Journal of Network and Systems Management, Vol. 12, No. 3, September 2004 (© 2004)

Dynamic Service Management in Heterogeneous Networks

Maurizio D'Arienzo,[1] Antonio Pescapè,[1,2] and Giorgio Ventre[1]

Novel network architectures allow users to get specific performance guarantees via the definition of a document where QoS (Quality of Service) requirements are linked to the user traffic description. Such a document is defined SLA (Service Level Agreement), and it is a formal high level definition (user view) of characteristics for a communication service, whereas low level specification (network view) is obtained translating the SLA in a different document named SLS (Service Level Specification). Although in the case of static services the network configuration process is a well defined activity, when dynamic services come into play a more complicated scenario where SLA translation into the appropriate SLS is not a one-step and static process, but it needs an active and consistent evaluation with respect to the current situation. We claim that to make this process happen automatically, we need to have intelligent devices able to translate request specified inside SLAs in the most appropriate network configuration (by means of dynamic SLS) depending on client's current "service conditions." In this paper we introduce a framework for distributed network management through an entity, namely AcMe (Active Mediator), which performs dynamic creation of network services in a transparent to the user fashion. A new protocol, HNMP (Heterogeneous Network Management Protocol), orchestrates all AcMe functionality. Finally, an experimental analysis is presented.

KEY WORDS: Service level agreement; service level specification; programmable networks; distributed management platform.

1. INTRODUCTION

As computer networking has become more ubiquitous, researchers are increasingly focusing their efforts on optimizing computer network performance, in particular with respect to well known parameters like bandwidth, delay, jitter, and packet

[1]Dipartimento di Informatica e Sistemistica, University of Napoli "Federico II," Napoli, Italy.
[2]To whom correspondence should be addressed at Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli "Federico II," Via Claudio, 21, I-80125, Napoli, Italy. E-mail: pescape@unina.it

1064-7570/04/0900-0349/0

loss. Even though it still remains a vexed question, QoS (*Quality of Service*) is a key factor for deployment of future new value added services. In fact, the introduction of QoS mechanisms in the Internet is expected to enable widespread use of real time services, such as VoIP (*Voice over IP*) and video streaming applications. The enhancement of network infrastructures necessarily stems from the adoption of a new paradigm in network management, in which systems are capable of transparently identifying the "*per user*/*per service traffic profile*" and automatically matching it with the most appropriate configuration of network devices.

Some of the most recent proposals of frameworks for deployment of new services rely on architectures where users are able to make explicit service requests by means of *Service Level Agreement* (SLA) subscription. While in the past SLAs were just used for regulating network interconnections agreements, nowadays they have been adopted as a tool for retail negotiations. SLAs are simple contracts established between one or more users and one or more *Service Providers*. These contracts might be renegotiated to allow recognized users to subscribe to new services offered from a generic Provider, or to modify the characteristics of an already established one. In case of static services, i.e., services where requirements are independent from variable conditions, the Provider is asked to accommodate the service at its subscription and at its invocation from the user. When this happens, it is needed to translate the SLA into a more technical document, the SLS (*Service Level Specification*), which is used for the actual network configuration.

However, with the introduction of *dynamic services*, an a-priori knowledge of performance requirements related to the subscribed SLA might become a problem. Dynamic services can be linked to the novelty of the service itself, or to the variability of the performance requirements for different instances of the same service. For example, SLAs for accessing a Video Distribution service should be linked to the content available and, in particular, to the requirements of the specific video that the user will select for purchase. This is clearly a case where the service accommodation is depending neither on the user needs or requirements, nor on the characteristics of the service involved, but rather on the content. In this situation, SLS definition is not a simple translation of high level service parameters contained in SLA, but it requires a consistent evaluation performed every time the service has to be accommodated [1].

A process of evaluation and computation of SLS is therefore needed in all cases where parameters to be specified within the SLA are difficult to identify. This is the case of services linked not only to the service itself, but also to uncertainties related to the way a user connects to the service, e.g., mobile users. There are, therefore, strong motivations that lead to the definition of frameworks where network configuration is a dynamic process. New heterogeneous networks represent a real scenario where SLA negotiation and the subsequent SLS computation is a

dynamic task. It is important to understand how users' requirements change in the scenario depicted above.

The rest of the paper is organized as follows. Section 2 introduces the concept of "*Service Condition*." In Section 3 a general overview of emerging requirements in a heterogeneous scenario is depicted. Section 4 describes the network configuration issues that our proposal deals with, whereas Section 5 presents the motivation for our work and the related work. In Section 6 we address the problem of dynamic resource management. Section 7 introduces a protocol for service management in heterogeneous networks, the Heterogeneous Network Management Protocol (HNMP). Some results from an experimental testbed are reported in Section 8, and finally Section 9 concludes with some considerations and issues for future research.

2. A NEW CONCEPT IN A NEW SCENARIO: THE SERVICE CONDITION

In a pervasive and ubiquitous computing scenario, several questions arise when we want to describe the way a service should be implemented to correctly fit requirements contained in a subscribed SLA, especially in case of dynamic services. We can summarize these questions in the concept of "*heterogeneity*" with respect to terminal, network and service.

