TAPAS

IST-2001-34069

Trusted and QoS-Aware Provision of Application Services

TAPAS
Application Hosting and Networking Req. Document, D1

Report Version: Deliverable DI

Report Delivery Date:

Classification: Public Circulation

Contract Start Date: 1 April 2002 **Duration:** 36m

Project Co-ordinator: Newcastle University

Partners: Adesso, Dortmund – Germany; University College London – UK; University of Bologna – Italy; University of Cambridge – UK

information society technologies

Project funded by the European Community under the "Information Society Technology" Programme (1998-2002)

TAPAS Deliverable D1

W.Beckman [(2)], J. Crowcroft [(1)], P. Gevros [(1)], M. Oleneva [(2)]

(1) Computer Laboratory, University of Cambridge (editors), (2) Adesso AG

2002-10-18

1 Overview

Application service provision is a relatively new way of selling Information Technology (IT) products and services. According to this model the client gets to use applications without paying for licenses and without incurring the configuration and maintenance costs. All that is required on the client site is a relatively low-end computing platform, which can connect over the network to an Application Service Provider (ASP) who is responsible for configuring, running and maintaining the application and the systems it runs on.

The ASP is the focal point both in the services architecture envisaged in the TAPAS Project and in the e-business scene in general. The ASP is interacting with end-users (humans and/or organisations), which outsource a wide variety of applications and the ASP provides “hosting” for these applications. However, an ASP also has to assume the role of the client in order to get access to network services, which ultimately enable distributed applications.

In this capacity the ASP interacts with Internet Service Providers (ISPs) and negotiates access to a wide range of Internet services, ranging from Internet connectivity to anything that might be required from the network for hosting a particular distributed application (e.g. name resolution, virtual private networks, multicast, network security, storage space, web site hosting, web caching).

Thus both end-users and ASPs rely on (other) service providers for services that are critical to their business and this requires a high level of trust between the client and the provider.

It is a common industry practice for the providers to guarantee specific levels of service and availability for their services in order to win the trust of their clients.

Service guarantees are provided in the form of a legally binding contract called a Service Level Agreement (SLA), which defines the services to be provided and the metrics, which determine the successful delivery of these services.

However the definition of such contracts is a particularly complex task. According to the ASP Industry Consortium [ASP] there are three elements that should be considered in an *end-to-end* SLA; the *application*, the *system* and the *network* part. These elements involve specifications of diverse factors, ranging from latency and packet loss to helpdesk availability and designated account managers with defined responsibilities to the customer.

Usually the ASPs do not own the networks that they use but they own the systems (computers, operating systems, databases) and the applications. However the performance of the network largely determines the quality of application provisioning. Therefore the network SLAs between ASP and the ISP(s) are of particular interest to the TAPAS Project.

In the first part of the document we discuss the application hosting requirements, which are relevant to the TAPAS project from the viewpoint of the ASP. Firstly we inspect the problem domain by highlighting the assumptions concerning the development of the ASP market which build the base for requirement discussion. An abstract ASP scenario then helps us to identify the major stakeholders and their interactions. The relationships between stakeholders are governed by SLAs, from which we extract the basic guarantees and commitments. Moreover we discuss the technology standards relevant for TAPAS and outline the technical and commercial criteria to be used in the assessment of the project's success. Additionally we present a business-to-business (B2B) auctioning platform as an application to be hosted by the ASP.

The questions to be answered in the second part of this deliverable are concerned with the network (IP) services that can be expected to be available to the TAPAS project during the course of the partners' work. In practice, this will inform us what is pragmatic, but also, in terms of long-term research, what further work may be required by the partners and in subsequent work, to address shortcomings in the lower layers.

Conceptually the ASP and the ISP have one thing in common; they are both service providers and in order to interact with their clients they need to offer a certain SLA. These SLAs rely on the same conceptual framework, there is a business and a technical aspect. The specifics are obviously different and this is the reason why we discuss the ASP and the network SLAs separately in this document.

2 Application Hosting Requirements

2.1 Problem domain

2.1.1 Assumptions

As electronic trade and communication become more and more important in today's business world, the complexity of organisational interactions increases rapidly. We are confronted with rising demands on the security and responsiveness of application providing.. As a study of electronic market places [DIS01] shows, B2B marketplaces are set to become the second most important corporate coordination mechanism after emails within the next five years. Specifically, services such as authentication, reporting and monitoring coupled with the key aspect of availability will become the core focus of B2B applications.

Generally speaking, organizations want to increase their degree of specialization in a way that is similar to other engineering disciplines (e.g. car manufacture) to increase competitiveness. The higher the specialization is, the more organizations are involved in executing an application. As organizations become more specialized, the total number of organizations involved in executing a process chain increases. In an ASP scenario we have an e-business application, an Application Service Provider (ASP), end users and other providers (s. Appendix A, chapter1.1). These organizations might be competing with each other (e.g. buyers in auctions or competing service providers).Generally, organizations establish a contractual relationship prior to engagement. The higher the Quality of Service (QoS) and Trust requirements are, the better defined the contractual relationship needs to be. The quality of contractual relationship depends also on the frequency of the interaction between distributed organizations. Increased frequency of interactions between distributed organizations makes it necessary to better define Service Level Agreements (SLA).

Both quantitative and qualitative aspects of QoS and Trust of services have to be defined in SLAs. This leads to the further differentiation of SLAs. Hence there will be standard and dedicated SLAs reflecting different levels of trust and QoS. Standard SLAs will be used for commodity services, governing the availability and problem management. Service providers will only negotiate dedicated SLAs for high-value service users, which need special trust clauses or advanced requirements on performance or timeliness. Service users will use SLAs as a selection criterion (alongside price and functionality) when evaluating competing service providers.

In current industrial practice the procedure of establishing and monitoring contractual relationships is inconsistent and expensive. Beyond that, the ASP model is momentarily handicapped because of a lack of trust: although economic considerations speak for outsourcing, concerns on industrial espionage obstruct its adoption.

QoS-aware application providing is limited by the fact that current SLAs mostly exist as part of a contract in paper form. The contract will then be manually transformed into an appropriate set-up of hardware, software and user services. This transformation does not only provoke transformation errors, but is also costly and time consuming. Hence organisations are interested in modelling and monitoring tools, which simplify and optimize the contract handling from the very first negotiations to the deployment and monitoring of an application. A tool-supported approach will thus leverage the hosting workflow and increase trust in the ASP model.

During contractual negotiations, an ASP will try to offer services at reasonable prices, while the client's interest is to minimize the price or maximise the quality of the included services. In today's ASP market, ASPs and clients will in most cases negotiate single SLA objectives rather than combinations of various objectives. For example, a bandwidth will probably be discussed in isolation from a throughput. Therefore it is often difficult for ASPs and clients to judge the results that changes in combinations of objectives may lead to. Negotiations are further complicated by limited timeframes, because the "time to market" is quite crucial for Internet businesses. Hence clients can only analyse a rather small set of SLAs offered by different ASPs during contract negotiations. However, formal SLA modelling is currently not being practiced in the ASP industry, because appropriate formal modelling languages do not exist. The most demanding aspect of today's ASP is the performance of the hosted application, if the business is intended to grow alongside the security and availability aspects. Modelling techniques and tools, which cover these aspects and allow to predict consequences of specifications will be beneficial for ASPs and for their clients.

Generally speaking, in current industrial practice the clauses concerning reporting and monitoring of the ASP's performance are not sufficient. From the technical point of view, the reports are not transparent to the client while there are few graphical tools for the SLA controlling, especially for distributed hosting. In most cases, an ASP produces reports and statistics themselves, based on their available log files. It is evident that this is the lowest level of security for the customer who has no independent access to the hosting environment. An ASP could thereby easily hide unwanted results. A first generation of monitoring tools, however, does exist. The support tool ILOG JRules, for example, provides an approach for QoS/SLA controlling [IL].

Nowadays, coupling between SLA fulfilment and monetary sanctions exist in ASP contracts, but is based on simple calculations and is commonly not sufficient. The dependencies between objectives of an SLA on the one hand and different SLAs on the other hand are often not taken into consideration when penalties are being defined. First of all, clients will profit from automatic computation of penalty fees. Computations may be carried out by an execution environment, which is governed by a Service Level Specification (SLS). It would be helpful to make the penalty chart transparent and visible for all parties in the value chain.

Applications which are composed of services of distributed parties, will not be successful without explicit definitions of the QoS and trust requirements in a contractual relationship. The existing technology is beginning to address this problem but is still inadequate in its support for QoS and Trust. As e-business applications become increasingly distributed and automated, the fraud problems become worse: when a critical infrastructure (e.g. the power-grid) is involved, the consequences of a criminal activity may change from simple fraud to a significant national emergency.

The TAPAS innovation objective is to enhance SLAs to make component containers QoS-aware and enrich them with trust specifications. For this purpose we will consider hosting requirements of multi-party applications that are characterised by many-to-many interactions.

2.1.2 Stakeholders in the ASP Model

In order to understand Service Level Agreements between the participants in an ASP scenario we now focus on the parties involved in ASP and the relationships between them. The ASP model described below is an abstraction of typical industrial ASP situations. For readers unfamiliar with ASP, we provide a discussion of a realistic scenario in Appendix A, chapter 1. Figure below shows the stakeholders and their SLA-based relationships in an abstract ASP model.

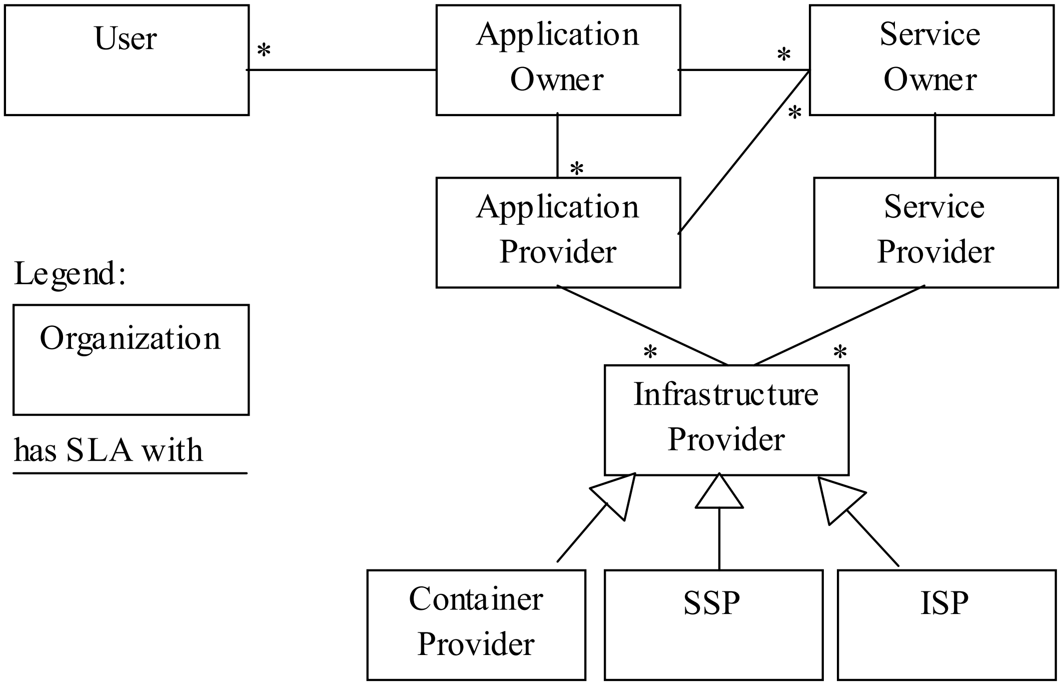


Figure 1

Users in the ASP model are human users who might not have any technical experience like the sellers or buyers in our auction scenario. Such users use an application through a computer interface, which is adapted to their needs. Current industrial e-Business solutions tend to use Internet browsers, sometimes enriched with Applets, JavaScript or Visual Basic. Each user has a relation to the application owner. This relation is typically governed by terms and conditions, some relationships might even be defined by formal contracts, especially, if the business relation is longer-term or more intensive.

