An Agent-Based Framework for Monitoring Service Contracts

Helmut Kneer[1], Henrik Stormer[1], Harald Häuschen[1], and Burkhard Stiller[2]

[1] Department of Information Technology, IFI, University of Zurich, Switzerland
{kneer, stormer, haeusche}@ifi.unizh.ch
[2] University of Federal Armed Forces Munich, IIS, Germany and ETH Zürich, TIK, Switzerland
stiller@tik.ee.ethz.ch

Abstract. Within the past few years, the variety of real-time multimedia streaming services on the Internet has grown steadily. Performance of streaming services is very sensitive to traffic congestion and results very often in poor service quality on today's best effort Internet. Reasons include the lack of any traffic prioritization mechanisms on the network level and its dependence on the cooperation of several Internet Service Providers and their reliable transmission of data packets. Therefore, service differentiation and its reliable delivery must be enforced on a business level through the introduction of service contracts between service providers and their customers. However, compliance with such service contracts is the crucial point that decides about successful improvement of the service delivery process. For that reason, an agent-based monitoring framework has been developed and introduced enabling the use of mobile agents to monitor compliance with contractual agreements between service providers and service customers. This framework describes the setup and the functionality of different kinds of mobile agents that allow monitoring of service contracts across domains of multiple service providers.

1 Introduction

The recent development on the E-Commerce services market shows an upcoming demand for streaming technology in form of video-conferencing, video-, audio-, or news-on-demand applications. Besides the time and resources that long and expensive downloads entail, streaming technology offers a cheap solution to widely broadcast real-time data making high demands on systems and networks. Streaming means that continuous data is cut into single units (packets) and subsequently sent from sender to receiver. The sequence of single packets is called stream [19].

The global Internet is considered a network of networks where the networks or domains of several Internet Service Providers (ISP) are interconnected with each other to span a network around the globe. Communication within such a network requires a standardized protocol, the Internet protocol [17], [5] allowing different domains to exchange data. In order to achieve global connectivity for every single host on the Internet, every autonomous domain needs to route data within its scope (intra-domain routing) but also between other neighboring domains (inter-domain routing).

Fig. 1 shows a general network model of a streaming scenario where end-customers receive data (e.g., streamed video data) from Content Service Providers

K. Bauknecht, A M. Tjoa, G. Quirchmayr (Eds.): EC-Web 2002, LNCS 2455, pp. 139–151, 2002.

(CSP) via several interconnected ISP networks [13]. The network model shows the integration of several business entities into the process of service provisioning. It is not only the cooperation of these entities that influences the service delivery but also the reliable transport service and other value-added services [12] provided by ISPs.

Unfortunately, current Internet technology is based on a packet-switched network where forwarding of data packets relies on a best effort service without any service guarantees and service prioritization mechanisms. Reliable and fast routing of data through the different networks underlies strict performance restrictions as far as data throughput, packet delay, loss and error rate among others are concerned.

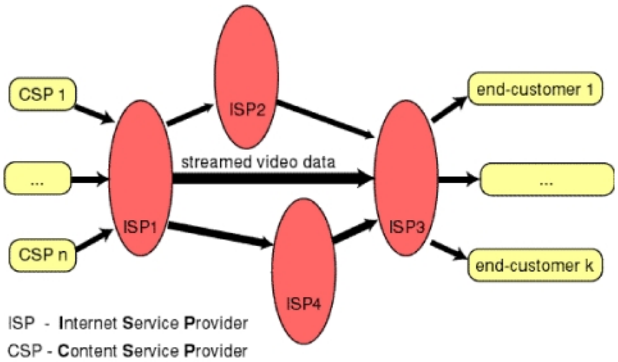

Fig. 1. General network model in a multi-provider environment

In order to offer and receive satisfying real-time Internet services, an efficient and economic resource management on the part of the service providers (CSPs and ISPs) is most urgently required. We claim that the introduction and monitoring of service contracts between service providers and service customers is one way to improve streaming real-time service provisioning. The usage of mobile agents for monitoring purposes provides the opportunity to control compliance with the guaranteed service levels across ISP domains. The agent-based monitoring framework describes the tasks of monitoring agents and their cooperation in order to identify malfunctioning service providers that cause service shortcomings or service failure.

