

A Unified Framework for the Negotiation and Deployment of Network Services

Spyros Denazis, Lidia Yamamoto

Hitachi Europe SAS, Sophia Antipolis Laboratory
1503 Route des Dolines, F-06560, Valbonne, France
{Spyros.Denazis, Lidia.Yamamoto}@hitachi-eu.com

Abstract. The Internet network technology today does not allow a sufficient degree of autonomy to express user choices, constraints and preferences in order to dynamically obtain the most suitable services. One of the goals of Autonomic Communication is to produce self-managing network elements able to provide the desired services in an automated way. In this context, we propose an architecture to automate user-provider and provider-provider relationships, by converting the Internet into an electronic market space where the commodities to be traded are network services. After an agreement has been reached via agent-based automated negotiation mechanisms, network elements must be automatically configured in order to enforce the agreed conditions. This is achieved by generating commands to programmable network elements via open interfaces. The ultimate goal is enable fully automatic installation, configuration and monitoring of protocols or service components involving multiple ownership domains, while taking into account the constraints and preferences of users and providers.

1. Introduction

The Autonomic Communication initiative [1] is investigating the inter-relation among network elements to understand how their behaviours can be learned, influenced or changed such that they can self-organize to provide the desired services in an automated way.

The Internet network technology today does not allow a sufficient degree of autonomy to express choices, constraints and preferences in order to dynamically select the most suitable services. The user typically has to undergo a series of manual steps, for example when trying to connect to a Wi-Fi hotspot, in which the user has to enter authentication and billing data. When roaming to a different domain, e.g. switching to a different hotspot, the user is most of the times obliged to enter the data again and decide whether to accept the service offer or not. When many offers are simultaneously available, the user must inspect each of them and make a choice. Roaming is thus most of the times not transparent, and when it is transparent (e.g. roaming between mobile telephone operators) it is not always guaranteed that the user will benefit from the best offer that matches his or her interests.

We investigate the transition from the current Internet towards an Autonomic Communication Network (ACN). We focus on ways of bridging heterogeneity in requirements, interests, constraints and preferences.

We propose to model the new ACN as an electronic market space where network services are treated as goods that are traded among the different parties involved. We propose an open architecture able to make and enforce decisions even across ownership and administrative borders. This requires a unified framework for service description, announcement, discovery, negotiation and provisioning that takes into account the different preferences and constraints of the parties involved, and is able to achieve a common ground that is interesting for all parties involved. We use automated negotiation as the conflict resolution technique, and open interfaces for automated reconfiguration of services.

The same approach is applicable to several granularities of ownership borders: Users, groups of users, companies, network providers, etc. We will pay particular attention to user-provider (service provisioning) and provider-provider (inter-domain) interactions.

This paper is structured as follows: Section 2 provides a brief survey of the current state of the art in this topic, and identifies several open issues where more research is necessary. Section 3 presents a unified framework for creating an environment able to search for, negotiate, deploy, configure, monitor, reconfigure, and tear down end-to-end network services in an automated way across domain borders. Section 4 concludes the paper.

2. State of the art

Today it is still difficult to get network services autonomously when cross-ownership interactions are required. This is the case for both the customer-provider case (e.g. fixed or wireless access) and the provider-provider case (inter-domain agreements). Nowadays these interactions still rely mostly on slow human communication (e-mail, fax, paper contracts, and so on). Solutions to automate specific parts of the process are available, and some of them are discussed in this section. However the integrated picture seems somehow still missing.

The challenge is to automate cross-ownership interactions in ACNs, in a way to accurately reflect users' preferences, providers' interests and concerns, as well as the multiple underlying network characteristics. Such interactions cannot assume cooperation from the communicating peers, therefore security and safety are major concerns. Considering the amount of already existing network providers and users, the potential pairs of interacting peers is large. In this context, interactions cannot rely exclusively on authentication and authorization as a security mechanism, since this would imply constraining the interactions to trusted peers. In this model, interactions are driven by the level of trust one peer places in another, and closely mimic the corresponding human interactions. When dealing with several levels of trust, binary access/deny mechanisms are not sufficient. Negotiation mechanisms are necessary to reach intermediate, compromise solutions (e.g. access to a certain amount of a resource, or to a specific part of a document).

Negotiation can be used for configuration set-up, information transfer, new service deployment, or the usage of some physical resource such as bandwidth, CPU, and memory. The parameters involved include quality requirements, performance level, prices, payment conditions, etc. The decision criteria relate to user preferences, for example, towards the fastest network access, or the closest, the most reliable, the cheapest, etc. The user could also prefer to combine service from multiple available providers in order to increase robustness.