An **application owner** is typically a company on whose behalf the application is provided. In most cases, applications are not available as standard software, hence the application owner typically is the one who ordered the construction of the application software as well. In a lot of cases the application owner is not the one running the application, but the one owning the business. In our market place scenario this would be the market place company. This results in a lot of relationships with other stakeholders:

- As has already been pointed out, users have contractual relationships with application owners with varying degrees of formality. However, in the end the users are the ones who pay a fee or some part of their benefit for using the application.
- Service owners are business partners of application owners. The relationship is ruled by contracts.
- The application provider is running and maintaining the technical aspects of the_application. An application owner may of course have multiple application providers, simply for e.g. different portals for different user groups or countries, though the underlying application is the same.

An **application provider** is the ASP, this stakeholder runs and maintains the application. From our industrial experience we can say that service owners sometimes have contractual relationships to the application provider instead of having contracts with the application owner. This is especially the case if the application owner only provides the business concept or if the application provider only benefits from special conditions defined by the service owner. The application provider however, is usually not able to supply all the services on his own. Therefore several infrastructure providers have relations to the application provider. Such infrastructure providers are in our market place scenario the ISP (Internet Service Provider) or the SSP (Storage Service Provider). In our market place scenario, these infrastructure providers are represented by the ISP or the SSP.

The **service owner** is a company providing a business service (such as credit rating as exemplified in our scenario). Between service owner and application owner or providers there are contracts in form of SLAs that govern the relationships. An e-business application consists of a combination of services, which are owned by different service owners. This combination is created by an application owner or an application provider.

In our model the service owner is separated from the service provider, as the providing may require technical skills and may therefore in turn be implemented by another application provider, who is acting as a service provider there.

A **service provider** is an organization that provides services that are used through well-defined interfaces to build more complex services or applications. As a consequence, service providers do not have to own the services like credit rating, they just offer the technical access. CSP (Certification Service Provider) exemplifies a service provider in our auction scenario. In this role CSP is responsible for the electronic signature validation according to the EU-Directive [EU00].

An **infrastructure provider** is an organization that provides the technical infrastructure to facilitate the hosting of services and distributed interaction between service providers (e.g. Internet Service Providers, Storage Service Providers, Component Containers Providers). ASP or other Service Providers have the right to choose under competing infrastructure providers depending on their offers (e.g. flat-rate, conditions).

2.1.3 Content of SLAs

The SLAs listed in this chapter reflect the best practices in industrial ASP contracts. This overview will not only motivate the discussion of SLA guarantees and commitments (see chapter 2.3), but shall also give a background for further research on Service Level Specifications.

It should be noted that SLAs are complemented by outsourcing or partnering contracts that cover corrective actions, penalty clauses, non-performance clauses, amount of deviation, modification procedure, reporting policies, termination criteria, intellectual property clauses, dispute resolution procedures, as well as others.

The Application Service Provider Industry Consortium [ASP] divides SLAs into four categories. The items for these SLA types are here taken from the adesso standard hosting contract [AD]:

- **Network SLA**. This covers the network connection between the customer and the ASP. Here, a Network Service Provider agrees upon a suitable service level agreement for delivery of IP Services to a business customer.

SLA item	Quantification
Bandwidth	bit/sec
Latency	ms
Jitter	%
Traffic monitoring	yes/no

- **Hosting SLA**. This covers the hosting services provided to the ASP. The ASP delivery model often mandates that hardware be hosted or co-located with a third party.

SLA item	Quantification
Service place	city/state
Duration of service	timeframe
SLA fulfilment monitoring	yes/no

- **Application SLA**. This addresses the measurement of application performance. The ASP defines the domain of its responsibility, service classes to be offered, parameters of performance, etc. This formulates the calculation of performance and penalties to be imposed for failure to meet agreed upon service levels.

Quality of Service

SLA item	Quantification
Service time	timeframe
Service rate	number/timeframe

Mean time between 2 failures	h
Recovery time	sec
Maintenance window	timeframe
Availability of all Components Connected to the Network	%
Availability of Applications on the Network (Access)	%
Availability of Clients	%
Availability of Servers	%
Availability for varying time regimes	% (e.g. evenings, overnight, weekends, public holidays)
Percentage of Transactions completed within defined Performance Levels	%
Application response Time during peak Periods	ms
Peak periods details	timeframe
Median Application response Time	ms
Network Round Trip Time (Delay)	ms
Server delay	ms
Client delay	ms
Database I/O performance	ms
External memory access time	ms
CPU usage	%
Data package loss	%

Security

SLA item	Quantification
Encrypted database storage	yes/no
Data can be downloaded automatically from Applications	yes/no
Certificated security form	yes/no
Scanning the content of incoming mail	yes/no

Data backup

SLA item	Quantification
Encrypted backups	yes/no
Separate backups for each customer provided	yes/no
Data backup interval	incremental every x min/complete every hour
Data backup types	file types

Archiving period	timeframe
Accessibility of the back-up copies for the customer	yes/no

Customer care. The terms "customer care" and "help desk" refer to a point of contact for customers who seek point-solution assistance. For an ASP, customer care is emerging as the means by which customers obtain answers to inquiries [ASP]

SLA item	Quantification
Support hours	timeframe

2.2 Standards

For TAPAS to have any impact it will be important to rely on, comply with and extend current industry standards. TAPAS aims to improve different areas of ASP, such as middleware architectures, message exchanges and deployment, modelling and reasoning about models, maintenance and support, external monitoring and, last but not least networking.

Regarding middleware architectures the software industry currently focuses on component based development. The mostly used framework for object-oriented technology is J2EE [J2E] (Java 2 Enterprise Edition), including established component model EJB (Enterprise JavaBeans). Furthermore, the still widely spread CORBA standard offers an own component framework called CORBA Component Model [CCM].

- **R1.1**[1]: Usage of widely spread component-based middleware technology such as EJB or CCM.

The most popular data exchange format XML (eXtensible Markup Language) is qualified for SLS (Service Level Specifications) as well as for deployment descriptors. TAPAS could use EbXML (Electronic Business using eXtensible Markup Language) for business message exchange and RosettaNet as an open e-Business process standard.

- **R1.2**: Representation of structured data, e.g. SLS, in XML and, if applicable, utilization of derived standards such as EbXML.

OMG provides UML (Unified Modelling Language) with Profiles as well as XMI (XML-based Metadata Interchange) and Meta-Object Facility (MOF) for meta-modelling and metadata repositories [OMG]. TAPAS should take advantage of OMG standards for SLS modelling.

- **R1.3**: Utilization of standard modelling languages such as UML for SLA modelling

In TAPAS there will be several components that manage specific parts of Quality of Service. It is beneficial to integrate such products from different sources to an end-to-end solution. The Operations Support Systems through Java (OSS/J) Initiative from SUN worked out a QoS API [OSS] that aims to reduce the effort caused by such integrations. If this API is implemented by the mentioned components, the integration could be done with minimal effort. It is divided into three areas: Performance or Usage Data Monitoring, Threshold Data Monitoring and Fault Data Monitoring. The application of this API would be very helpful for supervising the fulfilment of SLAs in

[1] Notation: **RX.Y.Z**: requirement, *where* **X** *– number of the requirement group,* **Y** *– number of the requirement in the chapter,* **Z** *– aspect number of the requirement* **Y** *(optional)*

distributed environments. However, as TAPAS aims to develop new solutions, an existing standard might turn out to be too restrictive for innovative approaches.

- **R1.4**: Evaluation of applicability of the OSS/J API for TAPAS components

Web Services applying SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration Specifications) and WSDL (Web Service Description Language) could be useful as ASP monitoring and reporting services [W3C].

- **R1.5**: Exploitation of Service-Oriented Applications using Web Services where applicable.

The network IETF standards DiffServ (Differentiated Services), IntServ (Integrated Services), PGM (Pragmatic General Multicast) and Sigtran (Signalling Transport) could be utilized by TAPAS for the development of QoS-aware protocols. The narrow description of the network standards and its usage are presented in the chapter *Networking Requirements*.

- **R1.6**: Exploitation of current IETF network standards where applicable.

2.3 Guarantees and Commitments defined in SLAs

A service level agreement (SLA) defines the responsibilities of a service provider and the users of that service. It also identifies the services provided as well as the supported products, measurement criteria, reporting criteria and quality standard of the service.

Based on the exploitation of current ASP's SLAs, which we consider in chapter 2.1.3, we find that guarantees and requirements, which different stakeholders want to determine in an SLA are:

1. **Performance** of a quantitatively characterized service. The quantitative parameters are typically arrival rate, service size and characteristics (e.g. storage requirements of the service request):
   - Service time: The amount (average, maximum, minimum, distribution) of time that it takes between the request of the service and the completion of the service
   - Service rate: The number of service completions that are guaranteed to perform in a given period of time
   - Timeliness of service providing in distributed scenarios and/or by critical applications (e.g. auctions, stock market)
2. **Availability**
   - Of service provision: The duration during which a service or application is guaranteed to be available. Can be defined by a given period of time (e.g. Mon-Fri 9-5) or as a percentage (e.g. 99.5% of the time) or a combination of these.
   - Maintenance window. An explicit timeframe, in which the system is allowed to be maintained, and therefore won't be classified as down time.
3. **Reliability**: Mean-time between two failures of the service
4. **Maintainability**: The maximum time required to recover a failure of the service to its normal operation.

The line fall-outs and error messages are subdivided into error classes [AD]:

- Critical problems (service class 1: priority »very high«)
- Endangering problems (service class 2: priority »high«)
- Disturbances (service class 3: priority »medium«)
- Impairments (service class 4: priority »low«)

The processing time begins with a fault report by the client. The escalation level and recovery time are defined according to the error classes.

5. **Security**:
   - Privacy: prevention of eavesdropping of data in transit or on storage
   - Authorization and authentication of service invocation
   - Fairness: prevention of misuse of data or time advantage
6. **Monitoring** produces the service level statistics.

   For any ASP market to work, it is necessary to offer non-refutable SLA monitoring services. As soon as there are severe penalty clauses it will be important that the service level statistics are correct. Otherwise the contract controlling cannot be assured. When the length of the value chain increases it will be more important to have trustworthy statistics. For example ASP has to assure service time, depending on the underlying network characteristics, assuring by the ISP. The transparent monitoring for all participants is indispensable in this case.

   For some type of service, owners will not enter into an SLA unless there are trusted guarantees about the service level, e.g. bank accounting.
7. **Penalty** clauses determine
   - Functions for the monetary compensation of failures (money back guarantee), based on the one of the following criteria:
     - Monthly percentage fee rebates correlated to degree of under performance or non-availability
     - Specific fee reductions or rebates if service level is not met
     - Promotion to higher service level
   - Contract termination if the service level is not met.