1. *Terminal heterogeneity*: First, we need to know the device characteristics that will be used to receive the content. Devices can range from high-performance workstations, to PDA (*Personal Digital Assistant*), down to mobile phones with limited video reproduction capabilities. It is reasonable to expect that future services impose to the same user the need of using a wide collection of terminals and of freely moving from one terminal to another depending on the situation. Of course, this information should be managed to have the content delivered to the user with the format most suitable to the device currently adopted.
2. *Network heterogeneity*: Second, we need to know the characteristics of the network that will be used to deliver the content, since also this one is an critical factor for the correct definition of an SLA (and of the subsequent SLS). In the current Internet, even if we consider as dynamically variable only the part of a network infrastructure that is closest to the user (i.e. the so called access network or edge network), we have a quite large number of options to deal with: wireline (*corporate LANs*, *cable*, *xDSL*, *modem*), wireless (*WLANs*, *Bluetooth*), 2/2.5/3 G mobile networks (*GSM*, *GPRS*, *UMTS*).
3. *Service heterogeneity*: Third, as already mentioned before, we need to know the characteristics of the service itself, in terms of media involved

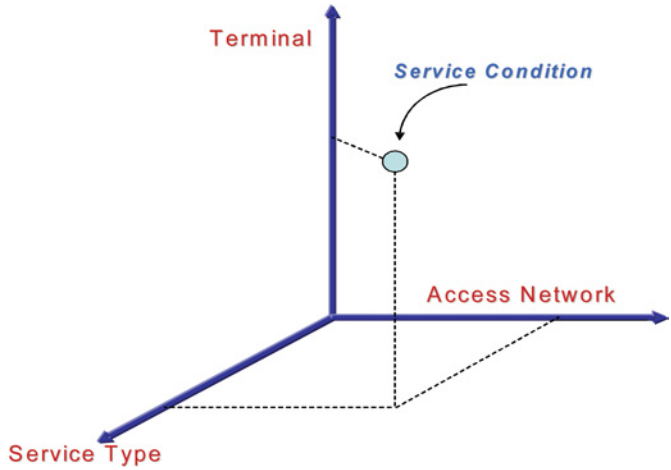


Fig. 1. Example of service condition concept.

(audio, video, graphics), of their format (coding and compression techniques), and in terms of the typology of the service (synchronous, asynchronous, transactional, . . .).

We have therefore a three-dimensional space where QoS requirements depend on the variability of three different technical aspects. A point in this space is called a *Service Condition* (Fig. 1).

In the previous schema we could of course place in an additional variable: time. In this way we would move our attention from a three-dimensional space to a more complex, four-dimensional space.

It is clear that to allow future users to have ubiquitous access to novel media services we need to allow them to roam transparently across different networks, terminals, and service technologies, in the same way today we are allowed to roam across different network operators with GSM/GPRS cellular devices. In our scenario this transparent roaming exploits variations in the *service condition* point.

3. QOS IN HETEROGENEOUS NETWORKS

In real life, while QoS mechanisms and architectures are growing and Network Operators are trying to plan their infrastructures taking into account QoS constraints, at the same time new network scenarios are coming up for the presence of "*Mobile Terminals*." As technology continues its dramatic progress, we are experiencing the creation of new paradigms and changes in the way technology impacts every day life. Always-on connectivity, location-awareness, and

environment-aware products are among those new paradigms. Smart devices, portable devices, wireless communications, turn up to be the underlying principles of a new revolution in technology. Pervasive computing deals with a broad range of information access methods enabled by mobility, wireless, small embedded systems, and broadband technologies. First QoS models like *Integrated Services* or *Differentiated Services* were designed without taking into consideration mobile nodes. QoS mechanisms enforce a differentiated sharing of bandwidth among services and users. Thus, the introduction of mechanisms to identify traffic flows with different QoS parameters is mandatory, allowing users' charging on the basis of requested quality [2]. Integration of fixed and portable wireless access (*Wireless LAN*, *GPRS*/*UMTS*, *Bluetooth*, . . .) using IP protocol presents a cost effective and an efficient way to provide seamless end-to-end connectivity and ubiquitous access in a market where demands of mobile Internet have grown rapidly and predicted to generate high revenues. But, both cited QoS approaches are limited to stationary hosts and cannot be roughly applied to the mobile environment directly: new paradigms and new architectures must be defined in order to provide the requested QoS in heterogeneous environments.

In this complex scenario, made even more challenging by the extensive use of broadband wireless access, provisioning of QoS guarantees may turn out to be exceedingly difficult. Even the original concept of QoS, as inherited from wireline broadband networks, needs to be revisited to take into account the new challenges that must be faced. Adaptability, automatic connection management, and "soft" QoS requirements are more likely to be managed and useful in this new environment. QoS provision across different wireless access technologies is a key issue that needs to be addressed and solved especially on an end-to-end path that may cross several networks [3]. Therefore, the conflicting requirements of maintaining a high network utilization level, while at the same time keeping network congestion under check (for ensuring a good level of QoS), makes it mandatory to understand at a basic level how to design and control next generation heterogeneous networks [4] and how to reach the same QoS level of the standard wireline network. Hence, the concept of *Service Condition* in a three-dimensional space where QoS requirements depend on the variability of three different technical aspects (terminal, access network, and application) steps from the assumption that the "*old QoS models*" need to be revisited and explained in case of new heterogeneous scenario.

In order to deploy this scenario, subscription of services by means of SLA may help in definition of new network management paradigms. However, SLAs are currently subscribed for only long term service provisioning. There are several factors that cause this limitation using SLAs for only this kind of service:

1. Network infrastructures do not allow a timely service creation;

2. SLA-based contracts are subscribed after a not-automated negotiation phase;
3. Temporal limits of a contract cannot be modified while the service is active;
4. Network configuration of subscribed SLAs typically requires manual interventions.

