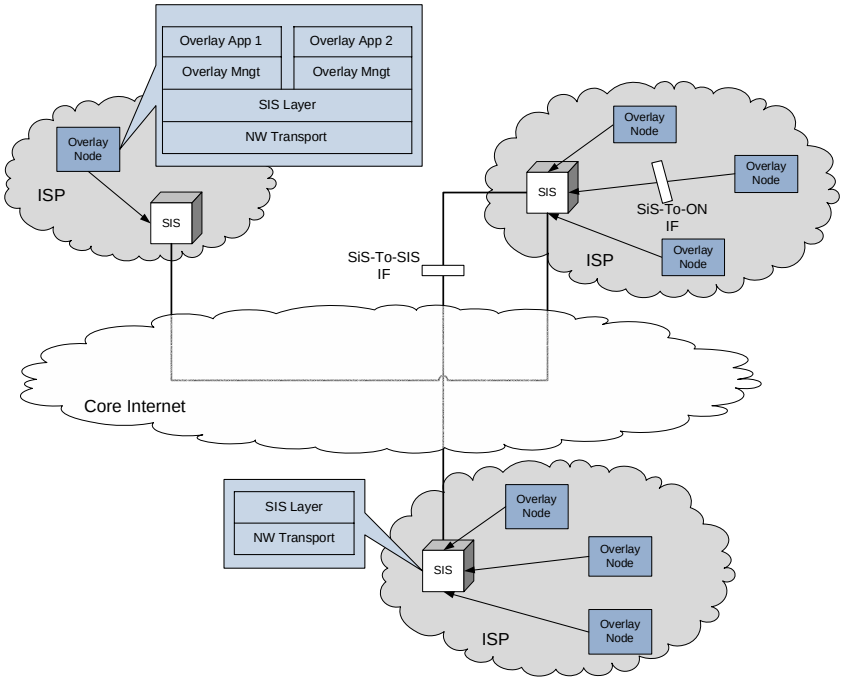

Figure 8.1 – Top-level Entities

Every ISP maintains a component called SIS. This component is the core of the architecture. It is responsible for providing support to the overlay formation process so that both the overlay and the ISP benefit. This must hold irrespective of whether the overlay spans multiple ISPs or the SIS components of multiple ISPs communicate. However, as the figure shows, we expect that all non-tier-1 ISPs will deploy their own SIS as there is a clear incentive for that and so the communication between them will be necessary. Note that SISes can be implemented as distributed systems, although the figure shows each of them as a single box.

The figure also shows that the SIS should effectively operate as a middleware component which can be plugged into any overlay application between the overlay management layer and the network layer. Different applications normally have different QoS requirements and the SIS must be able to tune its operation according to the requirements obtained from the application.

8.1.1 The SmoothIT Information Service (SIS)

The SIS is a distributed system that provides information from the ISPs to overlay applications and also vice versa. The service may provide information about policy, locality, congestion, and QoS, for example, to help overlay applications decide how to establish connections and how to use them. The information on the popularity of content, for example, may be tracked in the overlay and could be utilized by the ISP, *e.g.*, to cache popular contents.

The information can be useful in many ways for the overlay application. One possible use is at the peer selection process of overlay applications. While service discovery is a role of each overlay application, once it has discovered a set of hosts that can provide a certain resource – *e.g.*, a file or a video stream – it may query the SIS to decide from which hosts to get the resource.

Each SIS domain corresponds to the network domain for which one particular SIS Server is responsible to provide the SmoothIT Information Service. The local network domain may be the network of an ISP, and the SIS would be provided by the ISP.

An instance of a SIS Server can further request preference information from other SIS Server instances deployed in other network domains. Although this server-server interaction can be used to further refine provided preference information to include the preferences for peers belonging to other network domains, it is not a required feature for SIS to operate.

A basic SIS implements an RPC-like service that receives from the overlay application a list of IP addresses and any other optional information that may influence the reply. In general the SIS can be offered by the ISP or by a third-party. However, since the SIS needs privileged information about the underlay network, it is most likely deployed by the ISP. Based on the information about the network, the SIS can append to each given IP address a preference value and send it back to the overlay application. The overlay application can use this information in the peer selection process to give preference to higher-scoring peers.

The SIS does not simply order IP addresses from the request according to the preference metric, but does include a preference value for each address in the reply. Attributing a preference value to each address makes it possible for the client to cache and compare preference values among different requests.

8.1.2 Functionality

According to the functional requirements detailed in Section 6.1, this section identifies the set of functionalities to be provided by the SIS. Since the SIS aims to support multiple overlay applications, multiple working functionalities should be provided in order to address these different requirements. In particular, the following functionalities must, at least, be provided:

- Traffic engineering according to the ISP policies and to the network status. This means that the SIS will provide an interface to the network administrator, where the preferences will be shown and configured in the ETM mechanisms.
- In order to let the SIS be aware of the network status, it is also important to integrate the OAM capabilities of the domain.
- Enhanced connectivity capabilities: since the SIS aims to provide win-win solutions, it is also important to guarantee the network performance. As an important way to provide the QoE to the end users, the SIS takes advantage of the advanced capabilities of the NGN (Next Generation Networks) architectures as a first step, also ad-hoc solutions can be incorporated to the system.
- Communications between different SIS systems must be supported. This interface must be available for both intra-domain scenarios (*e.g.,* communication between SISes deployed in different PoPs of the same operator) and inter-domain scenarios (SISes deployed in different operators' networks can communicate between themselves).
- Accounting functionality will be provided by the SIS in order to charge premium services (*e.g.,* QoS guaranteed services).

In order to complete the specification of the SIS functionalities, it is also important to identify the way these functions are provided to external entities. In particular, at least the following external interactions are foreseen in the SIS implementation:

- First of all, an important interaction is the one performed by the network administrator to configure its preferences and the access policies to its network modules or network management systems, such as BGP reflectors, NGN components, etc.
- Overlay nodes will send a request to the SIS and the SIS will attach a preference value to each peer in the list according to different criteria (*e.g.*, reputation, throughput, delay, and locality). Moreover, the SIS could also enforce some kind of QoS mechanisms for specific sets of selected peers.
- A service provider negotiates a service agreement with the ISP and the network administrator configures network resources using the SIS admin interface.

8.2 Components

The components of the SmoothIT architecture are shown in Figure 8.2. The SIS Server provides an interface to the SIS Client, implemented in the overlay application, to the network administrator (admin component), and to other instances of a SIS Server, which may be located in another domain. The SIS Server uses services from other components which are the QoS Manager, the Metering, the Security and the Configuration Database. All these components of the architecture are described in the following sections.