2.1. Automated Negotiation

Automated negotiation [11,12] mimics human negotiation processes to reach agreements on one or more issues. It is an active research topic in the field of multi-agent systems, and has been applied to several areas including telecommunication and computer networks. A number of agent-based systems to enable provider selection and inter-domain interactions have been proposed [13,14,15,16,17]. The use of Agent Communication Languages enables rich and flexible interactions, which can be made interoperable through standardized specifications provided by the FIPA consortium [18]. Several FIPA standard protocols and languages are available that can be used for this purpose: Contract negotiation [19,20], brokering [21], proposals [22], auctions [23,24], QoS [25], network management [26]. Work is in progress towards a FIPA standard for agreement specification [27,28]. When ready it could be used, for instance, to specify a Service Level Agreement (SLA).

A multi-agent system for automated negotiation applied to VPN provisioning is described in [13]. Its agents comply with FIPA standards and implement multiple negotiation strategies. However it does not seek to optimise the VPN topology, resulting in a star configuration.

In the framework proposed in [16] the problem of inter-domain QoS routing is formulated as a Distributed Constraint Satisfaction Problem (DCSP). The QoS requirements considered are bandwidth and delay. A distributed algorithm is derived, which is suitable for unicast guaranteed QoS services. However, the algorithm is only valid when all domains on the end-to-end path support the specified agents and resource reservations, and cooperate to offer the requested service. In [17] the framework is extended with negotiation mechanisms and an ontology for VPN services. Price negotiation takes place after a path has been selected using the DCSP algorithm, therefore provider selection is not a direct outcome of such negotiation.

A formal model of the service selection problem is presented in [14], in the context of agent-mediated wireless access. A user agent called Personal Router acts on behalf of its owner to select wireless providers that better satisfy the user's preferences. These preferences are modelled as a utility function of receiving given service profiles. The selection problem is represented as a Markov Decision Process and the initial solution is to find those actions that maximize utility. However it is shown that the algorithm is computationally expensive. In [15] heuristic solutions are proposed that can make the problem more tractable.

2.2. Towards automated inter-domain interactions

Automated inter-domain interactions are mostly limited to inter-domain route advertisement via Border Gateway Protocol (BGP) messages. Most ISPs today rely on the BGP community mechanism to have tighter control upon route propagation [4] by specifying preferred paths and deviating traffic outside their domains. There are many problems with this approach, as frequently pointed out in literature [5,6]: sub-optimal end-to-end paths, instabilities (route flapping), slow convergence in response to link failures. It would be better for the set of domains to cooperate in order to obtain the best routes according to given metrics that satisfy users' requirements. We are talking about inter-domain QoS routing. Although there has been some research in this topic [5] as well as IETF guidelines [7] it remains largely an open issue at the moment.

Proposals to include QoS information in BGP are presented in [8,9]. A signalling approach for network state management is proposed in [10] that can be used for intra-domain as well as inter-domain QoS and other monitoring and configuration tasks. None of these approaches is generic enough to express the complexity of fully competitive inter-domain interactions related to end-to-end services, in which the trade-off between competition and cooperation must be taken into account and quantified. A richer approach is needed to cover the whole service cycle in an end-to-end basis, including service request, negotiation, selection, set-up, monitoring, renegotiation, and tear down. Our ideas to achieve this goal will be discussed in Section 3.

There are other gaps in existing work towards automating interactions among multiple, potentially competing ownership domains. First of all, partial deployment must be supported. For instance, let us consider a path is made up of domains A , B , C and D in sequence, with A as source domain and D as destination domain. If a service uses providers A , B , C and D , but only providers A and C provide automated negotiation capability, then the network characteristics of providers C and D should be measured as a black box, such that some information is available in order to provide the customer with an estimation of expected service level. Although no absolute guarantees can be provided in this case, such estimation can represent valuable information to influence the customer's decision in favour of a given provider.

Cascade negotiation towards an end-to-end service is only partially supported in existing approaches. In the case of [16] the first domain agent (in the source domain) communicates with all the other domain agents on the path to a given destination. In the example of path A - B - C - D above, the agent in domain A would send negotiation messages to B , C , and D . It would be more transparent if A would negotiate with B , B with C , and C with D following the path sequence. Cascade negotiations are directly related to inter-domain routing: if a negotiation fails or if a provider fails to comply with agreements, an alternative provider could be selected, resulting in a different end-to-end path. In the same way as unilateral BGP policies may have negative impact on global routing, cascade negotiations could lead to routing instability if conducted in an ad hoc manner. Further research is needed to fully understand the impact of cascade negotiations on inter-domain routing, and to provide methods that can guarantee that a stable route is found in reasonable time.