Hence, current SLAs can be considered as static. The effective implementation of network configuration, usually performed via policy enforcement over network devices [5], is made after a translation of SLA in a more formal and technical document called SLS. While this modus operandi is quite consolidated in traditional wired network, the situation is different in heterogeneous network scenario. In this case, it is difficult to create services using a static SLA negotiation combined with a static SLS configuration. In the next sections we present a new proposal of network management in heterogeneous scenario as well as a dynamic, automatic, and distributed process for SLA/SLS configuration.

4. TRANSPARENT AND DYNAMIC NETWORK CONFIGURATION

As discussed in previous sections, in a heterogeneous scenario a user might utilize different terminals with different capabilities in different situations. For example, a PC may be used at home or inside an office. While walking, a small handset (*advanced mobile phone*) might be more suitable. Finally, a PDA or a laptop will be used when traveling or by telecommuters in different mobile environments. These terminals are different not only in size, but also in processing and communication capabilities. Different applications will also be used in different terminals and they can generally require different QoS values from a network.

In a roaming scenario, an SLA/SLS static negotiation is of marginal utility. For instance, when a roaming user will move from GPRS to WLAN technologies its traffic profile will change too. In this situation network devices configured according to previous configuration could not cope with new requirements: in fact it is highly probable that current traffic profile is not consistent with previous network configuration performed via static SLS. In heterogeneous networks the situation described above can happen frequently because of possible combinations of "*terminal* ↔ *access-network* ↔ *application*" that can determine a de-alignment between "*current traffic profile*" and instantiated SLS.

In this work we consider the configuration process as dynamic rather than static. Following considerations are reported to clarify how the service creation concept via SLA should be modified, especially in case of heterogeneous scenario:

1. The negotiation of a new service is done via SLA subscription and should be performed on a limited number of key variables. The contract just

specifies the service required with no references on how the service has to be effectively implemented. In our vision, beside to the classical parameters (e.g. *QoS level*, *kind of service*, *time scope*, ...), an SLA for an heterogeneous scenario should contain:

a. List of users' devices (*Advanced Mobile Phone*, *PDA*, *Laptop*, *PC Desktop*)
b. List of users' access network technology subscribed (*GPRS*, *UMTS*, *WLAN*, *wired Ethernet*, *ADSL*, *ISDN*, *PSTN*, ...)
c. Indication of used application (*Transactional*, *Multimedia*, *Mission Critical*, ...)
d. E-QoS (*Extended QoS*). With the term E-QoS we mean dependability, up time, security, QoS statistical requirements, QoS soft requirements.

2. The SLA negotiation process is usually time consuming. It is not acceptable in case of negotiation with customers who are end users willing to subscribe simple and on short time scale communication services. A user simply makes a request of a service. Telecom Operators are responsible for the best service accommodation according to current network conditions.
3. In heterogeneous scenario, Telecom Operators configure the service taking into account also the real capabilities of the user's terminal (*PDA*, *laptop*/*notebook*, *PC*, ...) and what performance level his network connection (*WLAN*, *GPRS*, *Bluetooth*, ...) supports.
4. It is possible to specify a level of quality associated to the service required. As an example, an Olympic model can be adopted. The level of the chosen quality affects the final service charge and at first time the user is influenced by service charge/price. This user behavior could drag out some possible and maybe needed future value added service.
5. The operation of network configuration is repeatedly performed by Telecom Operators to dynamically follow variations both in users' traffic and in network load.
6. An SLA is subscribed once and "only" when a customer subscribes himself to the negotiation entity. When a customer changes access network technology, device type, or application, a special entity containing the user traffic profile is able to follow his changes. In this way, this entity follows "*User Service Condition Trend*" traced by user's behavior with respect to presented three-dimensional space (Fig. 2). Just in case of introduction of new variables (new access network, new application, new device, new E-QoS) a new SLA subscription is required.

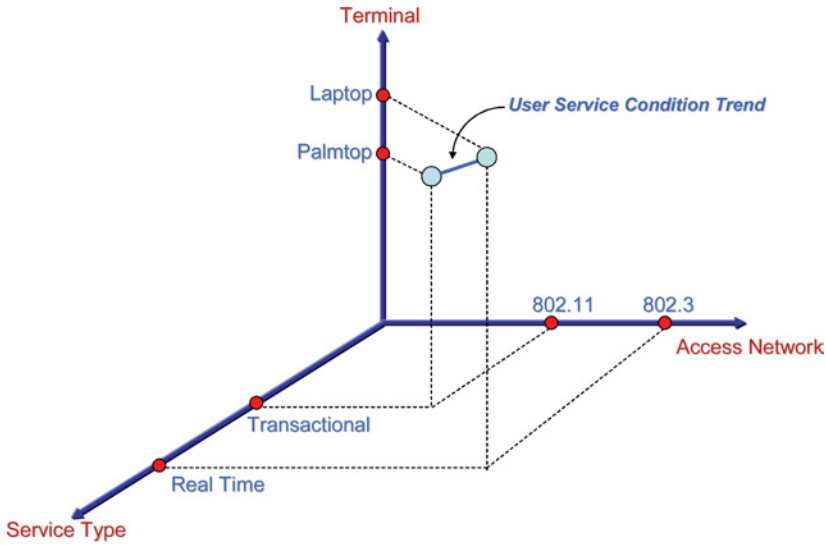


Fig. 2. Example of user service condition trend.