The paper is structured as follows. The next section provides technical background for the usage of service contracts and mobile agents for monitoring purposes before section 3 introduces a contracting protocol for the service level negotiation. Section 4 describes the agent-based monitoring framework with the setup of agents and monitoring tasks. Section 5 describes implementation issues for the framework and demonstrates a practical example of how the agent-based framework could be applied in a real world scenario. Finally, section 6 summarizes the paper, draws conclusions, and gives an outlook on future work.

2 Technical Background

This section provides scientific background for the understanding of service contracts in form of Service Level Agreements (SLA) and Operational Level Agreements (OLA) and the usage of mobile agents for monitoring purposes.

2.1 SLAs and OLAs for Streaming Internet Services

While the demand for high-bandwidth multimedia E-Commerce services (e.g. streaming Internet services) is increasing steadily, the expectations for the service quality is growing in just the same measure. This means for a big E-Commerce services market that service provisioning on today’s Internet is very exacting and challenging for service providers in order to meet all the requirements as far as the Quality of Service (QoS) is concerned. These QoS expectations will drive service customers more and more to negotiate service parameters with their service providers.

The concept of Service Level Agreements is fairly well understood and widely used, especially for the Service Level Management (SLM) within an enterprise [16]. A SLA defines a contract between a service provider and its customer, which are considered in a broad sense [7]. While its dedicated content may vary according to the service being offered, the general service description and additional parameters are required. Those include the customer and provider identification, a service definition, service monitoring and control information, pricing-, charging-, and business-related information as well as legal agreements [10]. SLAs for the Swiss market offered by UUNET guarantee 100% network and service availability, network latency, and outage reporting or they guarantee Virtual Private Network security [20].

Transferring the concept of SLAs and contractual agreements onto our network model of Fig. 1, we can distinguish two layers for applying the concept. The top layer entails a contractual agreement between the CSP and the end-customer while the bottom layer combines contracts among the ISPs for the data transport service. The bottom layer agreements are underpinning contracts and support the top layer to operate and fulfill the service between CSP and end-customer. Hence, they are called Operational Level Agreements. Negotiating service parameters (e.g., delay, error rate, jitter) and stipulating service contracts will help service providers to manage their network resources efficiently and economically and will improve customer satisfaction for QoS-based end-to-end service provisioning.

The Differentiated Services Internet (DiffServ) Architecture [1] uses SLAs to describe a “service contract … that specifies the forwarding service a customer should receive”. The detailed technical parameter specification is part of a second container, which, in case of DiffServ, is termed Service Level Specification (SLS). It mainly includes those parameter specifications, which cover values and traffic data within a dedicated DiffServ domain. The Traffic Conditioning Specification (TCS) forms an inherent part of a SLS and defines the detailed classifier rules and traffic profile values [7]. Therefore, a SLS can be considered a DiffServ OLA.

2.2 Mobile Agents for Inter-Domain Monitoring

The term ‘agent’ is used by a number of different research communities, for instance within the fields of artificial intelligence (intelligent agents) [15] or software engineering and distributed systems (software agents) [21]. The Object Management Group (OMG) defines a software agent as “a computer program that acts autonomously on behalf of a person or organization” [4]. Additionally, agents are often characterized using the following properties:

- proactive (support of the user's work)
- adaptive (learning the user's preferences or the ability to work on different platforms)

- autonomous (not communicating with its creator)
- intelligent (making ‘intelligent’ decisions [6])
- mobile (can actively migrate within networks to different systems and move directly to the local resources, like databases or application servers).

Most of the agents introduced in this paper are mobile. Therefore, each system needs to install a so-called ‘agent-place’ to create, delete and execute agents. Then, agents can migrate from system to system and perform their tasks locally.

2.3 Motivation for Using Mobile Agents

Monitoring of SLAs/OLAs is indispensable for an E-Commerce scenario where the end-customer pays money for guaranteed service provisioning and where the process of service delivery is dependent on several independent service providers (CSP and ISPs). The question to be discussed is why to use mobile agents for performing monitoring tasks.