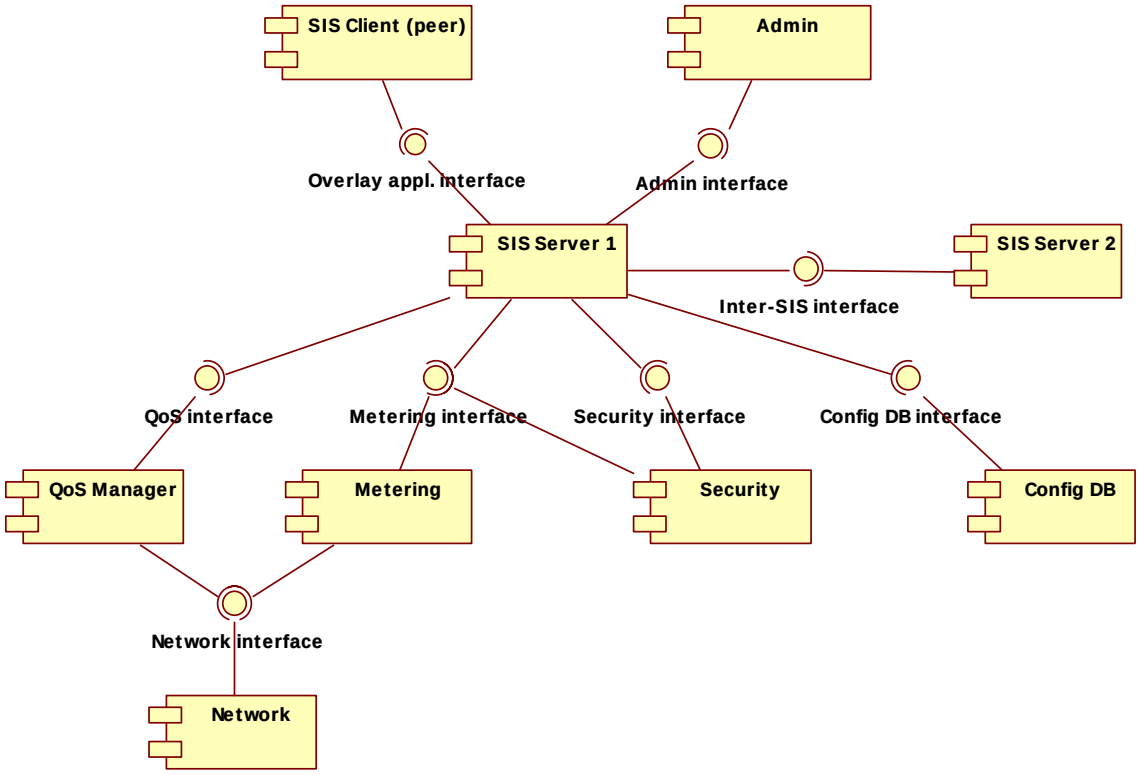

Figure 8.2 – Components of the SmoothIT Architecture

8.2.1 SIS Server

The SIS Server is the core of the SmoothIT architecture. Its main responsibility is to receive the request from the overlay application, perform calculations based on several factors, such as metering and policy information, and reply the preference values back to the overlay application.

The calculation of the preference values is influenced at least by the following factors:

- overlay application requirements and characteristics,
- underlay network conditions, and
- ISP policy.

In order to measure current conditions of the network, one or more metering components must be connected to the SIS Server. Examples of metering components are a BGP Information Module or a module for measuring latency or jitter.

ISP policy is configured through the Administrative interface, and influences the ranking of peers, for example, to avoid certain paths. The Administrative interface is used to configure any parameters of the preference calculation procedure.

8.2.1.1 The Preference Metric

The preference metric is used by the SIS Server to assign a preference to host addresses requested by the SIS Client. A preference value is expressed as a positive value, allowing a client to compare host addresses, where a higher value represents higher preference.

Additionally, a value or symbol is required to represent *no information available*, in case the SIS Server has no means to determine. The metric should allow a total ordering of IP addresses according to the associated preference value.

The preference value P used in the information service interface of SIS can be represented as an integer value such as

$$P \in \{x \in N : 1 \leq x \leq 100\}$$

To represent *no information available* the integer number 0 can be used. The preference value representation can then be directly interpreted as a greater than zero percentage value of maximum preference and is easily human readable.

8.2.1.2 Peer Ranking based on BGP Information

The SIS can perform peer ranking and sort IP addresses of potential peers based on the BGP information retrieved from the BGP Information Module of the Metering component. The ranking takes into account the local preference, the AS hop count, and the MED (Multi Exit Discriminator) BGP attributes. Based on these attributes the SIS can sort IP addresses and it can also assign a preference value to each IP address. The ranking procedure can be used to differentiate peers from outside the AS of the ISP.

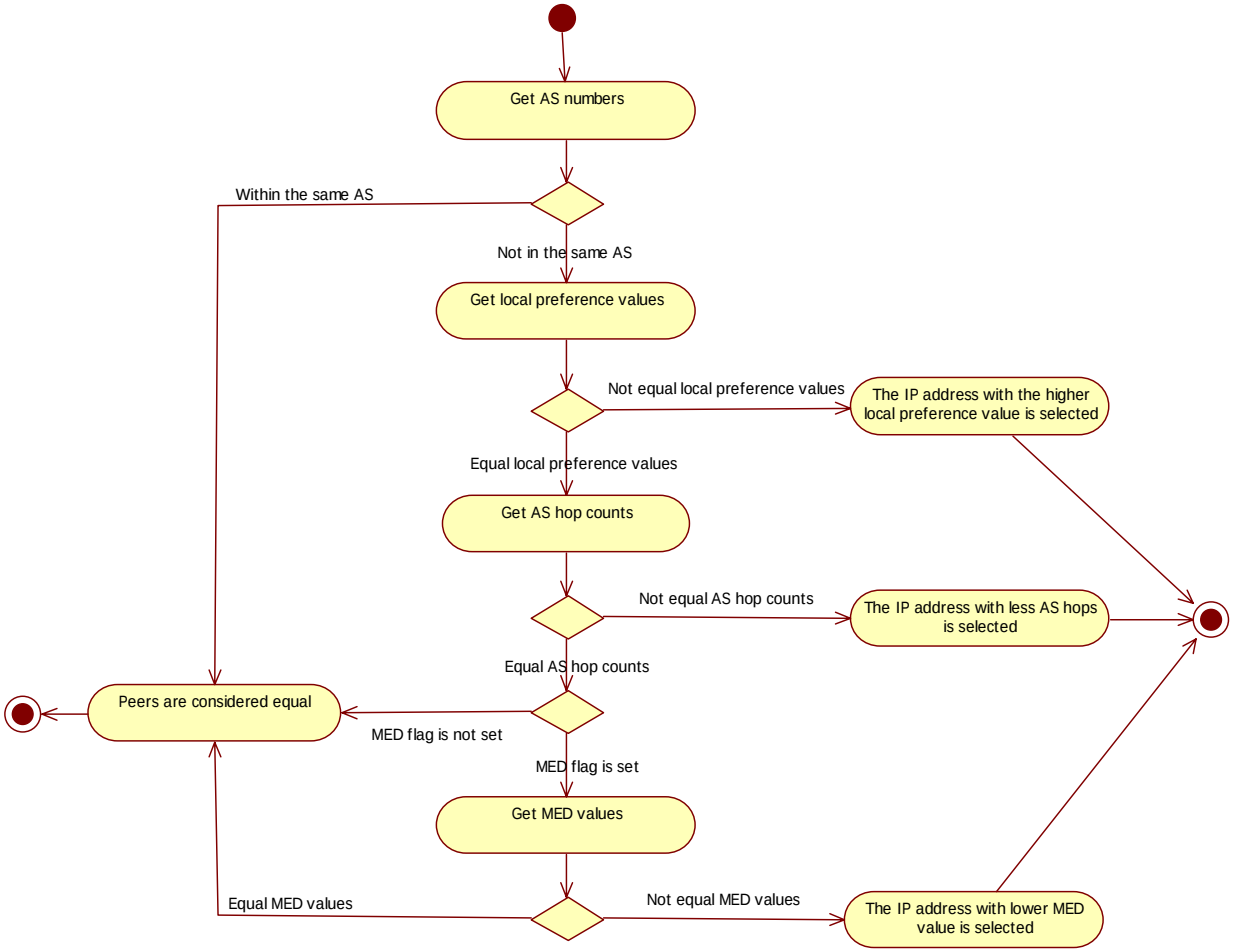

Figure 8.3 – Peer Ranking Algorithm

Given two different peers by their IP addresses, the peer ranking algorithm can be demonstrated by the following steps that are also illustrated in Figure 8.3:

- Two peers within the same AS are considered equal.
- If the peers are from different ASes, a peer with higher local preference value is ranked higher in the preference list of the ISP. This is because the local preference is set by the ISP according to business relations with other ISPs and it represents the preference of the ISP.
- If the routes to the peers have the same local preference value, then the peers are compared based on the AS hop counts. The peer that can be reached in less number of AS hops is selected and ranked higher. According to this ranking, the traffic of overlay applications has to cross less ASes, resulting in possible performance benefits for the application.
- If the peers are reachable in the same number of AS hops, the MED attribute is used. Generally routes from different ASes are not compared on the basis of MED value, except in special cases where a group of ISPs use a common policy to set the MED value. Therefore, the MED attribute is only used to differentiate peers if the MED flag is set.

8.2.1.3 Peer Ranking Using the Config DB

The SIS can perform peer ranking and sort IP addresses of potential peers by using the information stored in the configuration database as well. Compared to the ranking based on BGP information, this ranking can be also used in the case of intra-domain peers (peers that are located in the same network as the requesting peer) as well as it can consider performance related metrics stored in the database. While the BGP-based ranking can only differentiate between peers that are outside the AS of the requesting peer and therefore, all intra-domain peers are considered equal.