5. MOTIVATION AND RELATED WORK

During the last years the research community has been debating about the most important modifications to be introduced in current infrastructure in order to discipline access to resources on the Internet. Some models propose solutions aimed at optimizing resource allocation, some others simply rely on overprovisioning. In our work we embraced the first philosophy, which we further completed with the introduction of an advance resource reservation scheme. We do believe that such an approach represents an effective solution to the issue of providing a rich portfolio of services with quality assurance.

The CADENUS European Project [1] has defined a proposal of an architecture for the dynamic creation and provisioning of QoS-based communication services on top of Premium IP networks [6], i.e. architectures for the dynamic management of QoS-enabled infrastructures. Such a scenario opens new perspectives in end-to-end services creation because users take an active part in service negotiation. In particular, users subscribe SLA with a mediator for requiring a specified service. While in the past SLAs were just used for regulating interconnections among Network Operators, recently they have become useful for end-to-end service requests. The main result of CADENUS project has been the definition of mediation entities doing single tasks in the complex process of service creation.

The CADENUS service creation framework envisages a scenario where users contact an AM (*Access Mediator*) in order to gain access to a number

of value-added services, by means of negotiation of specific Service Level Agreements. The AM, in turn, needs to interact with one or more Service Mediators, each providing a certain set of services, to retrieve information about the characteristics of the services themselves. Afterwards, it organizes this information in order to let the user choose the service that most appropriately fits his needs.

Once a specific service has been chosen, the involved Service Mediator(s) is (are) in charge of interacting with one or more Resource Mediators, which eventually configure network elements to efficiently satisfy the negotiated requests.

The process described foresees the generation of a number of documents (SLA, SLS, policy rules), each describing the same instance of the service at a different level of abstraction and thus requiring creation/interpretation by the modules (Access Mediator, Service Mediator, Resource Mediator) belonging to the corresponding level of the overall architecture. Digging into the details of such mechanisms, the Service Level Agreement is a contract between the end-user and the Service Mediator, negotiated via mediation of the Access Mediator. Once this contract has been signed, the Service Mediator is in charge of translating it into an appropriate Service Level Specification, containing a technical description of the service itself. This translation is a uni-directional process, requiring some additional information on the SM's side in order to retrieve, where necessary, service-specific data.

The SLS is in turn given to the Resource Mediator, which translates it into a format that is the most appropriate for the QoS-capable network it manages. For example, it might build a list of policy rules, needed inside PDPs (*Policy Decision Points*) in order to configure the underlying network elements (or PEPs – *Policy Enforcement Points*) via a policy protocol like COPS [6, 7].

As far as other proposals, Mellia *et al*. [8] present an analytical approach and a methodology to determine the set of SLAs that can be effectively supported by a Diffserv IP network. In Wang and Schulzrinne [9] it has been developed as a protocol and architecture which enables network service negotiation for multiple delivery services and environments: the RNAP (*Resource Negotiation and Pricing Protocol*) enables service negotiation between user applications and the access network, as well as between adjoining network domains. The work is mainly focused on pricing issues and there aren't precise references both to SLA and SLS negotiation and reconfiguration. DSNP (*Dynamic Service Negotiation Protocol*) [10] is a protocol to negotiate the SLS at IP layer. It can be used for service negotiation from host to network, network to host, and network to network. DSNP, as our architecture, can be used in both wireline and wireless networks. Our work is strictly coupled to this. The protocol we designed, as presented in next sections, steps from DSNP but it is designed to face off the complex scenario of heterogeneous networks. Furthermore, DSNP does not map SLA into correspondent SLS: the paper presents only SLS negotiation without

taking into account SLAs. In this way it is not possible to map user needs on network condition.

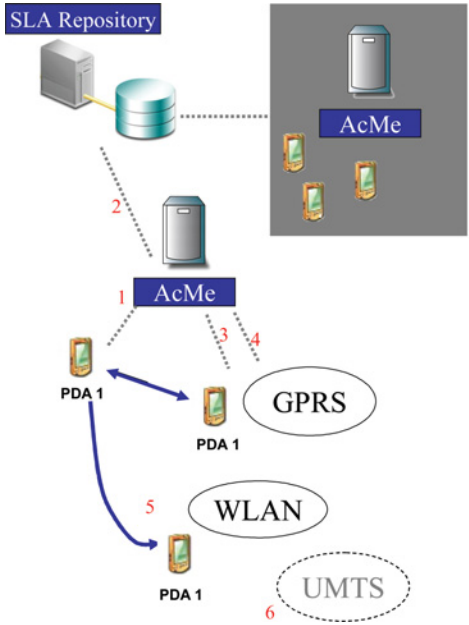
In Nagarajan [11], a simple case study describes the need for simulation in effective SLA documentation and SLA monitoring. It analyzes different scenarios within SLA levels using ARENA simulation software and it demonstrates how it helps in identifying end user services and in satisfying customer expectations. This simulation was useful for definition of a real implementation of our framework. The proposal presented in Czajkowski [12] is linked to a generalized resource management model in which resource interactions are mapped onto a well defined set of platform-independent SLAs. This model is used in the *Service Negotiation and Acquisition Protocol* (SNAP) which provides lifetime management and an at-most-once creation semantics for remote SLAs. Unfortunately, a concrete implementation of the SNAP model is missing.