2.3. Inter-domain network programmability made feasible

Programmable networks [36] have been proposed in response to the need of more flexible and customisable network nodes for improved services and faster service deployment. Using a programmable network infrastructure, applications can benefit from available processing time and memory storage in intermediate nodes, which could be used to install and execute customized service components. The use of such node resources within the network raises security and safety concerns, and can only be made realistic with tight security and resource usage control. In the wide area end-to-end case, the benefit of programmable networks might be realized only through the installation of customized components in several nodes potentially belonging to different administrative domains. This raises even deeper concerns as network providers will be more than reluctant to open their nodes to foreign code.

If we can design an automated negotiation mechanism which is rich enough to express the characteristics of dynamically deployable components to be installed in the network, including their provenance, resource consumption and reliability, this could encourage providers to allow trusted components to be installed and executed in the network nodes supplied for that effect, therefore stimulating the usage and deployment of new network services involving multiple domains.

2.4. Ubiquitous, ad hoc, sensor, and other small-device networks

Parallel to what is happening to the Internet infrastructure, several infrastructureless, self-organizing networks are emerging, such as ad hoc, sensor networks, ubiquitous networking, home networks. These networks should be formed spontaneously anywhere at any time, without requiring network operators or network managers. ACNs could benefit from ideas stemming from these networks to help automating tasks such as network management and service provisioning. On the other hand, these light-weight networks will also need to connect to existing more complex network infrastructures, where most of the content and services can be found, such as the Internet, Intranets, and Virtual Private Networks (VPNs). Multiple alternative providers may be available at a given location (e.g. wireless and wired), and multiple terminals may be able to act as gateways from the ad hoc network to the outside infrastructure.

The users of a device network are then faced with the problem of which provider or set of providers to select for access to an infrastructure, and through which gateway nodes. The choice of an optimum or nearly optimum connection solution may be non-trivial, involving many parameters such as expected throughput and delay, price, eventual service guarantees, level of trust in known providers, etc. This should be handled in an automated way, such that the users simply specify their preferences and the network nodes cooperate to find the optimum solution.

Resources dedicated to device networks are often limited, such as low bandwidth wireless links, and small terminals such as PDAs and cell phones with slow CPUs small memory space, and short battery life. Such limitation might also open up new markets for infrastructure-based computational services targeted at complementing these resources by outsourcing or through a Grid-style distributed computing

paradigm. The framework proposed in this paper can play a critical role in achieving this goal.

3. A Unified Framework for the Negotiation and Deployment of Network Services

In this section, we attempt to create a vision about the future Internet and some of the features it should manifest. According to it, we aspire to treat the Internet as a market place wherein the commodities are network services. The majority of electronic markets proposed so far, has been about tangible goods ranging from equipment to clothes, books, shares etc. In our case, we propose to treat the most basic service of Internet, namely, communication, as a commodity together with other related services and applications such as video conference, voice, etc.

Realising our vision and applying a methodology, we have borrowed from models describing interactions and operations taking place in a market economy leading us to a unified framework for service description, announcement, discovery, negotiation and provisioning. Such a framework also takes into account the different preferences and constraints of the parties involved.

3.1. Modelling customer-provider interactions

The interactions that occur between end user and provider, and between peer providers can be modelled as producer-consumer interactions in a market economy. Numerous e-commerce systems have been proposed or are in use nowadays, which also model these economic agents.

Initial systems e-commerce systems [29,30] employed software agents as mediators for handling and automating interactions taking place in physical commerce. They followed Consumer Buying Behaviour (CBB) [31] to model actions and decisions that happen when buying and selling goods augmented to incorporate concepts from Software agents research. CBB models originating from traditional market research can be abstracted into one CBB model consisting of six stages that coarsely reflect consumer behaviour [31]:

- **Stage 1: Need Identification:** This stage is where customer realises his or her need for a specific product or service.
- **Stage 2: Product Brokering:** This stage answers the question of ‘What to buy?’. The customer follows a course for gathering information in order to decide not only upon the product but also on its exact characteristics (product profile).
- **Stage 3: Merchant Brokering:** This stage answers the question of ‘Who to buy from?’. The customer, having decided on the product profile, takes into consideration additional information about the merchant which are filtered through the customer’s own criteria in order to reach a conclusion. For instance, lowest price, value for money, reputation, etc.
- **Stage 4: Negotiation:** This stage answers the question of ‘How to buy?’. It is revolved around the rules governing the transactions between two parties. For

instance, negotiating price or QoS level. This stage can be considered as a part of the previous two stages or a standalone stage depending on the type of market.

- **Stage 5: Purchase and Delivery:** This stage usually heralds the completion of the negotiation stage. It may also have an influence on the product and merchant brokering stages.
- **Stage 6: Service and Evaluation:** Finally, this is the stage where an evaluation period of the product and the promises that accompany it, commences. If such promises are not fulfilled the customer might decide to renegotiate or even choose alternative providers, going back to stages 4 or 2.