The ranking takes into account the IP address of the requesting peer (source address), the IP address of the potential peers (destination address), the business relation between ISPs in case of inter-domain communication, and the link capacity and delay to the destination. The SIS reads all entries from the configuration database (see Section 8.2.2) that matches the source and destination addresses and compares the relation, link capacity, and delay attributes of each entry. The SIS can then rank peers according to the relation, link capacity, and delay attributes depending on the SIS configuration and the requirements of the overlay application.

8.2.1.4 SIS Server Discovery

To be able to connect to the SIS server, peers in the overlay application need to get or resolve the IP address of the SIS server in some form of a service discovery. In the SmoothIT architecture the following discovery mechanisms are considered:

- Static configuration of the IP address or the name (Fully Qualified Domain Name) of the SIS server. In the case when the name is configured, peers can resolve the IP address by a DNS (Domain Name System) lookup. The name resolution through DNS also allows for load balancing by dynamic DNS replies, where the DNS server replies with addresses of different SIS servers to client requests.

- Peers can get the IP address or the name of the SIS server automatically over DHCP (Dynamic Host Configuration Protocol). The advantage of this approach is that peers do not need to configure anything statically. But this approach has the disadvantage that DHCP has to be adapted and the new version has to be widely deployed.
- Peers can use service discovery and send their discovery requests to a specific multicast address where a server is listening and answers the request. The peer can get the IP address or the name of the SIS server.

8.2.2 Configuration Database

The Configuration Database (Config DB) stores ISP policies and it is responsible for any information that an ISP can configure for the SIS architecture. The database can be based on an existing repository of the ISP and it contains information about different types of business relations the ISP has with other ISPs and performance related metrics corresponding to each network. The database includes the following data:

- src_prefix: the source IP prefix of the source network (the network of the ISP configuring the database).
- src_prefix_len: the prefix length of src_prefix.
- dest_prefix: the destination IP prefix of the destination network (the network the candidate peers are located in).
- dest_prefix_len: the prefix length of dest_prefix.
- relation: the type of relation the ISP has with the ISP represented by dest_prefix. The relation can be “provider”, “peer”, or “customer”.
- link_capacity: the capacity of the link to the network represented by dest_prefix.
- delay: the delay to the network represented by the dest_prefix.

The database provides an interface to access the stored information maintained by the ISP. The SIS component can read this information and use it in the traffic management and peer selection procedure.

8.2.3 Metering

The metering component is responsible for collecting any information from the network that is required by ETM mechanisms and components of the SmoothIT architecture. This information is provided to other components in the form of *metering data*. Based on the metering data received from the metering component, the SIS can perform its task and assist overlay applications in their peer selection process. Metering data are also used for accounting and monitoring of overlay applications.

The metering data can include information about the network, like its current state, its topology, and its performance metrics. The metering component of SmoothIT includes the BGP Information Module, the Performance Meter Module, and the Usage Meter Module, as shown in Figure 8.4.

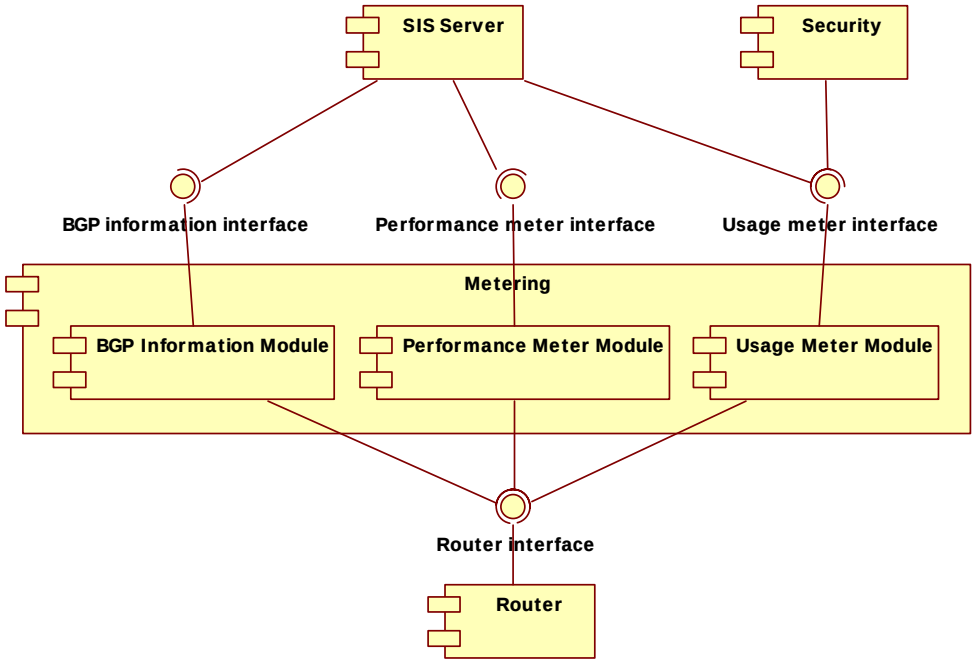

Figure 8.4 – Metering Component

8.2.3.1 BGP Information Module

The BGP information module collects BGP routing information from BGP routers of the ISP and provides this routing information to the SIS. BGP is an Exterior Gateway Protocol (EGP) used for routing information exchange between Autonomous Systems (AS). An AS is a group of networks under the common administration of a single network operator and having a common set of routing policies. ISPs implement inter-domain routing based on BGP according to their business relations with other ISPs. Therefore, the BGP routing information can be used by the SIS to assist inter-domain peer selection.

BGP selects the best route to a destination from several possible candidate routes. The decision is based upon different attributes associated with each route. These attributes are the following:

- Weight: The weight attribute is a Cisco-defined attribute and its value is local to the router and represents the weight given to a route. The weight attribute is not exchanged between routers and therefore weight values of two different routers cannot be compared.
- Local Preference: The local preference is an indication to the AS about which path has preference to exit the AS in order to reach a certain network. Routers in the same AS exchange the local preference value. Routes with higher values of local preference are preferred than routes with lower values of local preference. Generally ISPs assign high values of local preference for links that cost less.
- AS Path: It is the list of ASs encountered on the way to reach the destination. Routes with less number of AS hops are selected.
- Origin: The value of the origin attribute specifies how a route is learned. It has the following three possible values:

- IGP (i) – The route is interior to the originating AS.
- EGP (e) – The route is learnt from exterior gateway protocols.
- Incomplete (?) – The origin of the information is unknown. It usually happens when routes are redistributed from IGP.

- Multi Exit Discriminator (MED): MED is a hint to external neighbors about the preferred path into an AS that has multiple entry points. The MED is also known as the metric of a route. A lower MED value is preferred over a higher value.

The BGP Information Module collects BGP routing information from one of the BGP routers of the ISP. It retrieves the routing information via SNMP (Simple Network Management Protocol) from the router. Specifically the BGP sub tree of the router starting at OID (Object Identifier) .1.3.6.1.2.1.15.6 (containing the BGP routing information) is parsed to read the routing table.

The BGP Information Module provides an interface to read the BGP routing information. Via this interface the SIS can access routing information and use this to provide locality information to overlay applications. The interface allows the retrieval of the routing entry, including all BGP attributes, for a given IP address.

8.2.3.2 Performance Meter Module

The Performance Meter Module is responsible for collecting any performance related information from the network, like traffic load on links, packet loss, and latency. The Performance Meter Module provides this information to the SIS. The SIS in turn can use this information for traffic management purposes, *e.g.*, select least loaded links, or select low latency connections for real-time overlay applications.

The Performance Meter Module can be based on existing metering infrastructure of the ISP and it performs both passive and active measurement in order to gather performance values of the network. By using passive measurement the Performance Meter Module observes the traffic at several places in the network and gathers statistical information. It can read information from routers in the network or from separate traffic meters deployed in the network. As shown in Figure 8.4, it can read information via SNMP from several routers in the network, like traffic load on interfaces, packet loss, and traffic counters.

By means of active measurement, the Performance Meter Module injects probe packets into the network for measurement purposes. It can measure performance values like the latency and available bandwidth of a certain path. It can be queried for measured performance values or it can send a notification if a predefined threshold is exceeded.

8.2.3.3 Usage Meter Module

The Usage Meter Module collects information on the network usage, i.e. the traffic volume transferred in the network. It provides this information to the SIS and the accounting function in the Security component.