As far as network management activities carried out using an Active Network platform, in Raz and Shavitt (2000) [13] and Raz and Shavitt (2001) [14] it is presented as a work that describes how active techniques can be used to allow fast and easy deployment of distributed network management applications in IP networks. A prototype system where legacy routers are enhanced with an additional active engine is presented. Marshall *et al*. [15] present an architecture for an active network based management solution for multiservice networking. In Marshall and Roadknight (2001) [16] a novel approach to quality of service control in an active service network is described whereas Marshall and Roadknight (2000) [17] presents an autonomous adaptive control agent for dynamic servers in an active network. Finally, a practical dump of active service creation via SLA negotiation is described in D'Arienzo *et al*. [18].

Our contribution is mainly related to the introduction of active functionality in the network management plane to address dynamic behavior imposed by heterogeneous networks. In particular, we adopted the CADENUS architecture as the reference infrastructure for high level service negotiation, and we propose a new entity capable to follow dynamic changes of a Service Condition trend, as we discuss in next sections.

6. FROM A STATIC TO A DYNAMIC MODEL OF RESOURCE MANAGEMENT

From the experience of CADENUS project, we propose the introduction of new functionalities at management plane to address new complex requirements of heterogeneous networks. In particular, we present an innovative entity, namely *AcMe* (*Active Mediator*), which is able to manage network resources according to high level information: the AcMe is an Active and innovative version of Resource Mediator implemented in CADENUS project. The need for an innovative entity is due to the wide range of resources to be managed in a heterogeneous scenario,



1. SLA subscription. Parameters: Device, Access Network, Application, E-QoS
2. SLA saved in repository.
3. Telecom Operator using AcMe computes the best SLS and enforces Network Configuration.
4. Telecom Operator (using the AcMe) monitors user's traffic. In case of need (altered network conditions, altered user's profile, ...) computes a new SLS and makes a new Network Configuration.
   Following user traffic profile the AcMe follows the "***User Service Condition Trend***" traced by the user behavior in our three-dimensional space. This step is consistently repeated.
5. AcMe is capable to ensure SLA according to subscribed parameters: it is capable to ensure handoff among different access networks and among different devices. In this case. SLS is automatically re-configured.
6. When one of key variable is changed (access network, application, device, E-QoS) a new SLA subscription is required.

Fig. 3. AcMe activities in heterogeneous scenarios.

where resources mean all key points of the service condition definition. AcMe is needed in order to simplify network configuration process and to control scalability effects.

In Fig. 3 a high level description of AcMe functionality is reported. AcMe has been introduced to allow a trade-off between Telecom Operators' revenue and users' satisfaction.

From the Telecom Operator point of view, the search of an optimal balance between maximizing its resources usage and its profits must be made with respect to the negotiated SLAs. For example, mobile phone telecom operators of GSM networks have made available a new service: "*profile finder*." On the basis of user indication (i.e. total amount of calling period, busy hour factor, destination number, . . .) the operator can show the most appropriate SLA to the user. This operation is static and is carried out with the user cooperation: in the mobile phone world this situation is corresponding to a change of prepaid card or license fee. The user must indicate its traffic characteristics (*traffic profile*) or "*that one he thinks to be his traffic profile*." This mechanism is not transparent to the user and if the user changes its traffic characteristics he will be asked to explicitly change the contract as well. Anyway, over GSM network, the participation of the user to the contract compilation is possible because the parameters are simple for the user and

the negotiation phase is simpler: in an innovative scenario over IP heterogeneous networks, where new value added services are present, the decisional process can be difficult for a common user.

Moreover, for the sake of optimizing the network infrastructure a Telecom Operator could change the present or future negotiated SLA because network load is changed. For this reason a Telecom Operator should prefer a dynamic way to implement its SLS according to the subscribed SLA. Finally, in the new heterogeneous broadband access network it is very hard for a new Telecom Operator or Autonomous System manager to calculate the right number of users and consequently, for example, the right bandwidth amount. Wi-Fi, 3G networks, and Mobile IP are coming in play and in this scenario the number of users is variable and uncountable. Situations where network resources provisioning is sometimes unpredictable are very frequent today: in a train station, bus station, and airport when considerable bursts of passengers arrive simultaneously; a special event in a city (music concerts, political events, ...) may gather many unexpected network users. Due to mobility, the provisioning of network resources may not be accurate for actual demand.

In order to understand the benefits and novelty of our architecture we can compare the following "output variables" with and without AcMe using HNMP:

1. the total cost of configuration and reconfiguration;
2. number of admitted SLAs;
3. QoS perceived from users;
4. network resources utilization.

These four variables represent a crucial aspect in network planning and management processes. As previously introduced, the AcMe has been designed in order to take into account the kind of user terminal (i.e. *CC/PP* [19]), the kind of access network, and finally the kind of user application (e.g. using of RTP [20] in the case of real time applications).

The architecture has one AcMe for each domain. For each domain the proper AcMe is responsible for all different access network technologies. In our scenario, a user compiles a simple SLA. According to this request, the AcMe, checking the relative resource bundle and network condition, can accept or reject the request. If the response is positive the compiled SLA will be subscribed. In general, AcMe can use a statistical overbooking model where it is possible to accept a number of sessions calculated both on a worst case and on the network bottleneck as reported in Mellia *et al*. [8].