The CBB model and its stages above provide a very rough but reasonable guide for categorising the actions performed by any e-commerce system. Software agents materialise these actions in the context of the CBB model. Notably though, some of the stages often collapse into one, or overlap, and migration from one another can be non-sequential or iterative depending on the kind of e-commerce and eventually on the type of product(s) involved. Naturally, the variations of the CBB model have an immediate effect on the selection of agent technologies, languages, protocols, interfaces, and the actions agents perform which altogether constitute the e-commerce system.

Note however that the use of agent systems is restricted to the algorithms, concepts and protocols that are useful in a networking context, not the actual platform implementations. Agent platforms usually offer a complete infrastructure of services which is more suitable to support application-layer implementations. Network layer issues require lightweight methods that do not rely on an existing communication support - the framework itself is intended to provide such communication services, therefore cannot assume that they are ready for use. Ultimately, negotiation algorithms and protocols should be embedded as services into the autonomic communication system itself, at the same level of any other communication service also present there.

3.2. A High Level Description of the Unified Framework

The core idea behind our framework is the transformation of the Internet into an environment that acts as a distributed market place where potential merchants (Network or Internet Service Providers) and potential customers (home users, small enterprises, corporations, or even other ISPs) interact in order to compete and cooperate over selling, or buying a specific commodity, namely, *network services*. Starting with a competitive environment, the system should foster cooperation among providers in order to achieve improved end-to-end services.

Service Level Agreements (SLAs) play an important role in representing the profile of the product, i.e network services, in an unequivocal, and discrete manner. SLAs can, then, be used by the customers to quote a price, to negotiate with a provider(s), and evaluate the quality of service depicted in an SLA. In contrast, providers can compete with each other by lowering their prices over a requested SLA, or offering more advanced services on top of the requested ones, form pacts with other providers in order to increase their competitiveness, and reserve network

resources for guaranteeing the SLA. Moreover, the collection of SLAs offers them a picture of their current and future resources needs, thereby facilitating management, re-engineering, and provisioning of infrastructure.

An SLA is an agreement between two roles, that of a customer and a provider. The customer may be an end-user or another provider. A retail SLA (r-SLA) is an agreement between an end-user and a service provider. A wholesale SLA (w-SLA) is an agreement between providers, and is usually based on aggregates meant to carry traffic from several end users. W-SLAs may be established in advance on a static basis, as part of a network provisioning phase, and independent of r-SLAs. However, a given r-SLA may also trigger the establishment of several w-SLAs across the domains involved, in a dynamic way.

Widespread use of SLAs aspires to establish a universal interface language among the involved parties representing services together with their characteristics that can be uniquely recognised and interpreted along the end-to-end path. Such language will facilitate the automation of network operations like customer-provider negotiations, (re)configuration of network resources etc. We expect that SLAs will form ontologies of objects ranging from generic ones customised for the needs of technology agnostic customers to detailed ones addressing the needs of experts like network managers.

Our framework should give the possibility to a customer to choose from a basic set of service parameters, like availability, throughput, latency, privacy, etc., those that are most desirable resulting in a *provisional SLA*. The user specifies high-level preferences and the user agent maps preferences to service parameters, then requests service from one or more providers via an automated negotiation mechanism. This activity (Fig. 1) can be considered as representative of *stage 2 (product brokering)* in the CBB model of the previous section.

The next step is to discover available providers and if necessary inform the customer about which providers may be contacted to participate in the negotiations. Providers could announce themselves with help of directories, beacons, or other mechanisms.

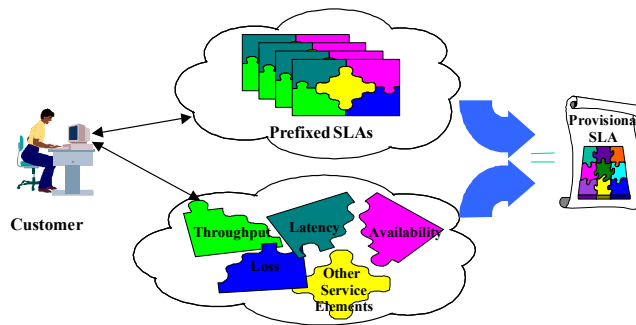


Fig. 1. Process of building Provisional SLAs

Using the customer service profile, the user agent must now decide upon the provider(s) to which to award the SLA. This is *stage 3 (merchant brokering)*. The decision may be completely automated or assisted by its human owner. Hence, the framework provides the means to support the submission of a provisional SLA to a

number of providers along with an additional set of parameters that constitute the customer's criteria, e.g. price. This could be implemented using a standard language to express agreements (work is in progress in FIPA [27,28]). Receiving the provisional SLA and the customer's criteria, providers may respond with an offer on the submitted criteria or may even add more service elements or services in order to form an appealing package. This can be done through argumentation-based negotiation [11]. This offer returns to the customer in the form of a *Combined SLA* (Fig. 2). At this point the customer's agent (assisted or not by this customer) either selects a provider and accepts the provider's Combined SLA, or makes counter-offers. This is *stage 4* of the CBB model, the *negotiation stage*. Note that in this iterative negotiation model involving alternative providers, stages 3 and 4 are combined.