Network usage data can be collected by the Usage Meter Module from routers in the network or from separately deployed traffic meters. Usage data can be retrieved from routers via SNMP and NetFlow. Via SNMP the Usage Meter Module can read traffic counters of each interface of routers in the network of the ISP, while via NetFlow it can gather information separately for each traffic flow. The Usage Meter Module supports metering per user, per overlay application, or for traffic aggregates.

8.2.4 Security

The Security module is a component which is responsible for security assurance. It provides the basic security services which are based on the following requirements: authentication, access control, non-repudiation, and data integrity. The Security component consists of two modules: AAA (Authentication, Authorization, Accounting) server and data integrity module (Figure 8.5).

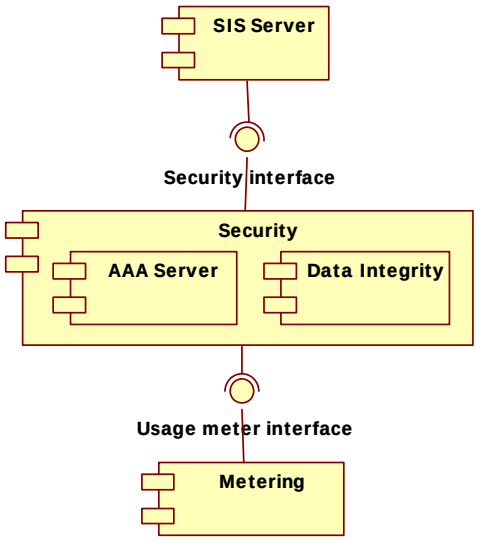

Figure 8.5 – Security Component

Also, the Security component is connected to a) the metering component which is responsible for collecting any information from the network and b) the SIS server to protect data against threats. Below, some of possible threats and attacks are presented:

- eavesdropping (an intruder listens to things he or she is not supposed to hear),
- masquerade (an intruder pretends to be a trusted user),
- authorization violation,
- modification or forgery of information.

To protect the data against passive and active threats, AAA server and data integrity module will be deployed in the Security component.

8.2.4.1 AAA Server

The AAA server is implemented in Security component. It stands for authentication, authorization and accounting standardized by IETF (Internet Engineering Task Force). There are various protocols defined by IETF but in the Security component the most popular RADIUS protocol or the new Diameter protocol [CLG+03] will be deployed.

Authentication service is always required when administrator or another SIS module wants to establish a connection. In this situation, validation of the identity is obligatory. Several methods of authenticating an entity can be implemented in the Security component such as static and one-time passwords, or challenge-response methods.

Authorization service deployed in the Security component creates access control mechanisms in order to ensure proper rights the entities have.

The Security module is connected to the Metering module which collects information on the network usage and provides this information to the accounting function.

8.2.4.2 Data Integrity

Data integrity module ensures message integrity during remote connections between two SISes or SIS and system administrator. In order to create secure connections, several algorithms will be deployed in this module such as MD5 or SHA-1.

The data integrity module could also support encryption. The encryption service will be used when:

- message integrity is not enough from security point of view
- the efficiency of the system is sufficient (ciphers are more complex then hash algorithms).

To ensure message confidentiality, symmetric ciphers or solutions based on PKI (Public Key Infrastructure) architecture will be deployed in the module.

8.2.5 QoS Manager

This module aims to check the availability of network resources and to guarantee resources requested by the end user as well as to enforce the QoS policies in the network. It must also provide a well known set of methods to the SIS Server module.

One of the main advantages of the SIS is that the design is modular, so this module can be developed by any ISP according to the mechanisms it has developed in its network. Therefore, if an ISP has its own platform to enforce QoS policies in its underlying network, the QoS Manager will be in charge of interfacing this platform and providing a common interface to the SIS server. Moreover, if the ISP has no platform in charge of managing QoS policies in its network, this QoS Manager will be developed in order to provide QoS incentives to the overlay networks; in this case, ad-hoc solutions will be developed taking into account the commercial equipment deployed in the ISP network (*e.g.,* in order to guarantee low delay, the QoS Module will interface an IP DSLAM in a xDSL access).

An important case to be taken into account is the integration with NGN architectures. Effectively, multiple ISPs are studying the deployment of NGN Control Planes in their networks in order to provide carrier class services over IP infrastructures. SmoothIT will take advance of this deployment in order to build their QoS mechanisms.

In order to implement this, it is important to be aware of the current limitations to QoS in current NGN architectures as they are described in [CE08]. In particular, one of the major limitations in current architectures is the lack of standardized configuration interfaces in current network equipment. Since SmoothIT aims to re-use existing standards and interfaces, the QoS solution to be implemented in the SIS will rely on the current QoS developments done in the field of NGN (taking as reference the work being done in ITU-T (International Telecommunications Union – Telecommunications Sector) and ETSI/TISPAN).

In order to achieve this goal, the QoS Manager will be composed of the following components: (i) the Interface to the SIS Server, (ii) a SIS QoS Core, where specific SIS

policies can be applied (*e.g.,* the administrator does not allow to reserve resources for more than 1 Mbps) and (iii) the interface to the NGN equipment.

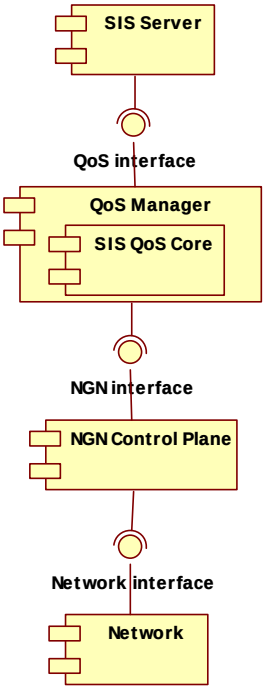

Figure 8-6 – QoS Module Components

In the following subsections, the different components will be described, but before that, in Section 8.2.5.1, the network performance requirements will be described in order to know in advance which capabilities can be expected from the network.

8.2.5.1 QoS Capabilities

First of all, the QoS Manager needs to define a set of performance objectives in order to provide a set of well-known incentives to the end users. The way to implement end users expectations in the network is by means of defining classes of services that allow the network operator to manage the traffic per aggregate. Therefore, as a first step, the SIS will need to know in advance which classes of services will be provided (*e.g.*, low-latency or streaming).

In order to support this decision, the ITU-T Y.1541 [Y.1541] defines the following QoS classes of services (taking into account the network performance parameters at the IP packet level):

Table 8-2: ITU-T Y.1541 QoS Classes

Network Performance Parameter	Classes of QoS					
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
IPTD	100 ms	400 ms	100 ms	400 ms	1 s	N/A
IPDV	50 ms	50 ms	N/A	N/A	N/A	N/A
IPLR	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}	N/A

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

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

This classification does not mean that all these classes must be implemented in each network (in fact, all these requirements could be implemented with just 2 classes of services: Premium and BE (Best Effort), and the Premium would meet the Class 0 requirements or the operator will just overprovision its IP network). But with this classification, the network administrator can select the way the different services will be provided.

The classes of services to be used by the SIS will strongly depend on the commercial equipment available, the set of services the ISP will provide and it will be a parameter that can be configured through the admin interface.

8.2.5.2 QoS Manager Interface

This interface will provide to the SIS Server a set of well known methods valid for any QoS Manager implementation. In particular, the following primitives must be available:

- *Reserve_resources request* will be sent from the SIS server to the QoS Manager in order to request the resource reservation for a set of flows. Therefore, in the request, the QoSRequest must be included. This object is composed of the flow(s) description and of the class of service that must be enforced for this set of flows.
- *Reserve_resources response* will be sent from the QoS Manager to the SIS server with the response to the request. This response will contain a reservation_id if the resources have been successfully reserved and a null value if the reservation was not done
- *Modify_resources request* primitive can be sent (this method can be considered as optional) if the SIS Server needs to modify the reservation. In this request, the reservation_id and the new QoS Request must be provided.
- *Modify_resources response* will be sent by the QoS Manager with the result to the modification request.
- *Release_resources request* is used by the SIS Server to release the resources reserved. It has to specify the reservation_id.
- *Release_resource response* is used by the QoS Manager to notify the result of the release request.

The following sequence diagram shows how these messages are exchanged.