The AcMe configures its network device according to negotiated SLAs in order to ensure correspondent SLSs: this process is made on two steps, SLS configuration and SLS enforcement. After these steps, network traffic needs

to be monitored. In particular, at each probing interval, the AcMe listens to:

- User traffic profile
- User device
- Access network
- Application requirements (Transactional, Real Time, . . .)
- Network conditions by means of polling on selected parameters

Stemming from this retrieved information, the AcMe is capable of checking the network configuration and, if necessary, it is capable of beginning a reconfiguration process. To limit time-consuming probing operations, if the observed user's behavior is quite constant during the initial probing interval, the AcMe will enlarge this interval assigned to the user, thus reducing probing requests. When AcMe senses for reasonable jitter in traffic envelope, it will provide for a soft and seamless to the user SLS renegotiation. Just in case of users who generate a traffic figure definitely out of profile, the AcMe will close the connection and it will contact the user to propose a new SLA subscription.

7. THE HETEROGENEOUS NETWORK MANAGEMENT PROTOCOL (HNMP)

Figure 4 presents an architectural model view for dynamic services deployment. The Active Mediator contains all details on supported access networks and on network devices present in the domain. Dotted lines represent interaction among entities.

While SLA negotiation phase has been addressed in CADENUS project, here we focus the attention on effective adaptation to dynamic changes of SLS implementation in case of heterogeneous scenario. More precisely, in order to ensure a correct management of heterogeneous network, a new protocol has been defined: HNMP, *Heterogeneous Network Management Protocol*. The HNMP steps from the work presented in Chen *et al*. [10] and acts among the following entities:

1. SLA Repository $\leftrightarrow$ AcMe
2. AcMe $\leftrightarrow$ AcMe (intra-AcMe communications)
3. AcMe $\leftrightarrow$ Network Devices
4. AcMe $\leftrightarrow$ AcMe (inter-AcMe communications)

We start from the assumption that the SLA negotiation phase has already been performed, the SLA has been subscribed, and finally the SLA has been successfully stored in the SLA repository. Furthermore, inter-AcMe messages are related to an interdomain scenario, whereas intra-AcMe messages represent a way for

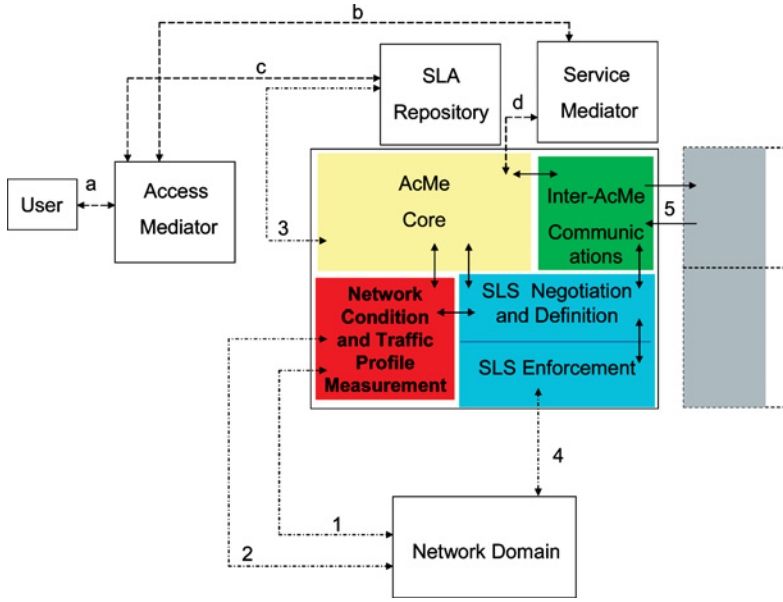


Fig. 4. AcMe architecture and HNMP messages.

implementing itself as a distributed element. In the following there is a description of protocol messages and AcMe interactions.

1. SLA repository ↔ AcMe
   Check_user_request: This message is sent to check if a user has correctly subscribed an SLA, and if it is regularly stored in the repository.
   Check_user_response: This message contains the answer to the Check_user_request: User is granted or not, what kind of service he can receive, and all other details reported in Section 6. The AcMe (core) collects the necessary information and stores it for further use.
2. AcMe ↔ AcMe (intra-AcMe communications)
   Digging into details of AcMe bricks, it is composed of four modules that can either be collapsed in a single box or distributed inside network domains. Following the notation reported in Fig. 3, the four bricks are:

   - AcMe Core
   - Network Condition and Traffic Profile Measurement (Probing module)
   - SLS Negotiation & Definition and SLS enforcement (SLS module)
   - Inter-AcMe communications

   AcMe Core and SLS module interact through two kinds of messages:

SLS_negotiation_request: Following the indication of Check_user_reply, a request for SLS negotiation is sent. This message is usually invoked inside the AcMe to request for a particular SLS negotiation. This message is sent at first negotiation time as well as when a network (re-)configuration is needed or the user traffic profile is changed. Finally, if the AcMe wants to forcefully terminate an SLS, it will send an SLS_negotiation_request message with appropriate fields set to particular code related to terminating motivation.
SLS_negotiation_response: This message is sent in response to the SLS_negotiation_request. This message indicates whether the requested SLS is accepted or rejected. If the requested SLS is not accepted, then the reason of rejection is provided. For example, if the network device does not accept the SLS of a user due to lack of resources, it will send back a response indicating a reject along with the list of SLS that could be supported. It is important to underline that for a single instance of an SLA, there could be more "SLS_negotiation_request – SLS_negotiation_response" messages.