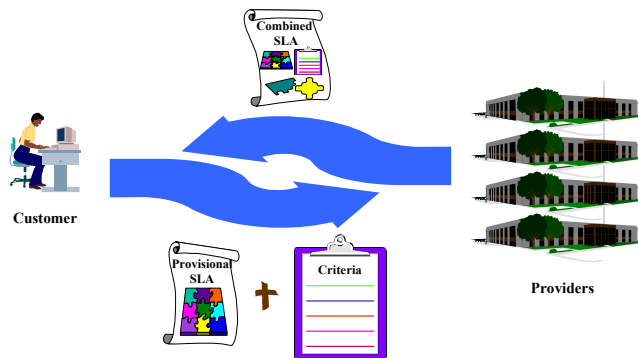


Fig. 2. Process of creating Combined SLAs

Taking the Combined SLA(s) for negotiation over some terms could be a complex and iterative process. For example, the customer's agent may request lowering the price in a gambling attempt, or by reducing some of the initial demands. Another possibility emerges when a final SLA has been agreed and the customer wishes to increase demand on some of the constituent parameters in the agreed SLA, i.e. more bandwidth. Providers may also make such offers when they see fit. It is expected that this stage will be involved in many temporal instances as the interactions between customers and providers occur dynamically. Each time that a new agreement takes place, the new terms are added to the existing SLA.

Given that an agreement has been reached between the two parties the delivery and charging mechanisms are triggered, entering thus *stage 5 (purchase and delivery)*. This corresponds to the configuration and operation of the service, together with an electronic charging and payment scheme. Furthermore, delivery requires from the provider to exert control on its infrastructure by reserving resources, applying scheduling policies, etc., in order to be able to deploy the services stated in the SLA. At the customer side this implies autoconfiguration of its network elements to support the desired service.

Finally, the last stage (*Service and Evaluation*) involves actions by both the provider and the customer. The provider should constantly monitor and manage its network so as SLAs from different customers are upheld, take correcting actions

when needed, and policing customers that abide to the SLA. This requires inter-domain negotiations to reach agreements on which parameters a provider allows to be monitored by its customer or by peer providers. On the other hand, customers should also be given the ability to monitor the network so as to make their own choices. Out of this process, customers may either decide to modify some of the terms of the SLA in due course, returning thus back to the negotiation stage, or decide to form new SLAs for other set of services returning back to stage 2. Providers that fail to fulfil the requirements in an SLA risk sanctions or bad reputation. This requires infrastructure monitoring for connectivity and performance. In a general end-to-end case cascade monitoring might be needed as discussed in Section 2.2.

The idea of automating the negotiation of SLA parameters is not new, and has been presented, for instance, in the CADENUS project. However, to the best of the authors' knowledge, CADENUS focused on Quality of Service management and control. It did not treat the problem from a network autonomies perspective, nor did it make use of automated negotiation algorithms and protocols such as those discussed in Section 2.1. In CADENUS, it is the user who participates in the negotiation process by selecting the desired service based on a sorted list of offers, while automated negotiation algorithms (Section 2.1) handle the whole negotiation process on behalf of the user, guided by previously defined user preferences (expressed, for example, in the form of utility functions).

It is important to clarify that our framework does not dictate the way to express preferences and constraints at the customer or provider side, nor does it specify any feedback mechanisms to eventually update such preferences and constraints based on negotiation outcomes. For example, at the provider side, such preferences and constraints could be expressed using policy languages. In this case, we assume that quantitative parameters can be extracted from such policies, and that these parameters can be used to determine upper and lower thresholds in the negotiation process.

3.3. Automated Deployment and Configuration

Realising stages 5 & 6 requires technologies and network architectures capable of supporting automated deployment of the agreed network services. This also entails the (re)-configuration of the network elements across the different domains that the end-to-end path traverses. Furthermore, before or when the service has been deployed detailed monitoring data must be collected and processed (discussed in the next section)

Accordingly, there is a need for higher degrees of flexibility supported by networks and the network elements thereof. Such flexibility is measured as the network's capability to dynamically extend and change its overall functionality and behaviour through the on-demand introduction and configuration of functional components.