   Penalties might be limited and not cover the loss of business or opportunities. Penalty clauses will have to be composed according to the composition of the service to allow the forwarding of compensation.

   From the service level, the parties should identify certain "key" service levels. It could be that the host CPU availability is a key service level in addition to certain response time and batch processing measures. Key service levels will be weighted by importance or severity so that they total 100 percent. Then if the outsourcer fails to achieve some of the key service levels, the percentage of key service levels missed for the month can be applied as a service level credit against a percentage of the invoice.

   In some cases, the parties may choose to identify not only a threshold level of acceptable performance for each service level, but also a level of "increased impact" if the performance is at an agreed level below the threshold service level. If the outsourcer's performance falls below the increased impact level, the percentage service credit may increase substantially. Another factor that may be included in calculating service credits is a "frequency factor" that measures the number of times a particular service level is missed during an interval, such as a rolling 12-month period. If the frequency factor is triggered, the percentage to be applied against the total service credit is increased by some factor [ASP].

It would be helpful to define the penalty fee for the lack of availability as a measurable function, depending on frequency and peak periods, and make it transparent to the stakeholders.

2.4 Success/Assessment Criteria

In this chapter we define the technical and commercial success criteria for the TAPAS research and development project. These criteria should be used during the evaluation of the project in order to compare the results with what has been planned. Most modern software development processes already anticipate the need to adjust one’s goals and targets during a project. The same is even more true for a research project which heads into unexplored areas of technology. Hence, it must be kept in mind, that it might be more fruitful to understand, why a criterion could not be fulfilled, rather than just to notice the failure.

2.4.1 Technical Criteria

Exploiting the SLA guarantees and commitments discussed in chapter 2.3, we can say that SLAs defined in TAPAS must include the aspects outlined in chapter 2.3.

- **R2.1**: Formal SLAs must allow to specify at least the aspects:
- **R2.1.1**: performance
- **R2.1.2**: availability
- **R2.1.3**: reliability
- **R2.1.4**: maintainability
- **R2.1.5**: security
- **R2.1.6**: monitoring
- **R2.1.7**: penalty

Service level specification (SLS) requires a dedicated language for modelling of qualitative and quantitative aspects of SLAs and interaction between them. Using the modelling language ASPs and clients should be able to negotiate SLAs by reasoning about consequences. Furthermore, the language should help to identify errors in even complex models. In order to enhance SLS and make it transparent for the client there should be modelling and reasoning tools for the specification of SLAs. A tool that can be used to formulate SLAs and automatically generate contracts in paper form whilst retaining an electronic representation of the data would avoid the necessity of transforming from paper form into an electronic form. An appreciated side effect of using such a tool for specification of SLAs is that it can be utilized in an early phase of contract negotiation. Customer and ASP can take advantage of the clear semantics. Changes during the negotiations can be incorporated into the actual hosting process immediately.

- **R2.2**: Modelling of SLAs should be supported by an appropriate modelling language.
- **R2.3**: The SLA modelling process should be supported by one or more modelling tools.
- **R2.4**: If modelling and reasoning tools should not only be integrated into one tool, they should interact seamlessly.
- **R2.5**: It would be desirable to have visual editors for the specification of SLAs.

A TAPAS SLA modelling language must provide adequate means to model the aspects discussed in chapter 2.3. In contrast to current mostly constant values or simple functions, users should be able to construct more sophisticated stochastic descriptions of the system's behaviour.

- **R2.6**: SLA modelling language should be expressive enough to allow reasoning about the aspects:
- **R2.6.1**: performance
- **R2.6.2**: availability
- **R2.6.3**: reliability
- **R2.6.4**: maintainability
- **R2.6.5**: security
- **R2.6.6**: monitoring
- **R2.6.7**: penalty

One of the most important criteria is the possibility to express complexity of SLAs. Due to the fact that the TAPAS approach incorporates many layers of the ASP model, we have to specify SLAs systematically, namely vertically and horizontally. Vertical SLAs are agreements between the layers, i.e. application versus network. Horizontal SLAs are agreements between elements of the same layer such as agreements between distributed components.

- **R2.7**: Specification of both horizontal and vertical SLAs must be supported by TAPAS techniques.

On the other side, the relationships between different stakeholders described in chapter 2.1.2 should be governed by appropriate agreements. We shall be able to define SLAs between users, owners and providers. Here it is important to determine correlations between depending SLAs. This is not a trivial aim, especially if we take the qualitative and quantitative aspects of SLAs into consideration.

- **R2.8**: SLA modelling process should regard the correlation of SLAs items
- **R2.9**: SLA modelling process should take into consideration the interaction of depending SLAs

- **R2.10**: It shall be possible to define SLAs between users, owners and providers described in chapter 2.1.2.

The next criterion is the possibility to translate an SLA into a deployment descriptor. That would enable the automatic deployment of TAPAS components in containers and guarantee the security level of services. QoS negotiation, establishment and adaptation facilities will be added to the middleware and will be used by component containers to make them QoS aware [EP01]. This criterion is very important for TAPAS middleware architecture governing component execution.

- **R2.11**: Tool-based translation of a SLA into a deployment descriptor that to some extent automatically enforces the service levels has to be provided.

Performance and reliability are often gained today by clustering mechanisms of application servers. In future APS scenarios clustering will still be used to achieve such features as availability and scalability. The TAPAS execution environments must therefore be able to interact with clustering mechanisms. Additionally there might be new QoS-aware clustering features, which might then already be utilized in the modelling of SLAs.

- **R2.12**: TAPAS QoS-aware component technology must be able to work together with clustering mechanisms.
- **R2.13**: SLA modelling techniques should regard clustering techniques and, if applicable, offer new means to deal with new QoS-aware clustering features.

In current application hosting an ASP is not really supervised regarding the fulfilment of SLA clauses across organisational boundaries. One can imagine a constellation where an ASP is dependent on services providers (s. chapter “Stakeholders in the ASP Model”). If the availability falls due to the lack of bandwidth, the component could warn the ASP about the situation. It would be desirable to implement an automatic service, such as switching to another ISP as a self-repairing mechanism. The component could also generate remedy charts that display the underperformance of upstream providers. Since the provision of bandwidth is the responsibility of the ISP, the ASP is entitled to impose penalties according to the SLA.

In order to eliminate mutual distrust in distributed ASP scenarios, it is necessary to provide transparent techniques for monitoring compliance to SLAs. If this last criterion is provided, the ASP model will enjoy more popularity and trust.

- **R2.14**: It must be possible to monitor and measure compliance to SLAs for all stakeholders
- **R2.15**: The auction application described in *appendix A* can be implemented using the technology presented in the TAPAS deliverables and documented by means of assessing conformance to all requirements

2.4.2 Economic Criteria

Apart from the technical improvements, the TAPAS project shall have a leverage effect on the economic situation of the ASP market and thereby on the IT market as a whole. The benefit from growth in the ASP market is not only economic wealth but even more importantly the creation of new jobs. Growth of the ASP market can be induced by the improvement of existing businesses or in creation of new business propositions.

- **R3.1**: Add additional value propositions for existing businesses

Firstly, the improvement of existing ASP business involves reducing costs for operating as an ASP. While hard- and software costs (e. g. for servers and software licenses) can hardly be influenced by TAPAS, the maintenance costs may be reduced by reducing the number of human interactions with the production system. This is the standard strategy in all industrial branches, because the labour share is the biggest cost driver in the service industry. When applying this method to ASP we see that the most frequent human interactions with the system refer to the negotiation, implementation and monitoring of service level agreements. The adesso hosting staff, for instance, spends a significant amount of time not only supervising the hosted applications (e.g. checking their availability), but also a large amount of time generating and analysing statistics of service quality (e.g. for the bandwidth from the underlying ISP). Hence, for a hosted application based on TAPAS results, the maintenance should be simplified in a measurable way.

- **R3.2**: Reduce the costs of defining, monitoring and entering service level agreements

Despite of the pure economic advantage there is as well an improvement in consulting and software development, which should be exploitable. IT-consulting and software development companies like adesso can make use of the knowledge gained by TAPAS in more than one way. Firstly, a client can be convinced by the features of QoS TAPAS technologies during the conceptual phases. This means the client’s business set-up will be able to offer a unique selling proposition as it benefits from the TAPAS results. As a second step, the consultants shall be able to assist the client’s business planning with modelling and reasoning techniques. Furthermore, on a more technical level, software engineers will know how to make use of TAPAS results like improved open source containers to build the client’s application and finally how to deploy and run the application. Therefore we can formulate the support of all phases of ASP software development as a success criterion for the TAPAS project.

- **R3.3**: Support for the specialization of businesses

On the other hand, improvement of existing business will not be able to cause the desired impact on the IT market on its own. During the recent years, the so-called “new economy” was a synonym for a rapidly growing market. Accordingly, we see that real growth was caused by completely new types of businesses together with a break-up of traditional business scenarios. Small, specialized companies offered services that where not possible only few years before, e.g. Yahoo! or amazon. Therefore TAPAS results shall as a criterion support the specialization of companies as well as enable completely new areas of business. Therefore, the TAPAS results shall support the specialization of companies as well as enabling completely new areas of business.

- **R3.4**: Enable new types of businesses

e-Business, however, consists in the implementation of a value chain with means of IT. After examining real world companies like amazon, it can be said that the longer the implemented value chain becomes, the higher the benefit for the owner of the application will be. If the staff members of an online shop would have to check manually if every credit card is valid, benefit would fall rapidly. If the staff of an online shop had to manually check the validity of every credit card, the benefit of the value chain for the owner would fall rapidly. Hence, we can postulate that a success criterion for TAPAS it shall be a support of business-to-business integration.

- **R3.5**: Facilitate the lengthening of the value chain

3 Internet service models and network SLAs

We distinguish and describe the IP services under two broad headings: i) *network performance engineering* and ii) *transport semantics and functionality.* Under the first heading, we are more concerned with the actual specifics of the way that an implementation of a network offering a set of services really performs under particular load while under the second heading we look at protocol behaviour and use; unicast vs. multicast, reliability, security etc. and the effects of these choices.

First we present a study of the state-of-the-art in IP standards and technologies. Then we propose some solutions for the project, and provide a rationale for our recommendations; providing some information on technical products for Quality of Service (QoS) and multicast. We discuss a few examples of Service Level Agreements (SLAs) offered by typical commercial and commercially minded Internet Service Providers (ISPs). Finally, we comment on work that will need to be done in the TAPAS project to fill any gaps in the available functions and services.

3.1 Introduction

It is necessary to provide an overview of some of the key network standards and technologies as they are likely to have an impact in shaping future IP services. These mechanisms are particularly relevant to the design and management of next generation Service Level Agreements (SLAs).

Standardisation within the Internet Engineering Task Force (IETF) Working Groups, initially attempted to provide guaranteed QoS and Internet-wide service definitions by proposing changes in the architecture of the core routers (intserv). However it soon became obvious that this attempt suffers from serious scalability problems and subsequent efforts focused in defining “building blocks” out of which new services could be built without defining the services per-se. Clearly the new trend is to push all complexity to the edges of the network leaving the core routers simple (diffserv).

In the time frame of the TAPAS project we have to make the following pragmatic assumptions; first that there is no Internet-wide QoS architecture and it is rather unlikely that there will be one in the near future (3-4 years) and second that IP multicast is still not ubiquitous at the network layer and it is provided mainly through application level solutions.