As far as interaction between AcMe Core and Probing module they exchange the following messages:

Probing_request: This message solicits the probing module for collecting information about network status and user's traffic profile. Hence, a Probing_request message is usually followed by both Network_status and User_Traffic_profile messages.
Probing_response: When requested information is definitely received from network devices, it is sent up to the AcMe Core and stored for future computations.

3. AcMe ↔ Network Device
SLS_configuration_enforce: After the SLS_negotiation_response message, this message is sent by AcMe to network devices using the parameters present in this last message. When an SLS must be released this message is sent with an appropriate field set to zero.
Network_status_request: This message is sent by an AcMe to network devices asking for a feedback on the statistics of its current usage and—in general—on its state. The AcMe could ask for statistics on parameters like packet loss, throughput, average delay, jitter, and total number of octets sent from/forwarded to the controlled network.
Network_status_response: This message is sent by network devices in response to a Network_status_request message. The AcMe collects the necessary information and stores it for further use. In a more dynamic scenario, this message could be sent without solicitations when network resources become scarce. After this message, it is probable that SLS_negotiation_request message will start containing more high costs.

Similarly, when there are unused resources available, network devices send this message to AcMe. After this message, it is possible to offer SLS at a lower price.
User_traffic_profile_request: This message is sent by an AcMe to network devices asking for a feedback on which users are currently connected to the network, and what kind of traffic they generate.
User_traffic_profile_response: This message is sent by network devices in response to a User_traffic_profile_request message.

4. AcMe ↔ AcMe (inter-AcMe)
Messages of this class are used in a multidomain scenario. In this paper we present the interaction in a single domain. See the conclusion section for inter-domain issues.
In Fig. 4, the line *1* is related to a Network_status message whereas the line *2* is related to a User_traffic_profile message. In this last case, when a traffic profile is analyzed, it is necessary to have an interaction with the SLA Repository in order to check the SLA negotiated (line *3*). Line *4* is related to SLS enforcement on all network devices involved in service implementation. Lines a, b, c, and d are typical of a CADENUS scenario and they represent the interaction among mediators in order to ensure service negotiation in a scenario where an AcMe entity is present. More precisely, lines a, b, c, and d are related to SLA negotiation and subscription phase. Lines *5* denote inter-domain AcMes communications.

8. AcMe PROTOTYPE AND EXPERIMENTAL RESULTS

AcMe implements HNMP in order to accomplish a proactive network configuration that is able to support a dynamic service management. To demonstrate AcMe functionalities, we made use of an experimental testbed that reproduces (on a small scale) a real scenario made by a single network domain where users of a corporate network exploit a service offered by a single service provider. We implemented an AcMe prototype that manages single network domains where gateways (ingress and egress routers) are based on active network technology. Notice that this approach requires the introduction of active nodes only on boundary nodes [21].

As represented in Fig. 5, AcMe pertains to a network domain and accomplishes the task of local domain devices (re-)configuration in order to manage internal domain resources. It directly interacts with network devices managing boundary nodes according to HNMP messages. We make the following assumptions:

1. users have to subscribe SLAs to request end-to-end services;
2. each network domain supports traffic accepted by its ingress routers.

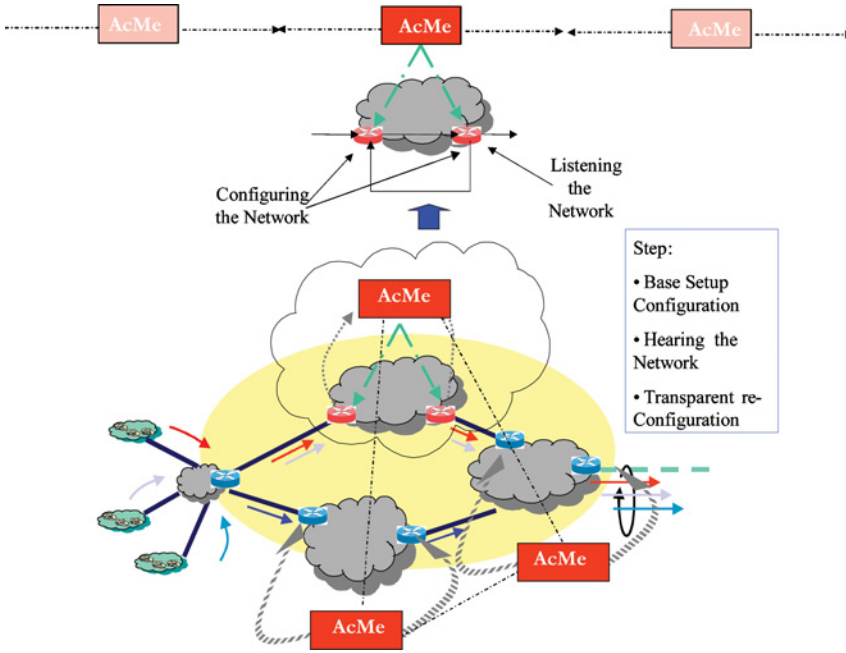


Fig. 5. AcMe activities.

AcMe simply operates on ingress routers to act as an implicit admission control [22]. Once the flows enter the network, they must be propagated inside the rest of the network according to the quality of requested service. Multiple AcMe managing network domains crossed by flows take into account the number and the class of service of these flows and consistently configure the domain without any knowledge of what happens in near domains. Notice that interaction between AcMes of different domains is not needed for inter-domain communications. This behavior helps in limiting scalability problems, both at the edge and the core of the network. In fact, it is self evident that when the number of users increases, many different requirements (SLAs) come into play. Considering a separate SLA for each different requirement coming from each user would cause a big SLS jam to be accommodated in each network domain. Using AcMe prototype, each network domain is managed in an independent way: each AcMe decides the best way SLAs have to be allocated inside its domain, and eventually could decide to aggregate more SLAs in fewer SLSs.