Lange and Oshima give seven reasons why to use mobile agents [14], which can be favorably applied for the given example of a network model as illustrated in Fig. 1. For example, there is the reduced network load if agents can process and evaluate monitoring data locally instead of being transmitted over the network. The use of mobile agents can also overcome network latency, especially in case of real-time applications like streaming Internet services where quick actions are required in case of service failure or shortcomings. Furthermore, mobile agents can adapt dynamically to changing environments and execute asynchronously and autonomously since they do not require a continuously open connection to a fixed network. And finally, mobile agents can encapsulate protocols by wrapping monitoring information and communicating it based on proprietary protocols.

Lange and Oshima have also defined some applications that benefit from using mobile agents. Included is “monitoring and notification”, where the main advantages are the asynchronous nature of the monitoring agents as well as the ability to monitor any given information source without being dependent on the system the information originates from. Providing an agent-place within the service providers’ networks allows the agents to have access to monitoring data in a controlled fashion.

3 Service Level Negotiation

This section introduces one approach how to integrate SLAs/OLAs into the network model of Fig. 1. Several bilateral contracts in form of SLAs and OLAs are required between the entities to enable a combined real-time streaming service between CSP and end-customer. The contracts are negotiated in a contracting process prior to the actual service delivery process. Therefore, a contracting protocol based on the exchange of XML documents was developed to implement the different steps of the contracting process (see also [10]). The single steps of the contracting protocol are illustrated in Fig. 2, which shows a simplified network structure compared to Fig. 1 with only one CSP, a single end-customer, two ISPs and a Service Broker (SB) embedded into the network model.

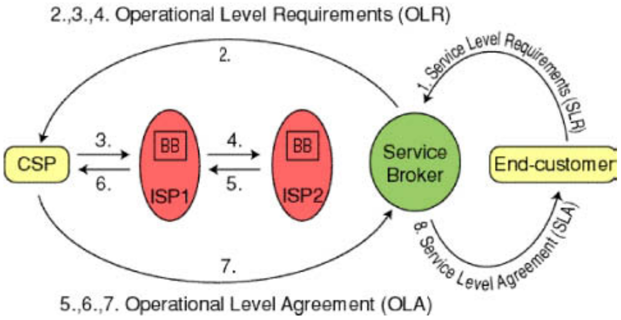
2.,3.,4. Operational Level Requirements (OLR)
2.
1. Service Level Requirements (SLR)
CSP
3.
6.
BB
ISP1
4.
5.
BB
ISP2
Service Broker
End-customer
7.
8. Service Level Agreement (SLA)
5.,6.,7. Operational Level Agreement (OLA)

Fig. 2. Contracting Protocol with a Service Broker to stipulate service contracts

The SB works as a negotiator between end-customer and CSP. It represents the point of contact for end-customers if they want to request and negotiate a service with certain QoS requirements. The SB can be an independent business entity but it is also possible that an ISP for example inherits that role.

In a first step, the end-customer as the receiver of the service (e.g., a video stream) defines Service Level Requirements (SLR) as the service parameters that define the QoS. After checking its own resources, the SB transforms the SLR into Operational Level Requirements (OLR), a technical translation of the informal SLR. The SB requests the OLR from the CSP (step 2), who checks its own resources before forwarding the request to ISP1 (step 3). ISP1 checks its resources with the help of a network manager (in Fig. 2, this role is taken by a Bandwidth Broker (BB)) and if resources available forwards it on to ISP2 (step 4). This process continues until the OLR reaches the last ISP in line that could deliver the service to the SB (in Fig. 2, this is ISP2). Only if the last ISP in line agrees to the OLR, the reverse process of creating OLAs can be initiated. ISP2 signs the OLR to make an OLA out of it before returning it to ISP1 (step 5). This reverse process continues until the SB receives an OLA from the CSP (step 7). Only now, the SB can react to the end-customer’s SLR with a signed SLA and the guarantee that the service will be delivered according to the pre-defined service parameters.

If an ISP cannot meet the specifications of the OLR, it rejects the OLR and either an alternative route needs to be found or the SLR needs to be redefined by the end-customer to fit the service conditions of all the ISPs [11]. Notice, that no OLA is required between ISP2 and the SB, since physical interconnection between the two entities must already exist on a network level and also, ISP2 is the last ISP in line to agree to the OLR originally requested by the SB.