Our goal in the rest of this document is to describe what an applications service provider (ASP) can expect from a internet service provider (ISP) in terms of available standards and technologies for *QoS* and for *reliable multicast transport*. These two issues are particularly relevant to the objectives of the TAPAS project.

First we provide an overview the Integrated Services Architecture (intserv) and the related mechanisms. Intserv is still the only option for requesting and obtaining “hard” quantitative guarantees from an IP network and although its scope of deployment is limited it can be useful in certain situations.

Then we discuss the Differentiated Services Architecture where the standard building blocks leave plenty of space for experimentation with the composition of new services. We also discuss problems with providing multicast in a Diffserv network, this is still research in its early stages and its relevance highly depends on the success of the Diffserv architecture.

We also provide a short description of the most popular solution for reliable multicast; Cisco’s Pragmatic General Multicast (PGM).

In the second part we discuss network SLAs (parameters, QoS metrics etc.) and provide a brief description of the main features present in production level SLAs offered by network providers, operators such as UKERNA, AT&T, WorldCom.

Finally we provide the general directions for the work we intend to pursue in the course of this project.

3.2 Integrated Services and RSVP

Integrated Services architecture [INTSERV] (RFC1633) specifies a service model that can accommodate multiple Quality of Service (QoS) requirements. In this model the application requests a specific kind of service from the network before sending data. The application uses explicit signalling in order to make this request; the application informs the network of its traffic profile and requests a particular kind of service that can encompass its bandwidth and delay requirements. The application is expected to send data only after it gets a confirmation from the network. It is also expected to send data that lies within its described traffic profile.

The network performs admission control, based on information from the application and available network resources. It also commits to meeting the QoS requirements of the application as long as the traffic remains within the profile specifications. The network fulfils its commitment by maintaining per-flow state and then performing packet classification, policing, and intelligent queueing based on that state.

There are two services defined within the Intserv architecture: i) the Controlled Load Service (RFC2211) which resembles best-effort but in a lightly utilised network (very low packet loss rate) and ii) the Guaranteed Service (RFC2212) which specifies guaranteed bandwidth and bounded delay.

The Intserv QoS API comes in the form of RSVP signalling. RSVP (RFC2205) is the first significant industry-standard protocol for dynamically setting up end-to-end QoS across a heterogeneous network. RSVP runs over IP and allows an application to dynamically reserve network bandwidth. Using RSVP, applications can request a certain level of QoS for a data flow across a network.

RSVP is the only standard signalling protocol designed to guarantee end-to-end network bandwidth. RSVP does not perform its own routing; instead it uses underlying routing protocols to determine where it should carry reservation requests. As routing changes paths to adapt to topology changes, RSVP adapts its reservation to the new paths wherever reservations are in place. It also provides transparent operation through router nodes that do not support it. RSVP works in conjunction with current queueing mechanisms (it is not a replacement of these mechanisms). RSVP requests the particular QoS, but it is up to the particular interface queueing mechanism, such as weighted fair queueing (WFQ) or Class Based Queueing (CBQ) to implement the reservation. RSVP can be used to make two types of reservations: controlled load and guaranteed rate services.

RSVP scales well with multicast. RSVP uses receiver-driven reservation requests that merge as they progress up the multicast tree thus it scales very well to large multicast groups. It can also be used to make unicast reservations though it does not scale as well with a large number of unicast reservations.

RSVP is an important QoS signalling mechanism, but it does not solve all problems addressed by QoS, and one of its weaknesses is the time required to set up an end-to-end reservation.

Hosts and routers use RSVP to deliver reservation requests to the routers along the paths of the flow and to maintain router and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate, the largest amount of data the router will keep in the queue and minimum QoS to determine bandwidth reservation.

A host uses RSVP to request specific QoS from the network on behalf of an application data stream. RSVP requests the particular QoS, but it is up to the interface queueing mechanism to implement the reservation. RSVP carries the request through the network, visiting each node the flow passes through. At each router, RSVP attempts to make a resource reservation for the flow using its own admission control module, exclusive to RSVP, which determines whether the node has sufficient available resources to grant the QoS request.

If either resources are not available or the user is denied administrative permission, then RSVP notifies the application that originated the request. If both attempts succeed, the RSVP daemon sets parameters in a packet classifier and packet scheduler to arrange so that the flow receives the appropriate QoS.

However, Integrated Services and RSVP are considered excessively complex and posses poor scaling properties.

Router forwarding performance can deteriorate due to the classification and scheduling overhead. Moreover the resource requirements (processing, memory) for implementing RSVP on the routers increases proportionally with the number of reservations. Also another important concern has to do with the policy and control issues related to RSVP; e.g who is authorised to make reservations which calls for additional access control and accounting.

Due to these concerns as discussed in RFC 2208 (applicability statement) it is very unlikely that providers with large backbone networks will deploy the Intserv mechanisms in the foreseeable future.

In more controlled environments, like corporate intranets, RSVP would be able to provide QoS guarantees since the issues of scale are less critical. Nevertheless such confined environments tend to be generally well-provisioned and QoS degradation is less of a problem.

3.3 Differentiated Services

The most recent attempt to introduce resource management and QoS in the Internet has been the work by the DiffServ Working Group of the IETF [DIFFSERV]. The DiffServ model improves the scalability of QoS provisioning by pushing state and complexity to the edges of the network and keeping the classification and packet handling functions in the core of the network as simple as possible. In summary the flows are classified, policed and shaped at the edges of a DiffServ domain.

The DiffServ approach is to classify services on a per-hop basis using the DSCP field in the IP header. The nodes at the core of the network handle packets according to a Per Hop Behaviour (PHB) that is selected on the basis of the contents of the DS field in the packet header. The number of DS code points and the number of PHBs is limited and therefore a large number of flows can be aggregated from the point of view of the core router.

A PHB is defined as “description of the externally observable forwarding behavior of a DS node applied to a particular DS behavior aggregate”. The actual mechanisms causing this behaviour are not part of the PHB description. Five PHBs have been defined; the *Expedited Forwarding* (EF) and the four *Assured Forwarding* (AF1, ..,4) PHBs:

The EF PHB provides a low delay, strict maximum rate packet treatment , it can be used for services which require low loss, low latency and assured bandwidth. The maximum arrival rate of the aggregate traffic should meet the configured rate.

Each one of the four AF PHBs has associated with it three drop precedences. The semantics each AF PHB (classification, metering, shaping) are determined by the network designer and may involve re-marking (assignment of a packet to different drop-precedence within the same AF PHB or even to another PHB).

Beyond the PHBs the other aspect of the DiffServ-type service description is related to traffic conditioning. The mechanisms here involve Classification, Metering, Marking, Shaping and Dropping. Together they form a traffic conditioning specification. Usually these parameters are set on the basis of classification parameters (how to recognize specific flow or aggregates of flows).

The Differentiated Services model can satisfy many QoS requirements. However, unlike the Integrated Services model, an application using DiffServ does *not* explicitly signal the router before sending data and the routers do not maintain per-flow state.

With differentiated services, the network tries to deliver a particular kind of service based on the QoS specified by each packet. This specification can occur in different ways, for example, using the IP Precedence bit or source and destination addresses. The network uses the QoS specification to classify, shape, and police traffic, and to perform intelligent queueing.

Typically, this service model is appropriate for aggregate flows because it performs a relatively coarse level of traffic classification.

In certain future services based on the DiffServ Architectue it may still be necessary for a user to signal a request to a DiffServ domain. This is achieved by communicating with a *Bandwidth Broker* (BB), a dedicated node in each domain, which keeps track of the amount of bandwidth that is available and processes admission control requests from customers or other BBs of the adjacent domains. A BB may also be responsible for installing or altering traffic profiles (SLA management) in the boundary nodes. For this type of signalling in a DiffServ environment RSVP could still be used with a DS specific reservation object.

With the development of the DiffServ model the focus has been mainly on packet level behaviour with the purpose defining building blocks for scalable differentiated services and there has been considerable progress in the area of Per Hop Behaviours (PHBs).

However there has not been systematic study in the areas of

- dynamic allocation and pricing of bandwidth at the edge links of a DiffServ domain
- the feasibility of maintaining consistent SLAs across interconnected networks where dynamic allocation happens only at the edges
- also multicast poses some interesting challenges with regards to the Differentiated Services and the management of SLAs.

3.4 DiffServ and Multicast

Since IP multicast implies that anyone can be a sender to multicast group (without requiring prior membership) resources should be managed on every potential tree (either source based tree or a shared tree).

With multicast it is difficult to predict the network resource requirements, because there may be one ingress node and multiple egress nodes as a result of the multicast packet replication in the interior routers. This can be a problem in situations with highly *dynamic group membership*; the addition of new members in the group causes more traffic out of the egress nodes and this could adversely affect other traffic in the network. Multicast packet replication happens during the routing process and the replicated packets get exactly the same DS codepoint as the incoming packet and therefore exactly the same forwarding treatment in subsequent nodes.

Another consideration with multicast is *group heterogeneity* (with respect to the QoS requirements of different receivers). Ideally participants with more modest QoS requirements should be able to participate in groups where the other participants use better service classes. Thus a challenge is to have concurrently more than one service classes within a group.

There are basically two approaches for providing multicast in a Diffserv environment:

Edge and Core Routers both Multicast Capable: the core routers as well as the edge routers maintain membership information – and on join or leave all the routers update their state for the flow that is affected. It is contrary to the ideas underlying the DiffServ approach of zero state in the core routers and it may be justifiable only if the number of multicast flows is considerably smaller than the number of unicast flows.

Only Edge Routers are Multicast capable: this approach is simple to implement and it is highly scalable but does not provide bandwidth savings inside the DiffServ domain as the same packet may traverse the same link several times depending on the number of times the link is used in the paths from the ingress to the egress router. The approach is desirable for sparse groups where packet replication in the core routers is less likely to happen compared to dense groups. It is compatible with the goals of the DiffServ architecture.

There is also the Encapsulation-based approach with multicast capable core routers: with this approach the multicast tree structure is encapsulated within the multicast packets as additional header. It does not require per-group state in the core routers being inline with the Diffserv design philosophy and it scales well with the number of multicast groups. However the fact that the core routers should understand how to interpret the embedded information in the packet header make this approach less favourite candidate.

Providing QoS is often associated with scalability problems since QoS requires flow specific information installed in the routers. These concerns are justifiable in the case of unicast flows (the routing tables do not maintain flow specific state). However Multicast Routing tables maintain multicast flow (point to multipoint stream of packet) in the form of (source, group) pairs. Thus adding flow specific QoS state should be straightforward for a relatively small increase in the routing state.

3.5 Pragmatic General Multicast

Among several reliable IP Multicast Transport Protocols that have been proposed to the IETF, PGM is unique in that it minimizes the probability of NAK implosion and at the same time minimizes the loading of the network due to retransmissions of lost packets. Thus PGM provides a highly reliable IP Multicast transport mechanism that should be able to scale to networks as large as the Internet.

Reliable multicast protocols focus on the elimination of positive acknowledgments (ACKs) which are typical in reliable unicast protocols such as TCP. Positive ACKs are an excessive network overhead for one-to-many communications. As a result, all reliable multicast protocols have substituted some form of negative acknowledgment

(NAK) that is invoked by receivers only when blocks or packets of information are not received.