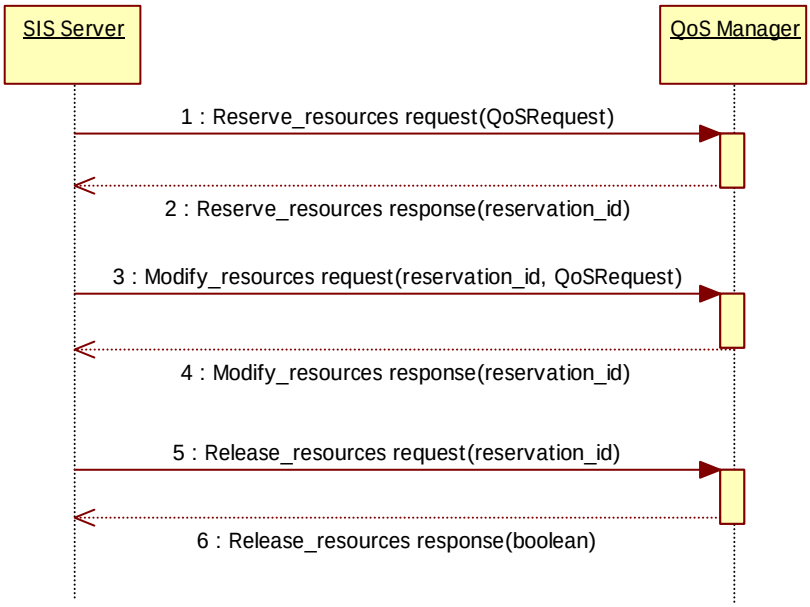

Figure 8-7: SIS Server and QoS Manager interaction

As shown in Figure 8-7, the interaction between the SIS Server and the QoS Manager is based on request-response transactions. So, even though the QoS Manager can be implemented as a software module managed by the SIS Server, it can be also implemented with communication protocols, such as SOAP (Simple Object Access Protocol) or COPS (Common Open Policy Service).

8.2.5.3 QoS Core Module

This part of the QoS module is in charge of applying specific policies for each request. The policies to be applied could be:

- Maximum bandwidth (for all CoSs (Class-of-Service) or for specific CoS)
- Maximum number of reservations
- Maximum number of sessions

Therefore, when the QoS core receives a request from the SIS Server through the SIS Server-QoS Manager interface, it will firstly check if the QoS request fulfills the requirements to be configured by the network operator.

8.2.5.4 Interface to the NGN

As a first step, it is important to know the current status of the development of current NGN architectures. Figure 8-8 and Figure 8-9 show the NGN architectures as they are defined in ITU-T [Y.2111] and ETSI/TISPAN [ES003] respectively.

As can be seen in the above figures, the main planes of the current NGN architectures are the following:

- Application/Service Plane (Service Control Functions or Application Functionalities) that are in charge of negotiating with the end users are aware of the application/session characteristics. This level could be implemented as an IMS (IP

Multimedia Subsystem) core (Session Initiation Protocol Proxies) or as a service provider platform.

- Control Plane (such as NASS, RACS): this plane binds the network specific issues to the application plane. It is in charge of managing the end user profile, performs admission control and interacts with the transport plane.
- Transport Plane: where the different network equipments with their own capabilities are deployed.

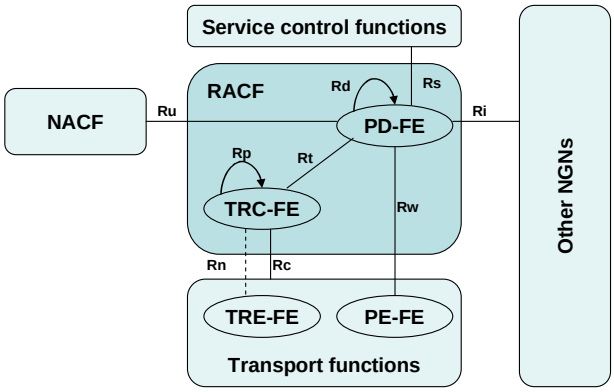

Figure 8-8 – ITU-T NGN Architecture

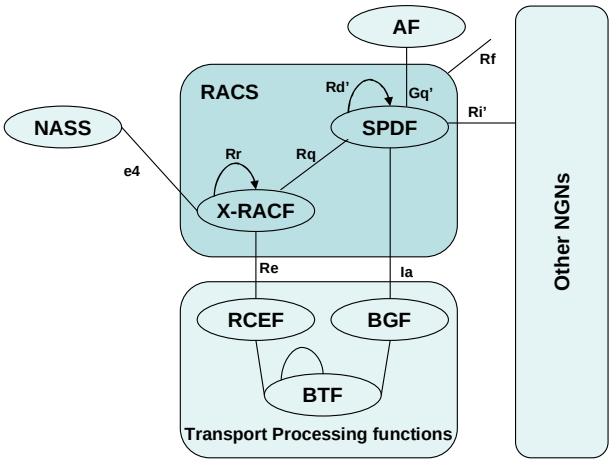

Figure 8-9 – ETSI/TISPAN NGN Architecture

Taking into account this distinction, the SIS modules will work as an Application Plane component that will interact with the Control Plane of the NGN (mainly with the RACS or RACF). Therefore, the SIS interface with the NGN will be an implementation of the Gq' interface (as defined in the ETSI/TISPAN standards).

8.3 SIS Server External Interfaces

Interaction with the SIS Server must be done through the specified interfaces for easier integration with existing technology. There are three interfaces that allow for communication between an external element and the SIS Server. These are 1) the interface between overlay applications and SIS Servers for exchanging information between overlay and underlay (*e.g.*, to get preference information), 2) the administrative

interface to configure policy information by the ISP, and 3) the interface between two SIS Servers for intra- or inter-domain communication.

8.3.1 Overlay Application – SIS Server

The interface between overlay application and SIS server enables the overlay application to query the preference information from the SIS server via the SIS protocol. The schematic figure of the SIS protocol interaction is shown in Figure 8-10, while the UML (Universal Modeling Language) representation of the interaction and the SIS request and reply messages are shown in Figure 8-11 and Figure 8-12, respectively. The SIS protocol is stateless in general but session state handling can be included through extension fields. The SIS protocol follows a request-reply interaction scheme. The interface between overlay application and SIS server defines a basic and an extended interaction.

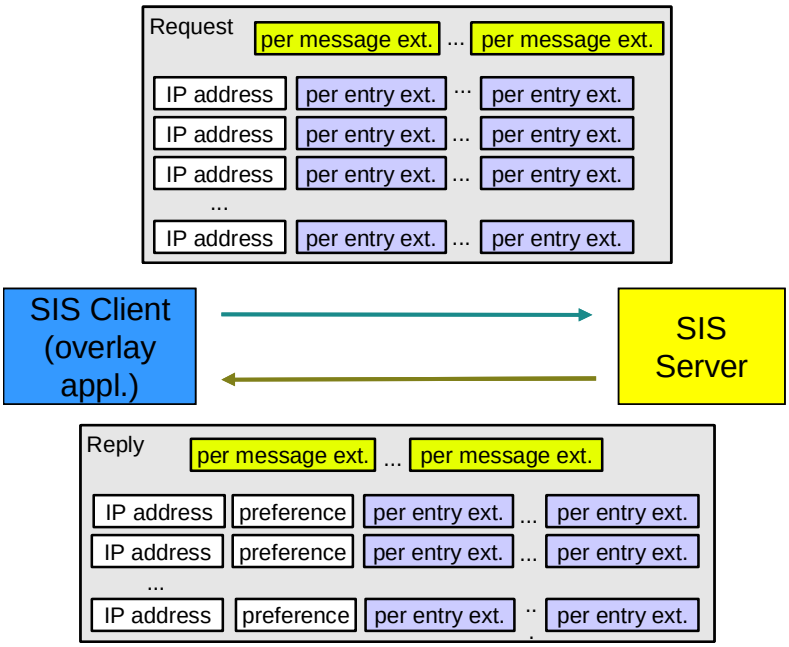

Figure 8-10 – SIS protocol

In the basic interaction, the SIS Client in the overlay application sends a request containing a list of IP addresses to the SIS Server. The SIS Server calculates a preference value to each IP address in the request and sends a reply containing the list of IP addresses and preference values back to the client.