The SB collects (flexible and temporally limited) service offers from CSPs and service requests from end-customers and tries to match them in an optimal way. The SB could even request resources from a CSP for a service that it is going to sell to end-customers in the future. Service requests can eventually be reformulated and resubmitted to the customer in case of major changes in price or QoS. In case several end-customers in a certain neighborhood request the same service (e.g., 100 m Olympic final) with slightly different QoS parameters, the SB can gather single requests in compound OLAs with the service providers. SLAs represent digital goods that can be traded on a marketplace [10].

4 Agent-Based Monitoring Framework

As illustrated in the previous section, the set up of several bilateral contracts in form of SLA/OLAs coordinates the cooperation of several business entities to provision a real-time streaming service to the end-customer. It involves every business entity and ties them to an official document that service customers can rely on if problems occur. Especially the service broker tries to represent a trustworthy entity to its end-customers. In order to maintain trustability and legitimacy of SLAs/OLAs, the existence of a monitoring system to observe compliance with the contracts is indispensable. Therefore, an agent-based monitoring framework has been developed to realize inter-domain monitoring across autonomous ISP domains.

4.1 Setup of the Monitoring Agents

The setup of the agents within the network of the different service providers and the service broker follows a four-step agent setup protocol, as illustrated in Fig. 3. The protocol is independent from the contracting protocol where the SLA/OLAs are negotiated. The negotiation of service contracts can be performed for an immediate service or service reservations can be made in advance where the end-customer reserves a service that will be delivered in the future (e.g., watching an Olympic hurdle race at some point in the future). If the service is provided immediately, the agent setup protocol is executed following the successful termination of the contracting protocol. In case the service is delivered in the future, the agent setup is triggered accordingly by the service broker at a reasonable time before the service.

Within the monitoring framework, we distinguish three different kinds of monitoring agents at different locations. There is a SB-agent, a CSP-agent, and possibly several ISP-agents with the following functionalities:

SB-agent:

- Monitoring of network communication and traffic parameters by collecting monitoring data delivered from the monitoring devices according to the specified filter
- Supervision of overall service provisioning
- Notification and taking steps in case of service failure or shortcomings
- Resides at the service broker

CSP-agent:

- Monitoring of network communication and traffic parameters by collecting monitoring data delivered from the monitoring devices according to the specified filter
- Eventually, monitoring of system components and hardware devices that are responsible for transmitting the content (e.g., CPU, operating system, memory, etc.)
- Eventually, real-time evaluation of collected data
- Resides at the CSP

ISP-agent:

- Monitoring of network communication and traffic parameters by collecting monitoring data delivered from the monitoring devices according to the specified filter
- Eventually, real-time evaluation of collected data
- Resides at the ISP

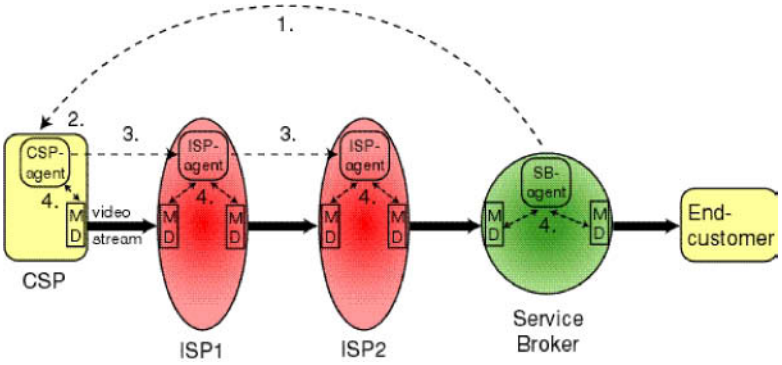

Fig. 3. The setup of the agents within the network of service providers

The sequence of steps to setup and place the different agents within the monitoring framework can be executed as follows:

1. In time, before the service is to be monitored, the service broker triggers the setup of the agents by creating the SB-agent. The SB-agent creates a CSP-agent that migrates to the CSP's agent platform.
2. The CSP-agent creates an ISP-agent. The CSP-agent provides the ISP-agent with the necessary information about location of ISP1 in order for the ISP-agent to migrate there.
3. The created ISP-agent migrates to ISP1. Having arrived there at ISP1's agent platform, the ISP-agent generates a copy of itself (also referred to as clone) which becomes the future ISP-agent of ISP2. Similarly to step 2, the clone receives the necessary information about ISP2 and migrates there. This procedure of "copying and migration" repeats until all the ISPs along the communication path that contribute to the service provisioning process have their ISP-agents. Information about the neighboring ISPs had been exchanged and stipulated by the bandwidth brokers (BB) in the previously executed contracting protocol.
4. The final step is to connect the ISP-agents as well as CSP- and SB-agent with the local monitoring devices (MD) through services and interfaces provided by the respective agent platform. Such monitoring devices could be agents as described in [8] or commercial monitoring tools provided for example by NeTraMet [2], Netflow [3], or IUM [9]. As illustrated in Fig. 3, we assume that monitoring devices consists of two devices that monitor both incoming and outgoing traffic at the ingress and egress routers of the ISP network. This has the advantage that all incoming traffic from a particular neighboring ISP can be monitored at the ingress router and compared with the outgoing traffic through the egress router. However, another option could be to place the monitoring devices at different and distributed locations within the network where traffic is reduced and monitoring can be performed more specifically.

After the agents have migrated to the different platforms and the connection is setup to the local monitoring devices, monitoring can be performed during service provisioning according to the QoS specification as stipulated in the SLA/OLAs. After the service has been finished and no monitoring is required anymore, the agents will be deleted.

The CSP-agent and the ISP-agents are local agents that are autonomous but work on behalf of the service broker. That means there is a fixed correlation between the broker agent and CSP/ISP-agents. However, other service brokers also have their broker-agents, which also generate their own CSP/ISP-agents that migrate to CSP/ISP networks accordingly. The monitoring devices are installed locally within the networks and systems of SB/CSP/ISPs and their task is it to provide the SB/CSP/ISP-agents with the relevant monitoring data after successful connection. That means the local monitoring devices provide several local agents with monitoring data. Monitoring devices scan the IP traffic by simply filtering out the information that is relevant to the corresponding agent. The filter is communicated to the monitoring devices during the connection process with the SB/CSP/ISP-agent. Filter functions are provided through specifications on the part of the local agent platform (see Section 5).

The service broker as an intermediary between CSP and end-customer might cause a bundling of several single requests of end-customers to one compound request to the CSP. In such a case, one pair of monitoring devices does not have to monitor several single services but just the one bundled service. Still, the service specifications for the bundle of single services have to be correct and monitored accordingly. After receiving the bundle of data, it is the task of the service broker to provide the different end-customers with the corresponding data. For that case, the service broker has its own monitoring devices to monitor incoming traffic (in form of a bundle) and also traffic that goes to every single end-customer (end-customer oriented monitoring).

4.2 Monitoring the Service Performance

Monitoring in general is applied in different areas (e.g., workflow applications, banking systems, web content) where monitoring is performed in form of data logging and reporting and simply serves for documentation purposes. However, within the agent-based monitoring framework, monitoring data is collected by SB/CSP/ISP-agents in order to evaluate the service delivery process, and as a consequence to take steps in case of service failure or shortcomings as defined in the SLA/OLAs. Monitoring can be subdivided into **static monitoring** and **dynamic monitoring**. Static monitoring implies a service evaluation after the service is performed, while dynamic monitoring foresees an examination of collected monitoring data ‘on-the-fly’ while the service delivery process is being executed. This is of prime importance if the corresponding service provider can be notified already as a preventive measure in case of service shortcomings and thus eventually avoid a complete service failure.

The MDs scan IP traffic by filtering relevant data for the corresponding agent and service with respect to service parameters such as delay, bandwidth, packet loss, etc. With static monitoring, an evaluation of the monitoring data is performed by the SB after the service delivery and after the CSP/ISP-agents have migrated back to the SB with the monitoring data attached. Eventually, the CSP/ISP-agents undertake a pre-evaluation of the results in order to reduce the amount of data before migrating back to the SB. With dynamic monitoring, evaluation of monitoring data is performed constantly by all the monitoring agents and therefore, the CSP/ISP-agents maintain a communication path to the SB-agent and exchange ‘on-the-fly’ evaluated monitoring data concerning the service level. In case of service shortcomings, the local BB can additionally be notified (preventive measure to avoid service failure). That means a communication path between the CSP/ISP-agents and the SB-agent exists for exchanging monitoring information.