In Pragmatic General Multicast (PGM) when a receiver detects a lost packet, it unicasts a NAK to the next-hop upstream PGM router. Any additional NAKs from this subnet are suppressed because the router supress them by multicasting a NAK Confirmation (NCF) to the subnet.

Once a receiver sees the NCF, it will not send a redundant NAK. The PGM router stores the group address and the interface ID requiring retransmission of the lost packet (i.e., the router stores "retransmit state" information) and then unicasts the NAK upstream. The process is repeated until the NAK is received by the source.

Once retransmit state has been set by a router for a given packet of data, additional NAKs for this data will be eliminated, i.e., not forwarded upstream. When the source retransmits the packet, the PGM router consults its retransmit state table and forwards it only to those segments containing receivers that require the packet.

PGM also uses the concept of local recovery of lost packets to address the problem of retransmissions loading of the network. In this process, any receiver that receives an NCF and is in possession of the missing data may multicast the missing data. Because the PGM router has maintained retransmit state, the multicast will only be propagated downstream of the retransmitting receiver. Dedicated hosts may also be configured as Designated Local Retransmitters (DLRs) for a number of multicast groups.

In response to the NCFs it receives for these groups, the DLR will multicast a repeat of the missing data to receivers below it in the distribution tree. Local recovery greatly reduces the load on the core of the network due to retransmissions and reduces the latency of packet recovery for all receivers.

3.6 Network Service Level Agreements

The Internet offers a best-effort packet delivery service, this means that there are no quantitative guarantees about whether a packet will be delivered, in general terms packet delivery “should be the rule not the exception”. This loose definition of the best-effort service places too much *performance risk* to clients who want to use the Internet for business critical applications.

Moreover, there is a conflict of interest as the Internet Service Providers (ISPs) try to increase their earnings potential by increasing the traffic levels on their network. The side effect of this strategy is increasing congestion levels and declining levels of service to the clients.

The Service Level Agreements is an attempt to impose some control on the extent to which the ISP can oversubscribe.

A network *Service Level Agreement* (SLA) is a *contract* between an Internet Service Provider (ISP) and a service client, which specifies the services and their performance using some quantifiable Quality of Service (QoS) metric. The SLA usually have clauses for the penalties, which the provider will incur in case of failure to meet the contractual obligations.

Network SLAs are almost always *bilateral* involving two domains that are logically adjacent. One of the domains is the service provider and the other the client (an organisation or an individual). We make a distinction between SLAs and Peering Agreements between ISPs, which occur at Network Access Points (NAPs) and involve the exchange of routing information (e.g. BGP4 routes).

Practice has shown that it is very difficult to provide *inter-domain* SLAs in the Internet because different segments of the end-to-end path are under the administrative control of different service providers. The ISPs simply cannot control what they do not own and they usually manage the SLA up to the network end-points that are under their control. Sometimes ISPs offer end-to-end SLAs by having their own point-of-presence into the clients' sites connected via leased lines.

Also The SLA imposes some minimum levels of service that the ISP should meet under the terms of the access contract. However the wide range of network and link level technologies in use makes the quantification of service levels a rather controversial issue.

There are two fundamental issues to be addressed here, first how to describe appropriately what constitutes "an acceptable service level" and second how to measure (monitor) an SLA, we shall discuss each one in turn.

3.6.1 Technical requirerments

The network SLA is both a legal and a technical document determined by network administrators, lawyers, business executives etc. The technical part of the SLA sometimes referred to as *Service Level Specification (SLS)* describes the performance levels of the service. It uses technical terms which may involve the traditional network QoS metrics; delay, loss, bandwidth, jitter that characterise the performance along a path, a domain or a single link and most likely higher level procedures like admission control, topology, traffic direction, security etc.

Common network performance parameters that appear in the technical part of the SLA include the following:

- *Service schedule* specifying when the contract should apply (e.g which time of day, days, the maintenance schedule etc.)
- *Network availablity* what is the guaranteed network up-time expressed as a percentage. An equivalent metric is the maximum downtime, usually defined per month or per year.
- *Network scalability* refers to the topological scope of the service. What is the network's reach, how is the network extended to reach a new customer site or a new branch office location. Also whether the network supports IP multicast for group communication. The issue of *Peering Arrangements* is very important here. Peering refers to the interconnections between the ISP in question and other ISP. Each peering arrangement has all the attributes found in a network SLA so this is a very complicated issue to be addressed in this document but we intend to do so in future work.
- *Redundancy,* this applies both to the interconnections with the other ISPs (peering) and the access link between the ISP and the client. It is important that there is no single point of failure in the ISP's network and details of how redundancy is provided are important in this context.
- *Network loss*, what is the acceptable level of packet loss (possibly specifying the packet size and the method for measurement). Some applications which involve real-time human interaction (e.g voice) require very low packet loss rates (1% or less) while other applications are more elastic (e.g Web browsing or file transfers) and they can tolerate higher packet loss rates.
- *Network dela*y, refers to the round-trip-time between client sites and predetermined points inside the network (specification of packet size and

measurement methodology is necessary here). This is an important metric because the response time for real-time interactive applications must be less than 100msec while others may tolerate higher delays.

- Network bandwidth. The issues here are more complex and an exhaustive analysis is outside the scope of this document. They concern the capacity of the network, the methods used for allocating and managing bandwidth, the level of utilisation.

Depending on the service model deployed by the ISP the technical parameters are also likely to involve lower level details such as

- *flow descriptors* used for identifying uniquely the packet stream(s) of the SLA (using the DSCP field or the common five-tuple of the IP header; the source and destination IP addresses, the corresponding port numbers and the protocol),
- *traffic descriptors* which specifies the traffic envelope of the sources (e.g identifies in and out of profile packets using a token bucket),
- *excess treatment* which specifies how the out-of-profile packets are being treated (potential options here are re-marking, shaping or dropping).

For example the popular *virtual leased line* (VLL) service should be described by an SLA with quantitative guarantees. The VLL is a point-to-point service of low latency, low loss probability at a specified transfer rate with the excess traffic being discarded. A popular application for such a service would be IP telephony. The flow descriptor would specify the EF PHB and the (source, destination) IP address pair, the traffic descriptor would specify the rate and depth of the token bucket and the performance metrics would give statistics (mean and quantile) of the end-to-end delay and packet loss rate.

Ultimately the purpose of the SLA and the description of the service provided will encompass the meaning and the relevance of the technical parameters discussed above as well as the level of detail to which they must be described.

3.6.2 SLA Monitoring

The issue of SLA involves the description of the measurement and reporting procedures. For example an ISP would find unacceptable end-to-end performance measurements where the flows in question traverse third–party ISP networks and the conditions (load) at the end-systems are not well specified.

On the other hand many clients do not have the means or the expertise required for supervising the fulfilment of their SLAs (for instance due to lack of technical knowledge or staff) and this may lead to acceptance problems with the ISP terms.

Another issue here is that measurement mechanisms should be non-intrusive, and not overload the network and be the cause for performance degradation.

A common technique is to use “ping clusters”, however the results produced by this type of measurement (round trip times, loss, jitter) are not well correlated with TCP and UDP performance.

Monitoring/management of an SLA becomes very important for guaranteeing the appropriate levels of service and building a trust relationship between the client and the service provider. Moreover it may become the differentiating factor between an ISP and its competitors.

3.6.3 Open issues

Multi-lateral SLAs ae simply unmanageable however even bilateral SLAs are still to a great extent ad-hoc and there are several problems that need to be addressed

- SLA Information Model, i.e. a standard template for SLAs which will allow co-operation and negotiation between the providers and clients involved.
- SLA negotiation protocol, which involves the functional aspect, the security and potentially (although not likely) an inter-domain aspect.
- the end-to-end aspects of service, which involves the agreements that have to be established between clients and providers.
- the automatic management of the SLA/QoS, there is no provision for automatic negotiation or for translating SLA requirements to parameters used for the technical configuration of network equipment.
- the issue of assurance; the perceived quality should be inline with the expected quality (i.e. the achieved performance is in line with the specifications in the contract).

All the above issues often appear in the research agenda, however we intend to concentrate mainly on the end-to-end aspects of the service and the SLA information model (as it applies to both application and network level SLAs). The issue of assurance, at least for the network SLAs, we expect the parties entering into the SLA to be able to manage the risk associated with the use of shared connectionless packet network by adopting appropriate engineering solutions and/or structure (clauses) in their contracts. The issues related to SLA negotiation we shall encounter later in the process and they will depend on the form of the contracts as well as whether the customer has access to several providers (i.e a market for such contracts is available).

3.7 Case Studies of Network SLAs

In this Section we examine the technical commitments that appear in several *production service* SLAs.

3.7.1 UKERNA IP transmission service SLA

UKERNA is the operator of the U.K academic network –which provides network services to all U.K academic institutions. UKERNA offers an array of services however the main network service we are interested in is the so called "*Basic IP Transmission Service*".

Central role in the UKERNA SLA have the concepts of *availability* and *mean time between failures*. In order to define *availability* of the basic IP transmission service, the *accessibility* of a client site must be defined first. Accessibility means the ability to successfully transfer data in both directions between the client institution and a well-known point in the provider's network. However accessibility cannot be equated to availability for the purposes of the SLA because the former includes factors that are outside the provider's operational control (e.g equipment or power failures at the client site or delays in fixing faults due to absence of staff at the client institution). Therefore network availability should be determined by the accessibility by discounting all inaccessibility periods that are due to scheduled service maintenance (usually published weeks in advance) or events local to the client institution. Network availability (*NetworkAvail*) is defined as the fraction:

$$NetworkAvail = (Sched – Unavail) / Sched$$

where *Sched* is the scheduled service time, *Unavail* is the total period of unavailability.

Failure means any interruption in the operation of the service, which causes a period of unavailability. So *Mean Time Between Failures* (MTBF) is a metric that is used for averaging the failure rate and it is measured in *incidents per hour*. For instance a target MTBF of more than a thousand hours is a rate of less than 0.001 incidents per hour and it is calculated each month by dividing the number of failures by the number of institutions and the number of hours in the month and for a twelve month period by averaging the rates observed in each of the constituent months of the period.

According to their model and terminology a service is defined in terms of i) a Specification, ii) Performance Indicators and Service Levels and iii) Reporting.

The Basic IP transmission service provided to all client organisations is the transmission of IP data and it is considered to consist of i) an *access component* between the institution’s site (client) and the provider’s backbone and ii) a *core component* for providing transit across the core (backbone).

For both components the network path is considered inaccessible if it is not possible to transmit or receive for a period in excess of 60 sec. Also if the performance of the path is severely degraded i.e. more than 60% of the packets transmitted on the path are lost for a period of more than 5 minutes.

In the future they will attempt to provide performance indicators that reflect the risk of a degraded service when the service is available but at poor performance level.

For the access to backbone the SLA commits to the following terms: *Availability* 99.7% to more than 90% of client organisations (99% to more than 96.5%, 97% to more than 99% and 93% to more than 99.7% of the client organisations) these figures are calculated annually from the monthly averages. A mean time between failures is at least 1000 hours. The target for *maximum latency* between the client and the nearest point on the national backbone is 15msec for 95% of transmissions using 128 byte packets over any 30 minute period.

The provider should report to each client institution every month: the number and duration of incidents when the institution was not accessible, the number and duration of incidents when the network is unavailable, the total traffic on the institution’s attachment point.