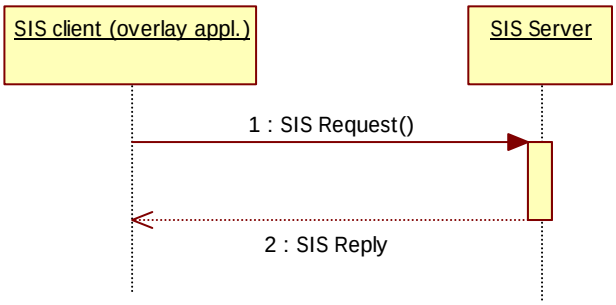

Figure 8-11 – SIS protocol interaction

Peer addresses sent to the SIS Server must fully identify the peer on the underlay network. They can be IPv4 or IPv6 addresses.

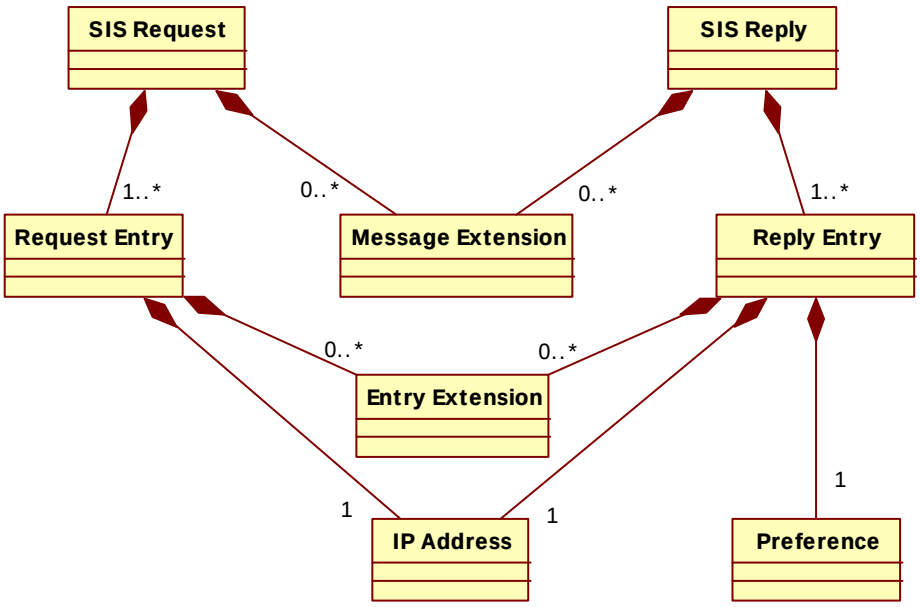

Figure 8-12 – SIS protocol messages

Both the request from SIS Client to the server and the reply from SIS Server to the client are extensible with additional information. This allows for the adaptation of the preference information service to additional requirements, even application specific. The extensions can be added to the request and reply messages in predefined places. All extensions to the basic preference information service are optional. Neither SIS Server nor client need to be able to interpret the extensions and do not have to process them. There are two possible types of extension entries in requests and replies as shown in Figure 8-12.

A per-message extension is an extension field containing an optional parameter that is valid for the entire message. Possible per-message extensions can be for example the type of application requesting the preference information or an indication to request additional application specific information to be considered during the processing of the request. A message can contain an arbitrary number of per-message extensions.

A per-entry extension is associated with a specific host address entry in the request or reply message. Per-entry extensions can be used to add additional information to the IP address entry as for example a port number or an application dependent peer ID of a peer-to-peer application.

A possible extension to the basic preference information is the addition of bandwidth information to the reply. A SIS request can contain a per-message extension field indicating the request for additional bandwidth information to the IP addresses listed in the request. The SIS Server operated by the ISP can add the bandwidth according to customers' network access subscription for the requested addresses. If a host address is not within the range of the ISP's customers, and therefore not in the SIS Server's network domain, a transitive request to a SIS Server in a neighboring network domain can be started, as described in Section 8.3.3, to request further bandwidth information. The bandwidth information can then be used by an application to request resources from hosts

that not only have high preference but also indicate high potential network access bandwidth.

In conjunction with a differentiating traffic shaping mechanism that allows high bandwidth for connections to intra-domain addresses and limits the bandwidth of connections to addresses in remote domain networks, a binary intra-domain/inter-domain metric can be added to the preference information service, to allow applications to distinguish IP addresses to be either within the own network domain or outside. With this information an overlay node can systematically choose connections for which the bandwidth is not limited through traffic shaping. By sending the request message with a special per-message extension entry, the SIS Client can indicate to the SIS Server to add the intra-domain/inter-domain metric as per-entry extensions to the reply.

Another possible per-message extension can be a description of the network connection QoS an application desires. Possible requirements of applications may be high bandwidth, constant bandwidth, low latency or low jitter. The SIS Server supporting this extension field can take the indicated requirements into consideration during the calculation of the preference values for the IP address list.

It is also possible to cluster peers according to their locality, keeping a map of peer distribution and adapting connection preference information accordingly. In this case, the SIS Server would return a Cluster ID in the reply. A Cluster ID that is assigned to different peers can be added to SIS's preference information service through a per-entry extension.

8.3.1.1 Errors

There are two types of sources for errors during the invocation of the preference information service: unsupported request source and invalid request.

If the SIS Server receives an incoming request from a client it is nor responsible for, the server can return the unsupported request source error. However, the server should limit the number of such error messages sent back and unsupported sources should be filtered additionally in order to avoid DoS attacks. A client receiving this type of error message should discover and contact the correct SIS Server.

The invalid request error is returned by the SIS Server if the preference information service request can not be processed due to errors in the request message. Incorrectly encoded request messages for example can be a reason for this error. The server does not return the invalid request error if the request contains extensions that are not understood.

If the SIS Server has no information available about one or more addresses in the request, it must simply return a special preference value to inform the SIS Client of such situation.

8.3.2 Admin Interface – SIS Server

The admin interface will be used by the network administrator to configure the following issues:

1. SIS internal parameters, such as the QoS classes of services, the maximum number of flows that can be prioritized or how locality must be enforced. This will be configured using a management graphical interface, where the network administrator can review the current configuration parameters and can, therefore,

modify them. In order to design this interface, one tab per Module must be provided.

2. Service Provider agreements. This interface will be used by the administrator to enforce specific policies associated to one service. The policies will include QoS level, charging parameters, locality, etc.

The graphical interface to configure the service provider agreements will allow the network administrator to upload a file (*e.g.*, in XML format) with the description of the application (how to identify the application flows in order to, *e.g.,* enforce a specific QoS policy) and with information about accounting.

8.3.3 SIS Server – SIS Server

There might be situations in which a SIS Server does not have enough information to calculate the preference value or extended information for a certain host. This can happen, for example, when querying for available bandwidth on the destination host. In these cases, the SIS Server can be configured to request further preference information from another SIS Server.

If network traffic crosses the borders of a local network domain, it is routed through one of possibly several neighboring networks. If a SIS Server exists in a neighboring network, a transitive request association can be configured between the two servers, if both servers support and offer a server-to-server preference information service based on the same metric.

Authentication and authorization mechanisms must be in place to prevent abuse of the system. Cryptography may also be used to protect confidentiality of the data.

A similar interface as described in Section 8.3.1 is foreseen, but only the IP addresses that belong to the target domain will be included in both the request and the response.

9 Summary and Conclusions

This deliverable presents a scenario for the SmoothIT architecture, describes requirements, discusses the design space for economic traffic management, and specifies the initial SmoothIT architecture. In the design space discussion, four main approaches for economic traffic management are addressed in detail and their advantages and disadvantages are analyzed.

Although the “honey pot” approach of the design space shows high optimization potential, its main deficiencies are its limited deployment potential due to legal issues and the complexity of a possible content analyzer for illegal contents. The “optimal anarchy” approach still needs further investigation and analysis in order to evaluate its complexity and scalability. Since this approach requires modifications in all or at least most of the routers in a network, its deployment is also challenging. The “block party” approach focuses on cooperative inter-ISP (Internet Service Provider) communication, where security and trust issues play an important role. The most promising approach seems to be the “control freak” approach. It is independent of overlay applications, it is easier to be deployed in an operational network than the other approaches, and it shows a high optimization potential. Therefore, the initial architecture design focuses on the “control freak” and “block party” approaches but it allows the support of all proposed approaches.