With static monitoring, consequences in case of non-compliance with the service contract can be taken after the evaluation of all monitoring data. With dynamic monitoring, steps in case of service failures can be taken immediately when discovered. This could result in an exchange of a malfunctioning ISP 'on-the-fly' during service delivery. An overview of differences between static and dynamic monitoring is given in the following table.

	Static monitoring	**Dynamic monitoring**
Functionality	Evaluation of monitoring data is performed by the SB after the service delivery	Evaluation of monitoring data is performed by all monitoring agents during service delivery
Communication	There exists no communication among the SB, CSP, and ISPs	A communication path between the CSP/ISP-agents and SB-agent exists
Type	Reactive monitoring	Preventive monitoring
Consequences	Delayed consequences after evaluation of monitoring data	Immediate consequences possible

5 Implementation and Example

The agent-based monitoring framework is designed with one standardized agent platform. The platform is installed at the SB, CSP, and ISPs and provides a minimal set of standardized functions as described in the following. Any extension of monitoring functionality is achieved by extending the corresponding agent's functionality. This keeps the agent platform simple and small, supports a quick diffusion of the platform and guarantees a better reliability. The agent platform consists of three areas: In/Out Area, Working Area and MD Connection Area (see also Fig. 4).

As described in section 4.1, agents migrate from SB to CSP and on to different ISPs. For security reasons, the agents' source code will be encrypted and digitally signed by the SB before they migrate. Therefore it is impossible to change or to forge the agent while it is transferred from one provider to the other. After arriving in the In/Out Area the system decrypts the agent, checks its integrity, and verifies the authorization. The authorization is verified with the aid of a digital signature. After the agent's successful authorization, it can proceed to the Working Area. In the Working Area the agent can execute all implemented functions and communicate with the MDs with the aid of an interface provided by the MD Connection Area. The MD Connection Area is different for every provider because not all the MDs are standardized and the framework is MD-independent. Therefore, we do not describe this area in more detail at this point.

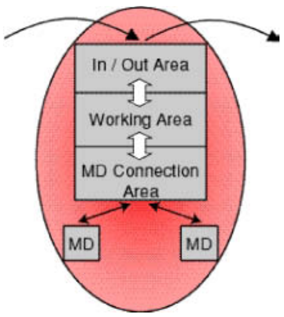

Fig. 4. The agent platform

5.1 In/Out Area

The In/Out Area performs the following security checks on the mobile agents. The checks are implemented with standard security functions like public key encryption, message authentication code and digital signature [18].

- Decryption of agents and information
- Integrity check
- Verification of Authorization
- Encryption and transfer to the next platform

5.2 Working Area

The Working Area serves to execute agents and to provide access to the MDs through the MD Connection Area. The agents are executed independently from each other and the area guaranties that an agent cannot change other agents and can only use the allocated resources. The Working Area must support the following functions:

- **OpenMD**: This function opens a connection to the MD (of either the ingress or egress router) and allows to monitor incoming or outgoing traffic related to the corresponding SLA/OLA.
- **RegisterFilter**: The agent can define an unlimited number of filters and use them for monitoring a stream (specified by source and destination address). To create such a filter for a certain stream the agent can define the following parameters among other things:
  - *size*: This parameter defines the size of the captured data.
  - *number*: This parameter defines how often a filter is executed.
  - *interval*: The interval defines the delay between two filter accesses.

  RegisterFilter(size=500, number=10, interval=2) means this filter is executed 10 times every two minutes where 500 bytes are sent to the agent.
- **StartFilter**: With help of this function the agent can start a monitoring process with a specific filter. The function requires a reference to an already defined filter in order to open connection to a MD.
- **StopFilter**: The monitoring with a specific filter will be stopped. The function requires a reference to an already activated filter.