For the core network service, the SLA supports the transmission of IP unicast and multicast between access points at the boundaries of the core. The core is defined in terms of backbone access routers. The availability of the core should exceed 99.9% and it is assessed by weighting each unavailability incident affecting the core network according to the number of paths disrupted as a proportion to the total number of paths across the core network. Thus

$$CoreAvail = (Serv - \Sigma (w * Out)) / Serv$$

where *CoreAvail* is the availability of the core *Serv* is the service time *w* is the weight of a given outage and *Out* is the duration of the outage.

The target for *maximum latency* between any two points on the national backbone is 15msec for 90% of the transmissions using 128 byte packets and measured over any 30 minute period.

The provider should report overall core network availability, number and total duration of incidents of network unavailability for each path across the core network (in the future possibly a traffic matrix for the core network).

3.7.2 AT&T Managed Internet Service

AT&T advertises its SLAs as “industry-leading” and offers guarantees of 99.99% availability, network-wide (U.S) monthly average delay of 60msec and packet loss rate of less than 0.7%. It also provides online access to traffic summary reports for access line utilisation and peak activity periods.

3.7.3 Worldcom IP VPN SLA

Worldcom also offers SLA for managed IP Virtual Private Network (VPN) with the following characteristics. The initial contract is at least for one year and it should involve at least three VPN sites, the “access circuits” are provided by Worldcom. The access circuits are in fact the links that connect the Customer’s Premises Equipment (CPE) to the Worldcom network. The SLA for the VPN is applicable if the sustained use level for each access circuit is less than or equal to 50% of the total capacity of that access circuit. If this happens for two consecutive months then the company notifies the customer. All calculations are based on the data and record kept by Worldcom.

The latency commitment is an average of maximum 120 msec (round trip) between CPEs on the customer sites for VPNs with all sites either in Europe or in North America and an average of maximum 300 msec for VPNs with sites both in Europe and North America. The clocks at the CPEs are synchronised using NTP for this measurements. With regards to availability they commit to 99.9% averaged over all the VPN sites for VPNs with ten or more sites and 99.8% for VPNs with three to nine sites.

3.7.4 The Worldcom -UKERNA SuperJanet4 SLA

UKERNA and Worldcom have entered into an SLA for the SuperJANET 4 backbone network. In this occasion UKERNA (network operator) has been the client purchasing the network from Worldcom (carrier). The details of the contract have been kept confidential but the general principles of the SLA were made publicly available. The target availability for the network is 100% and Worldcom guarantees that individual circuits will have 99.95% availability giving a maximum permitted loss of service of 22 minutes per month. Faults on links are classified according to the impact on the service and target repair times and priorities are being specified. Worldcom also provides eight core POPs (Points of Presence) throughout the U.K where the UKERNA backbone routers will be co-located with the Worldcom’s transmission equipment. Worldcom commits to provide monthly service reports with details about the fault incidents and the level of service achieved by the network. A quarterly service review to implement improvement plans and an annual SLA review.

3.7.5 Common features

From the SLAs discussed above the first three are of particular relevance to the project objectives because they involve agreements between institutions or enterprise clients and ISPs, for instance in TAPAS the client could be the Application Service Provider (ASP). The SuperJANET4 SLA is slightly different as it involves essentially an ISP (UKERNA) in the client role and WorldCom as the network infrastructure provider.

The specification of availability is very important in all SLAs. Another interesting aspect is that the SLAs above do not include bandwidth or throughput specifications.

These are implicitly built in the provisioning of the provider's backbone network, the peering arrangements with other ISPs and the capacity of the access lines to the ISP's point of presence (POP).

Latency (or round trip delay) between the client and a well known point in the provider's domain is always specified since this is an important attribute of the provider's physical infrastructure. The packet loss ratio is also specified but it applies only when the sustained use of the access links is sufficiently low, for instance below 50%.

When the provider fails to meet the commitments made in the SLA the customer is usually, reimbursed in discounts for subsequent service periods (service credits) while a financial settlement is less common.

3.8 Future directions for network SLAs

In this section we discuss the experiences from large-scale field trials with router based QoS mechanisms for SLA enforcement. We discuss briefly the issues of interactions between ISP and multihoming as an option for the ASP also an end-to-end approach to offering differentiated services through modifications in the behaviour of transport protocols.

3.8.1 The future of QoS – the Internet2 experience

Experience so far has shown that there are fundamental problems with the introduction of new Internet service models based on router mechanisms and it is rather unlikely to see in the future SLAs covering inter-domain paths.

One of the goals of the Qbone Working group [QBONE] for example was to implement an inter-domain, EF-based virtual wire service in the Internet2 environment. The effort was called Qbone Premium Service and its building blocks involved Expedited Forwarding (EF) treatment on all the routers along the path and well-specified SLAs across each trust boundary. The SLAs would describe the EF aggregates that will be allowed to transit and the assurances provided to each and strict policing/shaping at the ingress/egress of each domain.

After several years the team admitted that they had to face what they described as "intractable deployment problems" which led to the Qbone Premium Service initiative to be suspended indefinitely. The approach involved dramatic changes to network operations and business models, requirement for policing on all access interfaces on all providers and lack of means for verifying the service (both users and providers). Moreover the rationale of the overall effort was not sound because the Internet2 network was very well provisioned and application performance problems were almost never caused by congestion. The usual causes were problems in the end-points (e.g TCP stack problems due to small socket buffer sizes) or problems with the configuration of the local switches (shared media etc.).

The focus of the Qbone design and deployment efforts has now shifted to lightweight mechanisms that allow differentiation and therefore add value without adding significant operational complexity (no reservations, policing or admission control).

3.8.2 Statistical models for end-to-end performance

The last few years there has been a considerable amount of work on new resource management models for the Internet where the challenge is to use a combination of end-point control, network dimensioning and pricing mechanisms, in order to provide statistical bounds on the end-to-end performance.

The end-to-end performance is determined by a collection of phenomena that operate at different time-scales. We have buffer management (dropping, marking) and packet forwarding operating at the microsecond-to-millisecond time frame. Congestion avoidance and control for individual flows, which operates in the end-systems at the millisecond to second time-scale (in the order of a round trip time). Routing and traffic management for aggregates of flows, operating in time-scales ranging from seconds to minutes or even days. A combination of these phenomena can influence the structure of SLAs offered by ISPs [GIBB1].

The congestion control mechanisms in particular are very important because they are adaptive and provide the network the flexibility to operate under significantly different load conditions (with the users observing different performance). These mechanisms essentially allow the network capacity (bandwidth) to be shared “fairly” among users. The congestion control in TCP operates as follows; when a link inside the network becomes overloaded and one or more packets are lost then the sender infers this by observing certain patterns in the acknowledgement packets returned by the receiver, (e.g triple duplicate ACKs or no ACKs). Following that the sender decreases its sending rate or “window size” (the number of packets allowed to be outstanding in the network) appropriately and through a repeated cycle of increase decrease phases the available bandwidth is discovered and shared among users. The most widely used model for describing TCP throughput is the model by Padhye et al

$$Throughput = \min\left\{\frac{W\max}{RTT}, \frac{1}{RTT\sqrt{2bp/3} + T0\min(1,\sqrt{3bp/8})\,p(1+32p)}\right\}$$

where W_{max} is the limit on the maximum window size, T_0 is the retransmission timeout interval, *b* is the number of packets acknowledged by each ACK packet and *p* is the packet loss probability given that the previous packet is not lost and *RTT* the round trip time of the connection (assumed constant).

The above equation predicts the steady state throughput of a TCP connection with sufficient amount of data to transfer by modelling its congestion avoidance phase. However the majority of the flows are very short (mean sizes 10KB and median sizes less than 10KB) and they spent their short lifetime in the slow-start phase thus the Padhye equation is not appropriate for this type of flows. Cardwell et al [CARDW] recently combined models describing the performance of both short and long flows. Their model essentially describes the TCP latency as the expected time (*E[T]*) for completing the different phases of a TCP connection:

$$E[T] = E[Tss] + E[Tloss] + E[Tca] + E[Tdelack]$$

where $E[T_{ss}]$ is the expected latency for the initial slow start phase, $E[T_{loss}]$ is the expected delay from a fast recovery or a retransmission timeout at the end of a slow start, $E[T_{ca}]$ for the amount of data sent in congestion avoidance and $E[T_{delack}]$ for the amount of latency attributed to delayed acknowledgements. The exact expressions for the components are complex and the details are given in [CARDW].

There are also fixed-point models, which determine packet loss, link utilisation and TCP throughput across the entire network. The network is modelled as a set of resources (links) and the load across resources is assumed to be un-correlated. The resources are modelled as finite buffers and the input is a matrix of the mean number of sessions per route [GIBB2]. The fixed-point approach is promising because it takes into consideration both the entire network and the adaptive nature of TCP whose throughput is determined by the level of utilisation of the “weakest link” in a given route.

Dropping packets can be a waste of resources since the packet to be dropped may have already consumed enough bandwidth before it gets dropped. Therefore there have been proposals for *congestion marking* using Explicit Congestion Notification (ECN) so that the senders receive the congestion marks (copied in the ACK packets) and reduce their windows appropriately.

In theory the ECN marks could be used as price signals for managing congestion and TCP can be interpreted as a utility maximization algorithm, which balances the benefit a user gets from the achieved flow rate against the impact on other users signalled by the ECN marks.

There are different levels at which the marks can be aggregated to reflect costs or prices to the users, for instance an ISP may manage the risk associated with congestion pricing and offer service to its users defined in more traditional terms. However because the traffic may fluctuate widely this is likely to be reflected in the congestion charges and a method for providing an aggregate price may be necessary. Moreover a network owner has an incentive to increase congestion levels. There are recent efforts in designing contracts for usage, which the network owner (ISP) can sell to its customers. The structure of these contracts is such that it alleviates all the inappropriate incentives [ANDERS].

3.8.3 Overlay Networks

Another important area is that of Overlay Networks and IP based Virtual Private Networks (VPNs) in particular. VPNs are a special case of what is known as *overlay networks*; isolated virtual networks created over an existing network. They are composed of nodes (hosts or routers) and tunnels (paths on the underlying network that appear as links in the overlay).

A VPN is a network built over the shared IP infrastructure which operates with the security, management and possibly Quality of Service (QoS) policies of a private network. A VPN is a cost-effective means of building and deploying private communication networks for multi-site interconnection. The VPN service provider connects multiple IP addresses at geographically dispersed sites as appearing to be within the same private network.

Since IP has become common to corporate networks and applications there is a market opportunity for IP-based VPNs to provide connectivity between corporate sites and access to Application Service Providers (ASPs).

3.8.4 Peering and Multihoming

The simplicity of the best-effort service model allowed IP to be implemented over every possible link layer technology and allowed simple ISP interconnection. However this simplicity in ISP interconnection is in many cases responsible for causing congestion problems at the interconnection boundaries which result in poor performance.

An ISP interacts both with other ISPs and clients who purchase network service. The clients may be individual subscribers or corporate customers (like the ASP in

TAPAS) that rely on the ISP for connectivity. Thus the ISP can affect drastically the performance of their applications. There are several issues that all the clients (including the ASP) should consider when they decide on which ISP to use and the terms of their network SLAs.