Engineering the (heterogeneous) network with this kind of flexibility requires the design and adoption of a number of key concepts proposed or extended by active and programmable networks initiatives [36]: service component models [33,34], open interfaces [38], and new network element architecture models [35,40].

Component-based models consider services as comprised of self contained building blocks that act as primitives that make up more complex services when

combined in specific ways. Treating complex services as components increases the flexibility of their deployment and allows for better decision-making.

Open interfaces avoid dependencies on a small group of vendors created by proprietary interfaces. They allow algorithms and services to be developed independently of advances in the forwarding plane, as they allow seamless configuration of the control and management planes.

Network element and configuration models provide a common and unambiguous view of the network and its state thereby facilitating the communication among different parties. They also provide a richer set of information that can drive and enhance the decision-making and monitoring algorithms.

Collectively, these concepts constitute a universal network language in much the same way as SLAs represent what customers want and understand as network services. Through such language, providers are capable of representing and communicating with each other information about service entities, network resources and their location, implementation technologies, vendors' equipment etc. Based on this information they can then feed their decision making algorithms about where, when and how to deploy a service, manage it throughout its lifetime or reconfigure it according to customer changing requirements.

In this context, upon completion of the negotiation phase with the customer, a provider translates the SLA to its corresponding service component model in the form, say, of an XML schema, which is then processed by the service deployment algorithm [37]. The decision of where to deploy the components of the service depends on the network model that captures information about the available resources and implementation environments. This information is compared against service requirements, for instance, how much bandwidth is needed, the implementation environment that the service components require etc.

When a decision is made the service deployment mechanism is contacted in order to enforce the decision. The enforcement is facilitated by open interfaces that abstract a common set of deployment mechanisms implemented according to the implementation platform of the network element. The decision and deployment process enforced in the provider's domain may trigger a series of similar decisions and deployment operations made by other providers that eventually will form the end-to-end path across the different providers' domains.

We envisage that the deployment of a service will be comprised of two parts: the deployment of the QoS model that satisfies the resource requirements and the deployment of functional components that will process packets beyond store-and-forward processing and belong to the service. The QoS deployment heavily depends on open programmable interfaces too, like IEEE P1520 [38] or ForCES [40] that facilitate a suitable mapping between resource demands and configuration operations on the network elements independent of the vendor or platform of the network element.

The deployment of a service is followed by the management of the service in such a way that the obligations of both parties described in the SLA are fulfilled. To this end an intelligent monitoring system must collect and disseminate statistics and data to the interested parties. A special operation of the monitoring system is to feed the network management algorithms with alarms so that proactive actions must be initiated and carried out in an effort to stick to the agreed SLA as close as possible.

Again, the adoption of common models and of open interfaces enables automating and expediting these tasks through an ambient interoperability layer.

3.4. Automated Service Monitoring

After service deployment and configuration, the service should be monitored to comply with SLA, to identify non-cooperative or misconfigured domains, etc. The ability to perform inter-domain QoS measurements is crucial to provide reliable and high quality services. Monitoring is required for troubleshooting, and automated set-up of monitoring tasks is the first step towards automated diagnosis and repair. However today, monitoring an arbitrary end-to-end path today is difficult and restricted, and the obtained information is very limited and inaccurate. A great obstacle against global-scale performance monitoring today is that network providers are not willing to share information about their networks, due to fear of eavesdropping by competitors, fear of attacks, and various business reasons.

It would be beneficial to have an automated way to dynamically express which parameters may or may not be monitored across domains, depending on trust levels among providers. An automated negotiation mechanism would enable the automatic set-up of measurement tasks across domain, while at the same time respecting providers' policies and restrictions. This would also act as incentive for cooperation, as providers that cooperate to offer monitoring results would be in a better position to offer higher quality services appreciated by customers. We have taken a first step in this direction [2] by proposing an automated negotiation framework for the dynamic set-up of network monitoring tasks across domain borders.

In [2] we proposed to apply automated negotiation techniques as a way to dynamically agree on which QoS parameters may be monitored across domains, depending on the resources available within each domain, the current network conditions, the trust levels among providers, and their respective policies and constraints, including security and privacy constraints. We have identified the potential protocols and strategies that could be applied, and mapped monitoring parameters to them. As a format for the exchange of requested measurement parameters, we have selected the Specification of Monitoring Service (SMS) [3] proposed for inter-domain monitoring. The SMS format is a document format which contains the necessary parameters to request inter-domain QoS monitoring tasks.

The resulting monitoring data must be used as feedback information for the autonomic communication control system, such that deviations from the expected service can be promptly detected and a system reconfiguration can be triggered when necessary. This remains largely an open issue in current networks, where the focus is on database storage and visual analysis of measurement results.