- **RegisterEvent**: This function allows to define an event (e.g., MD failure). An event allows to start or stop a filter when a specific event occurs. The function requires an event, a reference to an already defined filter and open connection to a MD. The agent can define an unlimited number of events.
- **StartEvent**: This function activates an already registered event. After the event occurs the agent will be notified and monitoring with the specific filter will be started or stopped.
- **StopEvent**: This function deactivates an already activated event.
- **DeleteFilter**: An already defined filter will be deleted.
- **DeleteEvent**: An already defined event will be deleted.

All the filters and events are stored in a database. After the agent is deleted all the filters and events will be automatically deleted, too.

5.3 Practical Example

This practical example illustrates the applicability of the agent-based monitoring framework. Given a scenario with an end-customer connected to a CSP where the connecting path is routed across ISP1, ISP2, and a service broker (see Fig. 5). The end-customer requests a video stream from 09 to 11 pm with specific information about some QoS parameters such as error rate, delay time, jitter, etc. These parameters are negotiated between the end-customer, the service broker, and the service providers and subsequently stipulated within one SLA and three OLAs (see Fig. 5).

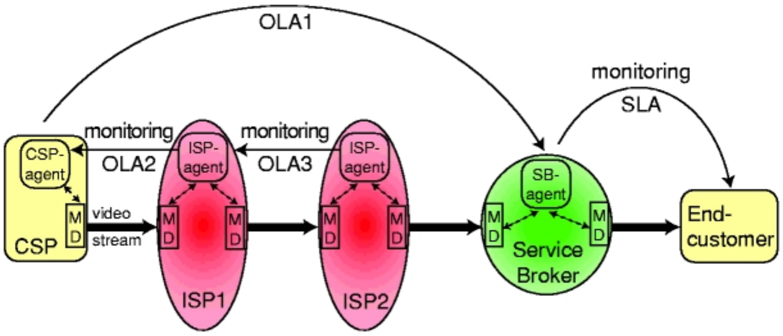

Fig. 5. Setup of the different agents to monitor the SLA/OLAs

Before the service provisioning process is executed, the service broker initiates the setup of the different monitoring agents for monitoring compliance with the contracts. Therefore, a SB-agent is created and positioned in the working area of the service broker. The SB-agent creates a CSP-agent that migrates to the CSP's working area. Afterwards, the CSP-agent creates an ISP-agent that migrates to ISP1. Finally, the ISP-agent of ISP1 generates a clone of itself that migrates to ISP2. Now, the different agents open connections with the corresponding local monitoring devices (MD) as described in section 5. By specifying filters, the agents can request from their MDs the kind of information they are interested in. By the time service provisioning starts, the agents are ready to perform their monitoring task.

Since streaming is very sensitive to the transmission of subsequent data packets, we choose delay as one QoS parameter to be monitored within this example. Therefore, the ISP-agents open a connection to their ingress and egress MDs and collect monitoring information about incoming and outgoing packets as defined in the monitoring filter. Depending on the filter specification monitoring can be performed permanently during the entire service duration or just temporarily within defined time intervals. The duration of the monitoring should be defined within the SLA/OLAs as well as the sanctions in case of service failure or service shortcomings. If equipped with the right functionality, the ISP-agents could calculate the delay time for the monitored data packets and pre-select and store only the kind of data that is not in compliance with the OLA. This information can be used by the service broker to take measures against the malfunctioned ISP.

If service shortcomings or service failure is noticed by ISP-agent of ISP1 and static monitoring is performed, sanctions are executed by the SB-agent after the service is finished. Possible sanctions could include a surcharge for the malfunctioned service provider. If dynamic monitoring is performed, steps can be taken during service provisioning such that the network manager of the corresponding ISP is notified about the shortcomings. If the situation is not improved, further steps could include for example the replacement of an ISP by choosing a new route with a different ISP.

6 Summary, Conclusions, and Future Work

This paper has presented an agent-based framework for monitoring service contracts. Agents provide useful functionality that was taken advantage of within the framework in order to monitor service compliance with SLAs/OLAs. The framework is based on assumptions about network infrastructure and resource management (e.g., resource reservation, differentiated services, QoS-based transport service) that are currently not fully applied by service providers. However, these assumptions are advantageous and will eventually be deployed for a QoS-based service provisioning and provide the basis for the stipulation of service contracts and their compliance with the real service delivery.