First, the interactions between the ISP and its neighbour ISPs are particularly relevant to current/prospective ISP clients because the quality of service cannot be improved beyond the quality provided by the ISP's neighbours and their neighbours in turn. This happens because the ISPs cannot possibly connect to all the subnetworks that make up the Internet but they must be able to move traffic from one network to the other so that their clients can *reach any client on any other ISP network*. Clients would almost never buy contracts that limit connectivity to other clients on the same ISP. Sometimes they buy SLAs which provide guaranteed service to clients on the same ISP.

Another issue is that not all ISPs have the same network reach. There are Tier-1 ISPs with large backbone networks and access to the global Internet routing table (e.g C&W, Sprint, AT&T). Tier-2 ISPs, which have regional or national backbones and lease part of the routing table from Tier-1 ISPs (e.g UKERNA) and small regional providers without national backbones referred to as Tier-3 providers.

The current model of interconnection between ISPs is to a large degree ad-hoc, in general two connected ISPs can be engaged in a *peering* or a *client/provider* relationship.

Peering occurs when two ISPs exchange traffic bound for each other's network over a direct link without fees. Peering usually occurs at *public Network Access Points* (NAPs), which tend to be highly congested and through *private peering points* set up by the ISPs with high-speed links, which are generally less congested.

The larger ISPs prefer to peer with each other as they have similar networks and they exchange similar amounts of traffic. Smaller ISPs tend to peer with others of similar size and with large ISP at the public NAPs.

However because public NAPs are congested the smaller ISPs are forced to *buy transit* from large ISPs – by buying a link directly into their backbone.

It is well known that a transfer, which crosses congested public NAPs may take up to 10 times longer than if it had used well-engineered backbones and well-provisioned private links.

We show these relatyionships between ISPs n the Figure below where the arrows show the dependencies for service. The ASP is relying on the quality of the interconnections between its ISP and its neighbours in order to serve clients in their networks. All the ISPs peer with each other at the (congested) NAP, two ISPs peer privately with each other (bi-directional arrow) and the ISP buys transit from the larger ISP1 in order to be able to provide better service to (some) of its clients bypassing the congested NAP.

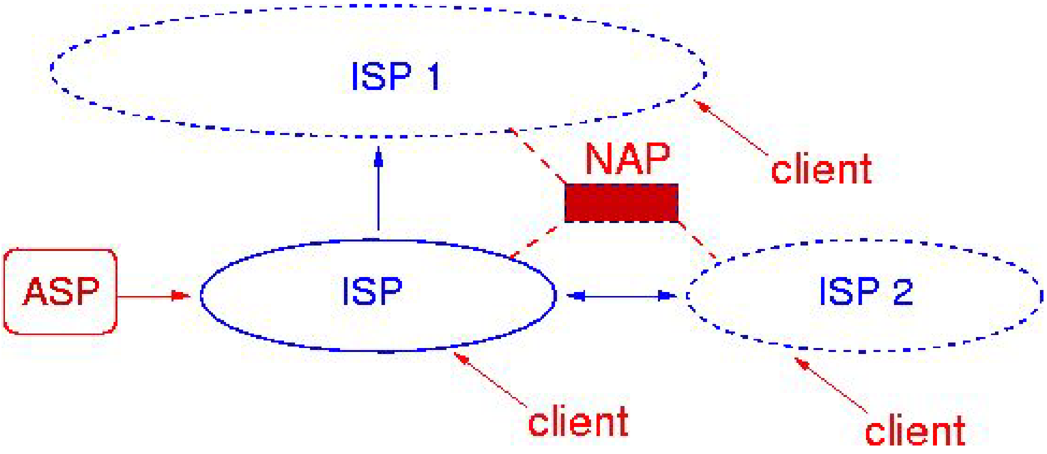
It follows that the type of interconnection of an ISP is very important to the ASP client. The quality of an ISP's interconnections can be described by

- *The number of peering agreements*, the greater the number of public and private peering agreements the better it is for connectivity, peering with Tier-1 ISPs certainly adds value,
- *The number of upstream providers* from which the ISP buys transit,
- *Information on interconnection policies*, this is essentially the SLAs between the ISP and its neighbours, it involves information about the capacity of the

links, the upgrade procedures (which level of utilisation triggers the upgrade), typical operating parameters (e.g utilisation, delay), redundancy on the links.

Nevertheless the ISPs are usually unwilling to disclose such information (even under a non-disclosure agreement) although some may do so.

In order to deal with this a client of an ISP will have to ask for multiple diversely routed connections from a single ISP so that it gets insurance over link failures. Another solution is to use multiple access links to the same or even different ISPs this is known as *multihoming* . The downside of multihoming is that it complicates routing for the ISP client while with a single ISP a default route upstream is sufficient. On the other hand multihoming may offer load balancing too and assume more control on how different destinations (e.g ASP clients) can be reached.



3.8.5 Transport level SLAs

The SLAs we have seen so far are always between ISPs. There is a group of ISP clients that primarily focuses on application level performance. In this group belongs the ASP whose goal is to meet the application performance requirements of its clients as well as individual *end-users* (ISP subscribers/clients). Obviously, with regards to application level performance, the current network SLA offerings in the industry are not comprehensive enough.

Currently almost the only differentiating factor offered to ISP clients with application performance requirements is the *bandwidth of the access link*. Less frequently and only a few ISPs, with the necessary technical expertise, offer some form of local prioritisation of traffic based on *classes of service (CoS)* defined on the access link. The prioritisation is based upon the type of traffic and the application performance requirements. For example AT&T Managed Internet Services offers four CoS: Real time, High-grade, Medium and Low-grade data classes.

Nevertheless, the influence that such a local traffic prioritisation scheme can have on the end-to-end performance is in many cases limited. Its effectiveness depends on where the bottleneck is.

In the case of access link speeds there are many offerings at different prices in the form of different modem speeds, ISDN lines, DSL, T1, T3, OC-3, OC-12, OC-48 etc. For modem users the bottleneck is usually the access line and one can do little to improve their performance, even a local prioritisation schemes would make no obvious difference.

Nevertheless there is an increasing population of end-users for whom the access link is no longer the bottleneck. There are several possibilities for enhancing the performance of these users by offering lower delays, (higher throughput) for their transfers.

In the Internet, application performance is jointly determined by the client's *transmission behaviour* and the characteristics of the network path (loss, delay) between the communicating end-points.

Packet loss can be due to noise in wireless links or to occasional buffer overflow in the routers, it is then called congestive loss. However, the latter does not mean that the network is necessarily congested. Congestion is the state of the network with persistent high packet loss rates, otherwise packet loss is the *feedback* mechanism which allows the adaptive sharing of the underlying bandwidth.

Reliable transport protocols are able to detect packet loss (using feedback control) and provide techniques for *loss concealment* removing this responsibility from the application developer. In broad terms, there are two techniques for loss concealment: i) retransmission and ii) forward error correction (FEC). Retransmission is being successfully used in TCP and it poses certain challenges in the case of reliable multicast transport (see the Section on PGM). FEC techniques require a fixed delay at the source to compute the redundancy, encoding etc.and tend to overload the network more but on the other hand provide lower delay compared to retransmission-based schemes.

The last few years there has been a lot of work on new transport mechanisms in order to address the requirements of different types of applications (recovery mechanisms for TCP with selective acknowledgments, new transport protocols like SCTP, protocols for reliable multicast transport like PGM etc.). Given that TCP is the de-facto standard transport protocol, the issue of “TCP-friendly” congestion control and fair bandwidth sharing has been central to these developments.

We would like to relax the fairness considerations and influence bandwidth sharing so that more aggressive transport protocols are available to those users with the willingness to pay extra for these services.

This can be achieved by using techniques such as packet level FEC [HUIT] or modified transport protocols for higher performance such as MulTCP [CO98]. These mechanisms give the client an *end-to-end competitive advantage* at congested points deep inside the network where neither the client nor the ISP could possibly have an SLA in place. Moreover they can be very useful to complex Internet business applications, which are transactional in nature (in contrast to web content distribution) and require fast response times.

The transport level SLAs are not mutually exclusive to the current IP level SLAs. IP level SLAs are certainly required but they have a well-defined local scope and specify the service in statistical terms (like the SLAs described already which specify percentiles of delay, availability or loss, but *not* bandwidth).

These transport level mechanisms operate in shorter timescales compared to the traffic engineering required for implementing IP level SLAs and their effect on service differentiation is not well understood.

The most important issue is to find a way so that the providers can make these transport options available to their clients and charge for their use. This is a complex issue because these mechanisms are traditionally implemented at the client's computer and thus they are outside the provider's control.

The monitoring/management and enforcement of transport level SLAs under these circumstances is difficult compared with the relatively simple IP level SLAs where loss, delay or the data transfer rate needs to be measured. In these cases what matters most is the behaviour (congestion/error control) of the transport protocol and this is hard to be inferred by an external observer. The reason for this is that these mechanisms are *adaptive* and their throughput may vary significantly depending on the conditions of the network path which almost certainly will span several ISPs.

In summary as far as the network SLAs are concerned we intend to evaluate all the options described above favouring clearly statistical guarantees in end-to-end performance achieved through a combination of end-point control mechanisms and network dimensioning.

4 References

[ASP] www.allaboutasp.org - ASP Industry Consortium White Papers
"*SLA for Application Service Provisioning*"

[EB] www.ebay.de - e-auction site

[VS] www.verisign.com - Digital trust services provider

[IL] www.ilog.com/industries/communications/ossj/ - Component developer and provider

[J2E] Java™ 2 Platform Enterprise Edition Specification, v1.3

[OMG] http://www.omg.org/ - OMG Homepage

[CCM] http://ditec.um.es/~dsevilla/ccm/- CORBA and CCM Homepage

[W3C] http://www.w3.org/ - World Wide Web Consortium

[OSS] http://java.sun.com/products/oss/ - OSS through Java Initiative
http://jcp.org/jsr/detail/90.jsp - Java Specification Request for OSS/J QoS API (JSR 90)

[AD] adesso contracts: *„SLA allgemein Rahmenvertrag für ASP-Betriebsdienstleistungen“*

[RA] Alessandro Amoroso, Fabio Panzieri. Extended abstract: "A Responsive Architecture for Distributed Auction Services Over the Internet"

[SAM99] Adrian Perrig, Sean Smith, Dawn Song, J. D. Tygar, UC Berkeley, Dartmouth College. "*SAM: A Flexible and Secure Auction Architecture Using Trusted Hardware"*

[EU00] http://www.acm.org/sigs/sigmm/MM2000/ep/keus/ *"Technical harmonised implementation of the European Directive for electronic Signatures"*

[TA87] S. White and L. Comerford. "*Abyss: A trusted architecture for software protection*"

[DIS01] Jan Schlüchter. Dissertation: „Prognose der künftigen Entwicklung elektronischer B2B Marktplätze“

[EP01] TAPAS Proposal, Part B.

[INTSERV] Integrated Services http://www.ietf.org/htl.charters/intserv-charter.html

[DIFSERV] Differentiated Services http://www.ietf.org/htl.charters/diffserv-charter.html

[UKERNA] UKERNA SLAs http://www.jisc-tau.ac.uk/ukerna-sla.html

Summary of the SuperJANET4 SLA
http://www.superjanet4.net/backbone_procurement/public_sla.pdf

[WCOM] SLA WorldCom IP VPN Dedicated Services Fully Managed
http://www1.worldcom.com/uunet/terms/sla/uk/ipvpn_dedicated_sla_UK.pdf

[CO98] Crowcroft J. and Oechslin P. "Differentiated end-to-end Internet services using a weighted fair sharing TCP", ACM Computer Communication Review.