An example of how monitoring information could be used as feedback for decisions processes in Autonomic Communication would be to automate diagnose and repair of network problems (troubleshooting). The challenge is to perform this across multiple ownership boundaries, in order to achieve consistent end-to-end service. Thaler et al. [41] propose a generic architecture for distributed troubleshooting which also works in the inter-domain scenario. The architecture includes a protocol called Globally Distributed Troubleshooting (GDT), for

automated problem and status reporting across different domains. Nevertheless, automating network troubleshooting remains a non-trivial problem.

4. Conclusions

This paper aspires to increase awareness among researchers for greater degrees of automation in the network, and to identify specific aspects that must be engineered into the network in order to achieve this. Automated negotiation algorithms and protocols can be applied between customers-providers and providers-providers, as the current static model is very restrictive and outdated to cope with the requirements of a truly autonomic network, which must detect and resolve conflicts of interest in an automated way.

With this in mind, an initial unified framework has been proposed aspiring to transform today's Internet into a shopping place for network services, the basis for communication between endpoints. The framework was kept intentionally as generic as possible in an effort to serve as an ambitious and long-term research programme where different technologies, solutions and algorithms may be tried and evaluated.

To this end, the proposed approach may foster cooperation among providers, since those providers that cooperate and negotiate mutually beneficial agreements will be in a better position to provide better services and to promptly react to customers' requests. Moreover, since it will become easier for users to select providers, they will be more likely to select those providers that offer a better cost-benefit relation, and this will only be possible if they have agreements for feedback and measurements on the performance levels and open their infrastructures to customized services.

5. References

1. Mikhail Smirnov, "Autonomic Communication: Research Agenda for a New Communication Paradigm", White Paper, Fraunhofer FOKUS, Berlin, Germany, November 2003, <http://www.autonomic-communication.org/publications/>
2. Lidia Yamamoto, "Automated Negotiation for On-Demand Inter-Domain Performance Monitoring", Proceedings of 2nd International Workshop on Inter-Domain Performance and Simulation (IPS 2004), Budapest, Hungary, March 2004, pp.159-169.
3. Elisa Boschi, Salvatore D'Antonio, Giorgio Ventre, "Inter-domain Communication and Data Exchange", Proceedings of 2nd International Workshop on Inter-Domain Performance and Simulation (IPS 2004), Budapest, Hungary, March 2004, pp.65-72.
4. Bruno Quoitin and Olivier Bonaventure, "A survey of the utilization of the BGP community attribute", Internet Draft draft-quoitin-bgp-comm-survey-00.txt (work in progress), February 2002.
5. Samphel Norden and Jonathan Turner, "Interdomain QoS Routing Algorithms", Washington University, Department of Computer Science Technical Report, WUCS-02-03, 2002.
6. Timothy G. Griffin, Gordon Wilfong, "Analysis of the MED Oscillation Problem in BGP", 10th IEEE International Conference on Network Protocols (ICNP'02), Paris, France, November 2002.

7. E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, "A Framework for QoS-based Routing in the Internet", Section 5: "Interdomain Routing", Internet RFC 2386 (Informational), August 1998.
8. Olivier Bonaventure, "Using BGP to distribute flexible QoS information" Internet Draft draft-bonaventure-bgp-qos-00.txt (work in progress), February, 2001.
9. Li Xiao, King-Shan Lui, Jun Wang, Klara Nahrsted, "QoS Extension to BGP", 10th IEEE International Conference on Network Protocols (ICNP'02), Paris, France, November 2002.
10. Cornel Pampu, Henning Schulzrinne, Xiaoming Fu, Cornelia Kappler: "Design of Technology Independent QoS Signalling Protocol for intra- and interdomain environment", First international workshop on Inter-domain performance and simulation (IPS 2003), Salzburg, Austria, February 2003.
11. N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra and M. Wooldridge, "Automated negotiation: prospects, methods and challenges", In International Journal of Group Decision and Negotiation, 10(2), pages 199-215, 2001.
12. M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam, "Protocols for Negotiating Complex Contracts", IEEE Intelligent Systems Journal, Special Issue on Agents and Markets, Volume 18, Number 6, pp. 32-38, 2003.
13. P. Faratin, N. Jennings, P. Buckle and C. Sierra, "Automated Negotiation for Provisioning Virtual Private Networks using FIPA-Compliant Agents", The Fifth International Conference and Exhibition on the Practical Application Of Intelligent Agents And Multi-Agent Technology (PAAM-2000), pp. 185-202, Manchester, UK, 2000.
14. P. Faratin, J. Wroclawski, G. Lee and S. Parsons, "The Personal Router: An Agent for Wireless Access", In Proceedings of the AAAI Fall Symposium on Personal Agents, pp. 13-21, N. Falmouth, Massachusetts, US, 2002.
15. P. Faratin, J. Wroclawski, G. Lee and S. Parsons, "Social User Agents for Dynamic Access to Wireless Networks", Proceedings of the AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments, Stanford, PA, US, 2003.
16. M. Calisti, B. Faltings, "Distributed constrained agents for allocating service demands in multi-provider networks", Journal of the Italian Operational Research Society, Special Issue on Constraint-Based Problem Solving, volume XXIX, number 91, 2000.
17. M. Calisti, B. Faltings, "Agent-Based Negotiations for Multi-Provider Interactions", Proceedings of 2nd International Symposium on Agent Systems and Applications (ASA 2000), Zurich, Switzerland, September 2000.
18. The Foundation for Intelligent Physical Agents (FIPA), <http://www.fipa.org/>
19. FIPA Contract Net Interaction Protocol Specification, SC00029H, December 2002, <http://www.fipa.org/>.
20. FIPA Iterated Contract Net Interaction Protocol Specification, SC00030H, December 2002, <http://www.fipa.org/>.
21. FIPA Brokering Interaction Protocol Specification, SC00033H, December 2002, <http://www.fipa.org/>.
22. FIPA Propose Interaction Protocol Specification, SC00036H, December 2002, <http://www.fipa.org/>.
23. FIPA English Auction Interaction Protocol Specification, XC00031F, August 2001, <http://www.fipa.org/>.
24. FIPA Dutch Auction Interaction Protocol Specification, XC00032F, August 2001, <http://www.fipa.org/>.
25. FIPA Quality of Service Ontology Specification, SC00094A, December 2002, <http://www.fipa.org/>.
26. FIPA Network Management and Provisioning Specification, XC00082B August 2001, <http://www.fipa.org/>.