Advantages of the monitoring framework comprise its generic usability for monitoring all kinds of different services and parameters (e.g., delay time, error rate, correct order of data packets), its dependence on a number of small and cooperative agents, and its customer-friendliness due to monitoring based on customer requirements. Furthermore, monitoring was classified into static and dynamic monitoring such that static monitoring evaluates monitoring data after the service delivery while dynamic monitoring offers possibilities to evaluate monitoring data ‘on-the-fly’ during the service performance.

Future work in this area will deal with the detailed specification of different interfaces for special filter functions that establish communications between the monitoring agents and the local monitoring devices (MD). Most importantly, the developed and presented approach is based on a standardized format for service contracts and, therefore, future attempts have to address the definition of special SLAs and OLAs that allow automated monitoring activities based on the content of service contracts.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss: *An Architecture for Differentiated Services;* IETF, Request for Comments RFC 2475, December 1998.

[2] N. Brownlee: *NeTraMet (Network Traffic Meter);* http://www2.auckland.ac.nz/net/Accounting/ntm.Release.note.html, February 2001.

[3] Cisco: *IOS Netflow*; http://www.cisco.com/warp/public/732/Tech/netflow/, February 2002.

[4] Crystaliz, Inc., General Magic, Inc., GMD Focus, IBM Corp.: *Mobile Agent Facility Specification*; Technical Report, Object Management Group (OMG), 1997.

[5] S. Deering, R. Hinden: *Internet Protocol, Version 6 (IPv6) - Specification*; IETF, Request for Comments RFC 2460, December 1998.

[6] J. Ferber: *Multi-Agent Systems: An Introduction to Artificial Intelligence;* Addison Wesley Publishing Company, 1999.

[7] D. Grossman: *New Terminology and Clarifications for Diffserv*; Internet Draft, draft-ietf-diffserv-new-terms-08.txt, January 2002.

[8] M. Günter, T. Braun: *Internet Service Delivery Control with Mobile Code*; IFIP Conference on Intelligence in Networks (SmartNet 2000), Vienna, Austria, September 18-22, 2000.

[9] Hewlett-Packard: *IUM (Internet Usage Manager)*; http://www.hp.com/communications/usage/ium/index.html, February 2002.

[10] H. Kneer, H. Häuschen, K. Bauknecht: *Tradable Service Level Agreements to Manage Network Resources for Streaming Internet Services*; 10th European Conference on Information Systems (ECIS 2002), Gdansk, Poland, June 6-8, 2002.

[11] H. Kneer, R. Marfurt: *Contracting Protocol for Managing Quality of Service in a Multi-Provider Environment;* Submitted to IADIS International Conference WWW/Internet, Lisbon, Portugal, 2002.

[12] H. Kneer, U. Zurfluh, B. Stiller: *A Model for Value-added Internet Service Provisioning*; First IFIP Conference on E-Commerce, E-Business, and E-Government, Zurich, Switzerland, October 3-5, 2001.

[13] H. Kneer, U. Zurfluh, G. Dermler, and B. Stiller: *A Business Model for Charging and Accounting of Internet Services*; International Conference on Electronic Commerce and Web Technologies (EC-Web 2000), Greenwich, U.K., September 4-6, 2000.

[14] D.B. Lange, M. Oshima: *Seven Good Reasons for Mobile Agents*; Communications of the ACM, Vol. 42, No. 3, pages 88-89, March 1999.

[15] R. Murch, T. Johnson: *Intelligent Software Agents*; Prentice Hall, 1999.

[16] J. Niessen, P. Oldenburg: *Service Level Management - Customer Focused*; The Stationary Office, Central Computer and Telecommunications Agency (CCTA), 1997.

[17] J. Postel (Edt.): *Internet Protocol*; IETF, Request for Comments RFC 791, September 1981.

[18] B. Schneier: *Applied Cryptography*; Wiley, 1995.

[19] R. Steinmetz: *Multimedia-Technologie: Grundlagen, Komponenten und Systeme*; 2nd edition, Springer, 1999.

[20] UUNET: *Service Level Agreements*; http://www.uu.net/terms/sla/, February 2002.

[21] M.J. Wooldridge, N.R. Jennings: *Software engineering with agents: pitfalls and pratfalls*; IEEE Internet Computing, pages 20-27, May/June 1999.