[HUIT] C.Huitema "The Case for Packet level FEC" In Proc of IFIP 5th International Workshop on Protocols for High Speed Networks, Sophia Antipolis, France, October 1996.

[MPLA] Multi-Lateral Peering Agreement http://nap.aads.net/MLPA.html

D. Allen “The Impact of Peering on ISP Performance: What’s Best for You” Nov 5, 2001 http://www.networkmagazine.com/article/NMG20011102S0006/3

[QBONE] The Qbone http://qbone.internet2.edu/

[GIBB1] R. Gibbens et al “An approach to service level agreements for IP networks with differentiated services” http://www.statslab.cam.ac.uk/~richard/research/topics/royalsoc1999/

[GIBB2] R. Gibbens et al “Fixed Point models for the end-to-end performance analysis of IP networks”, In Proc of the 13th ITC Specialist seminar: IP Traffic Measurement Modeling and Management, Sep 2000, Monterey, CA.

[CARDW] N.Cardwell et al “Modelling TCP Latency” In Proc of the IEEE Infocom 2000 Conference, Tel Aviv, Israel March 2000.

[PADHYE] Padhye, J. et al “Modelling TCP Throughput: Asimple model and its empirical validation” In Proc. Of SIGCOMM’98

[ANDER] Anderson, E. Kelly F, Steinberg, R. “A contract and balancing mechanism for sharing capacity in a communication network” http://www.statslab.cam.ac.uk/~frank/aks.html

Appendix A

1 An ASP scenario: auctions in a B2B market place

In this chapter we describe a scenario for a marketplace as a B2B-application that is implemented by an ASP solution. Starting with this scenario, we focus on a particular part for auctions, which brings in networking-relevant aspects. Particular attention of our research is devoted to trust management in face of the security and privacy requirements of the parties with consideration to their conflicting interests.

1.1 A distributed, multi-party scenario

Auctions are an important economic mechanism which are widely used to sell a variety of commodities, such as treasury bills, mineral rights, real estate, art works, etc. With the growing popularity of the Internet, many traditional auctions are transforming into electronic auctions, and many new electronic auctions are being created. As a result, a number of web-based auction markets [AU] have emerged. These range from relatively public markets such as auctions run by e-Bay, Amazon, and Yahoo! to B2B auctions (freemarkets.com, commerceonce.com) to double auctions such as on-line stock markets. Compared to traditional auctions, electronic auctions have several advantages in that they are global in scope and may be less expensive than traditional auctions. Participants in electronic auctions require neither physical presence nor (for *off-line* auctions) a connected electronic presence [TA87].

From our industrial experience we can identify the following actors in our auction scenario:

- Buyers
- Sellers
- Auctioneer
- Application Service Provider (ASP)
- Internet Service Provider (ISP)
- Storage Service Provider (SSP)
- Trusted Third Party (TTP)
- Certification Service Provider (CSP)
- Other Service Providers: credit rating agency, retail banks etc

Such roles as buyers, sellers and auctioneer do exist in traditional auctions as well. Sellers place the goods at the auctioneer’s disposal. Buyers bid for an object they want to purchase according to the auction type – open or secret. The auctioneer starts the auction by setting an asking price for an item on sale, and requests bids. Periodically, he resets the asking price to the value of the highest bid received from the sellers during a certain timeframe, and starts a new round [RA]. The auction is terminated if there are no more bids for the item within a timeframe from the last valid bid. Buyers and sellers want to participate in a fair auction. So the auctioneer has to assure a secure procedure. Hence trust management is an issue of every auction.

In an electronic market place scenario there are additional actors such as the ASP that is hosting the market place applications. Furthermore there are other service providers, such as ISP and SSP, offering communication and storage resources. ASP undertakes a role of integrator in this scenario, because SLAs of different parties are focused on it. A CSP is responsible for the authentication mechanism and thereby increases the security and safety standards. Though the role is not traditional, we refer to the EU-Directive defining the Qualified Certificate (QS) and CSP requirements [EU00].

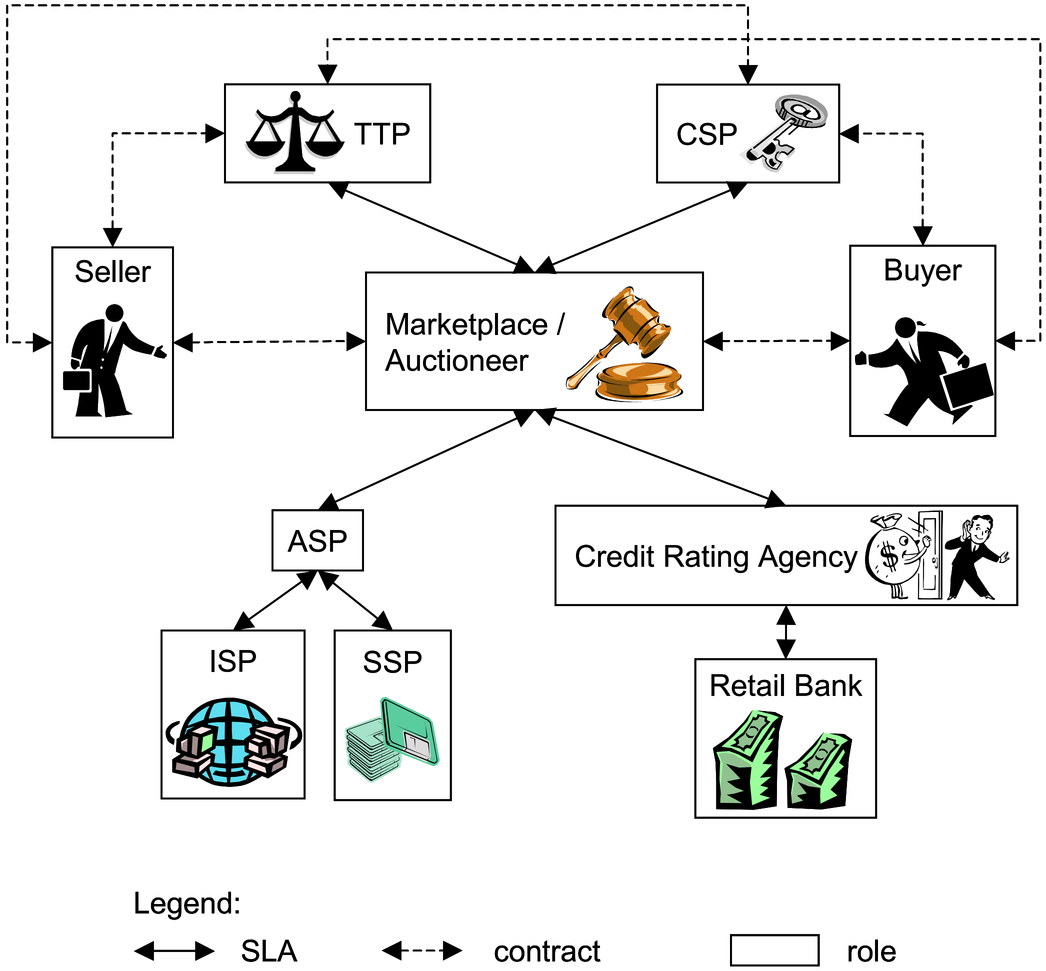
The role of the TTP is necessary in this scenario to eliminate the lack of trust, which some market partners may have. The TTP acts as a kind of independent referee interacting between the distrusted parties. Besides fraud prevention, the function of TTP would be to supervise the SLA fulfilment as well as authentic reporting on the behaviour of the service providers.

Service providers like credit rating agencies, retail banks or other service providers are very important for a full-featured scenario. The auction application sends requests for proposals (RFP) to such a provider’s component, asking the credit rating agency for information. The agency itself implements its services using data obtained from account history services provided by retail banks. Thereby we have a chain of calls between subsequent service providers.

Regarding the service providers it must be said that only a certain level of standardization offers real benefit to the ASP model. Service providers can easily be exchanged if switching from one service provider to another does not lead to further interface programming. For example, credit rating services are not yet unified throughout the European Community. Hence, usage of different service providers for e.g. customers from different European countries would result in the need of creating interfaces to all of those. This situation might become simpler in future if Web Services initiatives become more successful.

We can assume that all these parties are bound to SLAs regarding security, privacy, responsiveness and measurable quality of the underlying network. Some of them depend on each other, for example the availability provided by ASP will depend on the bandwidth characteristic supported by ISP.

The figure below mirrors our auction scenario from the hosting point of view:



Commentary: SLA is a subject of the hosting contract per contra a universal contract

This auction scenario and thereby the market place scenario is distributed in the way that all actors will probably maintain their own IT-infrastructure such as desktop boxes for the buyers and sellers or backend host systems for the credit rating application. Regarding the ASP itself, however, there may be another kind of distribution involved: usually reliability and safety are ensured by clustering servers not only for databases but nowadays as well for application servers. Though this clustering is quite important, distribution in TAPAS refers to distribution between the sites of the participants.

1.2 Trust Management

Implementing auctions in the setting of distributed computing is complicated by several fundamental properties:

- Auctions involve *multiple parties*, such as the auctioneer, buyers, sellers — and possibly other stakeholders, such as government regulatory agencies.
- These parties have *conflicting interests*.
- Auctions involve *private information*, such as bids, bidding strategies — and possibly fraud patterns.

- Auctions involve *computation* on this information, such as execution of the auction, decisions on bids, recognition and suppression of fraud.

In a distributed setting, these properties create a fundamental trust challenge: we need to distribute this information and computation among the parties themselves, in way such that the computation is still correct, and all involved parties can still trust that their respective interests are preserved [SAM99].

As to trust management, a minimal concept is currently established. The procedure looks like this: the customer has to register himself, and then they can act as seller or bidder. The authentication results by indicating their email address, postal address, telephone number or credit card number.

Electronic auction can define email address as untrustworthy if it is an anonymous one, like from *gmx* or *yahoo.* In this case the customer has to give his credit card number or to use a special code sent to his post address. Otherwise he may be identified by his telephone number through a TTP, for example the Authentication Service Bureau by VeriSign [VS]. The verification by a postal clerk (Post Ident) seems more secure as by telephone call, but not very convenient.

The most secure way to identify a person is to use CSP and to get a QC. A QC is a certificate defined according to the EU-Directive for electronic signatures [EU00]. If auctions use QC for user authentication, they have to process an electronic signature validation.

All this options are more or less secure in face of fraud detection and have to be examined exactly. These options have varying degrees of security in the face of fraud and need to be examined in detail.

An auction site like eBay [EB] gives an opportunity to draw on a trust organisation for assuring a buyer-seller relationship. In this context, a clearinghouse functions as a TTP in that it carries out all transfers between the market partners and guarantees the availability of the object that is being handled.

Furthermore, the seller as well as the bidder can be assessed through the rating points given by the participants. If the seller is trustworthy, he will most likely get the maximum points. However, this depends on the judgement of each of the buyers, so after several auctions, a high number of points signals trustworthy behaviour. There is also a possibility for customer to have an agent sell goods and services on contract. In this sense an agent is generally considered to be a TTP. The agent is an experienced seller, having gathered enough trust points over a certain period of time. This method has disadvantages in that the assessment is subjective and that it takes place after the deal.

In view of the current electronic auction practice we deduce that this approach to trust management should be improved.