27. Nicholas R. Jennings, Abe Mamdani, Timothy J. Norman, Jeremy Pitt, "Agreements Work Plan", FIPA work plan (work in progress) <http://www.fipa.org/docs/wps/f-wp-00008/f-wp-00008.html>, August 2000.
28. FIPA Board of Directors, "TC Agreement Management Resolutions", <http://www.fipa.org/docs/output/f-out-00084/f-out-00084.html>, April 2001.
29. A. Chavez and P. Maes, "Kasbah: An Agent Marketplace for Buying and Selling Goods", In Proceedings of the First International Conference on the Practical Applications of Intelligent Agents and Multiagent Technology (PAAM '96), London, UK, 1996
30. P.R. Wurman, M.P. Wellman, and W.E. Walsh, "The Michigan Internet AuctionBot: A Configurable Auction server for Human and Software Agents", In Proceedings of the Second International Conference on Autonomous Agents (Agents-98), Minneapolis, MN, USA, May '98.
31. R.H. Guttman, A.G. Moukas, and P. Maes, "Agent-mediated Electronic Commerce: A Survey", Knowledge Engineering Review, June 1998.
32. G. Cortese, R. Fiutem, P. Cremonese, S. D'antonio, M. Esposito, S.P. Romano, A. Diaconescu, "Cadenus: creation and deployment of end-user services in premium IP networks", IEEE Communications Magazine, Volume 41, Issue 1, pp.54-60, Jan 2003.
33. Object Management Group: CORBA Components, Revision 3.0, OMG document formal/02-06-65
34. Vicente, J., S. Denazis, et al., "L-interface Building Block APIs", IEEE P1520.3, P1520.3TSIP016, 2001.
35. S. Denazis, S. Karnouskos, T. Suzuki, S. Yoshizawa, "Component-based Execution Environments of Network Elements and a Protocol for their Configuration", IEEE - Transactions on Systems, Man and Cybernetics, Special Issue on Technologies that promote computational intelligence, openness and programmability in networks and Internet services, Autumn 2003 (in press)
36. A. Galis, S. Denazis, C. Brou, C. Klein (ed), "Programmable Networks for IP Service Deployment", Artech House Books, ISBN 1-58053-745-6, 2004.
37. "Overview FAIN Programmable Network and Management Architecture", FAIN Project Deliverable 14, <http://www.ist-fain.org/>
38. Biswas, J., et al., "The IEEE P1520 Standards Initiative for Programmable Network Interfaces", IEEE Communications, Special Issue on Programmable Networks, Vol. 36, No 10, October, 1998.
39. IETF ForCES, draft-ietf-forces-framework-04.txt, December 2002, <http://www.ietf.org/internet-drafts/draft-ietf-forces-framework-04.txt>
40. Yang, L., J. Halpern, R. Gopal, R. Dantu, "ForCES Forwarding Element Functional Model", March 2003.
41. David G. Thaler and China V. Ravishankar, "An Architecture for Inter-Domain Troubleshooting", Journal of Network and Systems Management, 12 (2): 155-189, June 2004.