The initial SmoothIT architecture specifies main components, their functionality, and their interactions. The architecture introduces the SmoothIT Information Service (SIS) which is a new service deployed in the network of an operator and supports the economic traffic management of overlay application traffic. The SIS interacts with the overlay application and it conveys information between overlay application and network infrastructure, *e.g.*, it can assist peer selection in overlay applications by assigning a preference value to each possible peer. The operation of the SIS and the preference value calculation are influenced by overlay application requirements, network conditions, and ISP policies. Besides the SIS, the architecture defines the following components: Metering, QoS Manager, Security, and Configuration Database. The metering component collects information from the network relevant for the SIS, *e.g.*, network status, performance metrics, and network usage metrics. To extract information about inter-ISP relations the metering component includes the BGP (Border Gateway Protocol) information module which collects BGP routing information. The performance meter module collects network performance related metrics, while the usage meter module is responsible for collecting network usage on a user or application basis. The QoS (Quality-of-Service) support of the SmoothIT architecture takes the current Next Generation Networks (NGN) architecture as its basis in order to be standards-compliant. The QoS manager is responsible for managing available resources and enforcing QoS policies in the network. Different traffic classes are defined for traffic with different characteristics, like real-time, interactive, and best effort traffic. The configuration database stores ISP policies, *e.g.*, business relations to other ISPs and preferred networks. Based on the information collected by the metering component and the information stored by the configuration database, the SIS can make informed decisions for traffic management purposes.

This deliverable also specifies the initial version of a protocol for the communication between overlay application and the SIS server. The protocol defines a stateless request-reply interaction, where peers in the overlay application can request different kind of information from the SIS server. As a basic service the SIS server provides a preference

value to each peer indicated in the request. The messages of the protocol have a flexible structure, enabling the extension of messages with additional new attributes.

Based on the results of this deliverable, future work in Task T3.2 includes the further evaluation of ETM (Economic Traffic Management) mechanisms with respect to their scalability, imposed overhead, and applicability to different type of overlay applications, like file sharing and video streaming. Additionally, the specification of the final architecture and of all interactions between its components determines future work.

10 References

[AAF08] V. Aggarwal, O. Akonjang, A. Feldmann. *Improving User and ISP Experience through ISP-aided P2P Locality*. In Proceedings of 11th IEEE Global Internet Symposium 2008 (GI'08), (Location: Phoenix, AZ, USA), IEEE Computer Society, Washington, DC, USA, April 2008.

[AFK07] V. Aggarwal, A. Feldmann, R. Karrer. *An Internet Coordinate system to enable collaboration between ISPs and P2P systems*. In Proceedings of the 11th International ICIN Conference, (Location: Bordeaux, France), October 2007.

[AFS07] V. Aggarwal, A. Feldmann, C. Scheideler. *Can ISPs and P2P systems co-operate for improved performance?*. ACM SIGCOMM Computer Communications Review (CCR), 37(3):29-40, July 2007.

[BCC+06] R. Bindal, P. Cao, W. Chan, J. Medval, G. Suwala, T. Bates, A. Zhang. *Improving Traffic Locality in BitTorrent via Biased Neighbor Selection*, Proc. IEEE ICDCS'06, 2006.

[BT] BitTorrent. http://www.bittorrent.com. Accessed in August 2008

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss: *An Architecture for Differentiated Service*; IETF RFC 2475, December 1998.

[BCS94] R. Braden, D. Clark and S. Shenker: *Integrated Services in the Internet Architecture: an Overview*; IETF RFC 1633, June 1994.

[B03] S. Brecht: *Investigation of application layer routing*, Master's thesis, University of Federal Armed Forces Munich, Germany, Supervisors: P. Racz, B. Stiller, December 2003.

[CLG+03] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko: *Diameter Base Protocol*; IETF RFC 3588, September 2003.

[CS05] Cisco Systems Inc: *DiffServ-The Scalable End-to-End Quality of Service Model*, White Paper, Cisco Systems, August 2005.

[CE08] M. A. Callejo-Rodríguez, J. Enríquez-Gabeiras: *Bridging the Standardization gap to provide QoS in current NGN Architectures*; IEEE Communications Magazine, October 2008.

[CS08] Cisco Systems Inc: *Cisco Visual Networking Index – Forecast and Methodology, 2007-2012*. June 2008. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf. Accessed in August 2008

[C03] B. Cohen, *Incentives build robustness in BitTorrent*, in Workshop on Economics of Peer-to-Peer Systems, May 2003

[C08] B. Cohen, *The BitTorrent protocol specification*, 2008. http://www.bittorrent.org/beps/bep_0003.html, accessed August 2008.

[D1.1] SmoothIT Project: *Requirements and Application Classes and Traffic Characteristics (Initial Version)*; Deliverable D1.1, July 2008.

[D2.1] SmoothIT Project: *Self-Organization Mechanisms for Economic Traffic Management*; Deliverable 2.1, July 2008.

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

[ECR+03] T.S. Eugene, Y.-H. Chu, S.G. Rao, K. Sripanidkulchai, H. Zhang: *Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems*, INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE , Vol. 3, pp. 2199-2209 vol.3, 30 March-3 April 2003.

[GHN+03] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen: *P2P, the Gorilla in the Cable*. Proceedings of National Cable & Telecommunications Association (NCTA), pp 8-11, June 2003.

[HB96] J. Hawkinson and T. Bates: *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. IETF RFC 1930, March 1996.

[OSP+08] S. Oechsner, S. Soursos, I. Papafili, T. Hossfeld, G. D. Stamoulis, B. Stiller, M. Angeles Callejo, D. Staehle: *A framework of economic traffic management employing self-organization overlay mechanisms*, IWSOS 2008, Vienna, Austria, 2008.

[PPLive] PPLive. http://www.pplive.com/. Accessed in August 2008

[Skype] Skype. http://www.skype.com/. Accessed in August 2008

[WLR+04] A. Wierzbicki, N. Leibowitz, M. Ripean, R. Wozniak. *Cache Replacement Policies Revisited: The Case of P2P Traffic*. European Transactions on Telecommunications, November 2004, Volume 15 Issue 6, Pages 559 – 569. Wiley.

[XKS+07] H. Xie, A. Krishnamurthy, A. Silberschatz, Y. R. Yang: *P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers*, 2007. http://www.dcia.info/documents/P4P_Overview.pdf, accessed August 2008.

[XYK+08] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz. *P4P: Provider Portal for Applications*. ACM SIGCOMM 2008.

[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.

11 Abbreviations

AAA	Authentication, Authorization, Accounting
AS	Autonomous System
BE	Best Effort
BGP	Border Gateway Protocol
CAC	Call Admission Control
COPS	Common Open Policy Service
CoS	Class-of-Service
DB	Data Base
DHCP	Dynamic Host Configuration Protocol
DHT	Distributed Hash Table
DNS	Domain Name System
DoS	Denial-of-Service
DPI	Deep Packet Inspection
DSL	Digital Subscriber Loop
DSLAM	Digital Subscriber Line Access Multiplexer
EGP	Exterior Gateway Protocol
ETM	Economic Traffic Management
ETSI	European Institute for Telecommunication Standardization
FQDN	Fully Qualified Domain Name
HTTP	Hyper-text Transfer Protocol
ID	Identifier
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISP	Internet Service Provider
ITU-T	International Telecommunications Union – Telecommunications Sector
MED	Multi Exit Discriminator
NAT	Network Address Translation
NGN	Next Generation Networks
NOC	Network Operations Center
NTIDS	Network Topology Information Desk Service
NMS	Network Management System

NOC	Network Operation Center
OAM	Operation and Management
OID	Object Identifier
P2P	Peer-to-Peer
P4P	Proactive Network Provider Participation for P2P
PKI	Public Key Infrastructure
PoP	Point of Presence
QoE	Quality of Experience
QoS	Quality of Service
RFC	Request for Comments
SIS	SmoothIT Information Service
SmoothIT	Simple Economic Management Approaches of Overlay Traffic in Heterogeneous Internet Topologies
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
STREP	Specific Targeted Research Project
TCP	Transmission Control Protocol
ToS	Type-of-Service
UML	Unified Modeling Language
VoIP	Voice-over-IP
xDSL	x-type Digital Subscriber Line

12 Acknowledgements

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

13 Appendix A – Use Cases

This section presents the typical use cases that can occur in the SmoothIT system. Possible actors are identified to be Downloading and Uploading peers. The rest of the system, including SIS, ETM, NMS, Billing and Subscription management is treated as a "black box" and is referred to as SIS. The description of these use cases refers to three settings, which are either specified by the peer user or obtained in some other way.

- The SIS node address is the URI of the SmoothIT Information System.
- Price/QoS ratio controls the trade of between the quality of service (i.e. higher throughput and lower delay) and the ISP charges the rate customer.
- "Harmony" ratio allows peers to balance the influence of SIS and Overlay information and to avoid the problem when swarm becomes overly localized rendering it difficult to distant peers to join. The value of 0 corresponds to relying completely on SIS information, and 1 – overlay information.

13.1 Connect to SIS Node

Any operation with SIS node relies on the fact that the peer knows its IP address. The name is obtained from the settings, resolved via DNS and the session is initialized.

Actors: Peer

Pre-conditions: Void

Steps:

1. Read SIS node address from the settings
2. Resolve SIS IP via DNS
3. Initialize the session

Alternative paths: None

13.2 Register in Swarm

When peers join a swarm it is essential that it will notify the SIS node about this event and by providing an ID that would globally and uniquely identify that specific swarm. During the registration process the peer informs SIS about the Price/QoS ratio it requires to perform all swarm-related operations during the current session.

Actors: Peer

Pre-conditions: Void

Steps:

1. <<include>> Connect to SIS node
2. Calculate swarm id
3. Send SIS node message about participating in the swarm with given ID along with desired *Price/QoS* ratio

Alternative paths: None

13.3 Download Next Chunk

When a peer has an available download slot it starts the procedure of downloading the next chunk of data. This involves making decisions on which chunk to download now and from which peer.

Actors: Downloader

Pre-condition: Other peers exist, which seed chunks that the current downloader does not have.

Steps:

1. <<include>> Identify highest priority chunk
2. Find which peers have the required chunk
3. <<include>> Populate candidate peer list
4. Enter download queue
5. Notify SIS node about intention to download a chunk

Alternative paths: If step 5 fails, bypass it.

13.4 Populate Candidate Peer List

When a peer is trying to download a chunk of a file, it needs to identify one or more potential peers to connect to. In the SIS enabled this is done by calculating a weighted average of overlay and SIS ratings of potential peers and choosing the one with the highest average rating. The weights of the rating system are calculated based on the *harmony* ratio.

Actors: Downloader

Pre-condition: Available peer list is not empty.

Steps:

1. Populate Candidate Peer List by applying overlay network constraints to available peer list.
2. <<include>> Obtain SIS rating for every candidate peer
3. Apply rating constraint assessment to the candidate peer list

Alternative paths: If Step 2 fails for any node, assign that node with zero rating, being the lowest and if SIS node is not available at all then all nodes will be on equal terms.

13.5 Obtain SIS Rating

The most frequently occurring use case between peer and the SIS node is the obtainment of the SIS rating for the given peers. The downloading peer requires this in order to perform optimal choice of candidate peers to download from. The uploading peer needs this information in order to maximize its reward.

Actors: Peer

Pre-condition: Non-empty peer list, Connection to SIS node is established.

Steps:

1. Send to SIS a request for rating with following parameters: Swarm ID, Traffic Direction (Up/Down), Peer List
2. Receive hash of peer IDs with respective rating expressed in absolute values

Alternative paths: If any step fails assume all peers have the same zero rating.

13.6 Respond to a Download Request

When a peer receives a request from another peer for a given chunk of data it needs to decide how to prioritize in the queue this request. This decision is normally taken based on some criteria of the overlay network, such as the peer's community rating. In the SIS-enabled network the uploading peer is also encouraged to check SIS-rating of the requesting peer in order to maximize its ETM reward.

Trigger: Download request

Actors: Uploader

Pre-Conditions: Void

Steps:

1. <<include>> Request upload rating for given IP Address / Swarm ID from SIS node.
2. Calculate requesting peer's *average* rating based on its overlay and *SIS* ratings and the *harmony* ratio.
3. Allocate the requesting peer a place in the download queue respective to the average rating value.

Alternative paths: If step 1 fails, assume the peer has zero SIS rating.

13.7 Identify Next Chunk to Download

When peer makes decision about next chunk to be downloaded it uses different metrics suggested by the overlay network, user settings and the nature of the content. In SIS-enabled network it can optionally query the SIS node to get the list of possible chunks sorted in such a way that promotes traffic-locality and maintain peers Price/QoS parameter.

Actors: Downloader

Pre-Conditions: At least one more chunk left to download.

Steps:

1. Identify potential following chunks to be downloaded based on non-SIS criteria.
2. Send the swarm id and the list of chunks to the SIS node
3. Receive ordered list of chunks with the most recommended ones being first

Alternative paths: If step 2 or 3 fails, bypass.

13.8 Notify SIS Node about Completed Upload

Once the peer finishes uploading a chunk it informs the SIS node. This is necessary for the reward to be debited.

Actors: Uploader

Pre-Conditions: Connection to SIS node is established.

Trigger: Upload completed

Steps:

1. Send "*Upload Complete*" message with swarm ID and chunk ID.

13.9 Notify SIS Node about Completed Download

Actors: Downloader

Pre-Conditions: Connection to SIS node is established.

Once the peer finishes downloading a chunk it informs the SIS node. This is necessary to confirm uploaders message.

Trigger: Download completed

Steps:

1. Send "download complete" message with swarm ID and chunk ID.

13.10 Notify SIS node about intention to download a chunk

Actors: Downloader

Pre-Conditions: Connection to SIS node is established.

Once the peer is allocated a slot in the uploaders queue, it notifies the SIS node. This allows the SIS node to better optimize future requests.

Steps:

1. Send "*Expecting Download*" message with swarm ID and chunk ID.

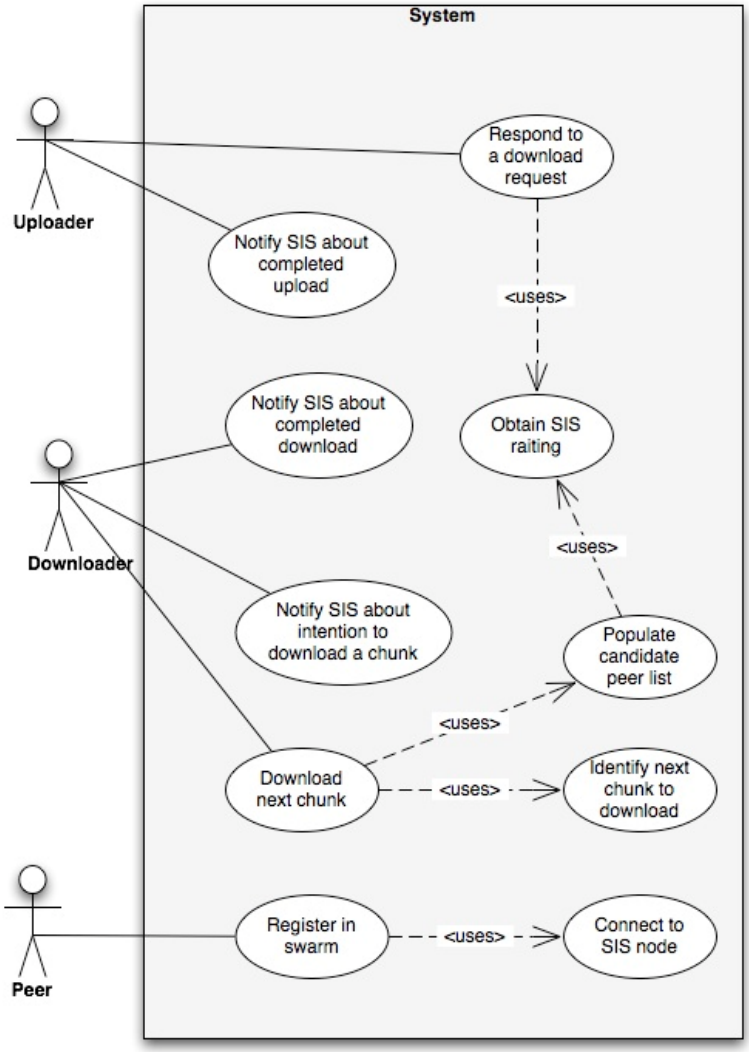

Figure 13.1 – Use cases