Obviously, this distribution of network resources inside the domain will be managed by the AcMe on the basis of the customer's contract. For instance, customers who have subscribed for higher quality SLAs (e.g., Gold) are preferred

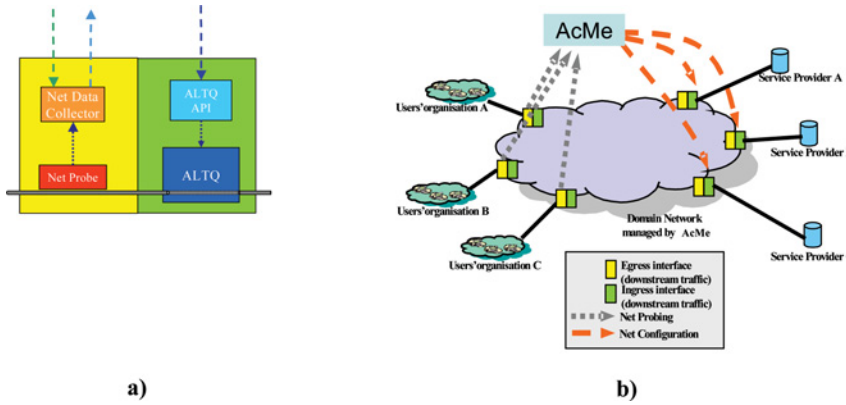


Fig. 6. Explosion of AcMe activities in experimental testbed scheme.

to lower quality contract (e.g. Silver). Hence the AcMe performs network probing and commands (re-)configuration of specific network entities.

In each domain, the AcMe works two times: in a first time it sends probing requests (Network_status_request messages) to its network resources and waits for collected information concerning current traffic load (Network_status_response messages). In a second time, the *closed loop control* comes into play. The AcMe performs evaluation on received data and, if needed, it makes a new SLS computation (SLS_negotiation_request) and the final enforcement (SLS_configuration_enforce): this new configuration is sent to correct network devices.

Digging into details of a single domain, Fig. 6 describes main components of the proposed architecture.

Routers have network interfaces to act as ingress/egress router of a network domain. The yellow part represents the egress network interface whereas the green part is the ingress network interface. Each single router is an Open platform based on Programmable Network Paradigm [23]. During its design, we followed the Elastic Network [24] approach. The Elastic Networks aim at getting over the limitations of the opposite traditional models (OCN – *Open Control Networks* and AN – *Active Networks*) by proposing an OCN platform which has the flexibility proper of the AN architecture, in particular it has to provide support for the code installation. However, it is not supposed to perform computations on data path, thus, the whole node performances are not compromised.

In our Elastic Node prototype, operations on network interfaces like probing and Traffic Control configuration can be performed remotely. The prototype has been implemented on a FreeBSD environment with ALTQ Traffic Control Module enabled [25]. The executed operations are:

1. After the initial-static network configuration, the network provider, by means of HNMP carried out by AcMe, consistently probes router network

interfaces (inside Probing_request/Probing_response cycle). In the example shown, the network provider probes the *Egress* network interfaces of its domain (the nearest to the user) to understand the real traffic profile of the user (User_traffic_profile_request/User_traffic_profile_response).

2. In case this profile is significantly different from that previously configured, the provider performs a new network configuration on the *Ingress* router interfaces (one nearest to the service provider) by means of SLS_negotiation_request – SLS_negotiation_response – SLS_configuration_enforce messages sequence. Obviously, this distribution of network resources inside the domain will be managed by the AcMe, based on the user subscription, for instance, one who has subscribed for higher quality SLAs (e.g., Gold better than Silver).

In this work we show the capabilities of Probing module with regard to collected information concerning an HTTP service using inferred TCP protocol information [26, 27]. Hence, the testbed is composed of a domain positioned between several HTTP clients and a Web server, with traffic generated using an HTTP client request generator [28] and the cross traffic generated using a synthetic traffic generator called D-ITG (*Distributed Internet Traffic Generator*) [29]. Initially, configuration of Ingress network interface does not allow the download of web traffic. As drawn in Fig. 7(a), at time 20 s the client starts requesting HTTP pages, but no pages are downloaded until time of 35 s. This interval is related to the time when the network is probed, and we call this as the *probing interval*. At 35 s, the AcMe collects the information probed from the Egress network interface and compares it with traffic profile of users' organization (Probing_request – Probing_response cycle). After checking network resources, the Ingress network interface of the router is then updated with the new configuration (SLS_negotiation and SLS_configuration messages). Clients are then able to download the content.

In Fig. 7(b) the situation reproduces a change over time in client requirements. At about 135 s, client requirement increases (seen as increase in the number of HTTP requests). The AcMe updates this information within the probing interval (i.e.15 s), and performs a refresh in network configuration to meet the new requirements of the user. These trials are simple demonstration of automatic and transparent network reconfiguration using active capabilities on boundary routers.

9. CONCLUSIONS

The paper has presented a framework for management of services in heterogeneous networks. After the introduction of the *Service Condition* concept, which is useful for definition of users' requirement in a heterogeneous context, we introduced a network entity called *AcMe*. Besides, a new protocol, HNMP, has been expressly designed to automatically manage dynamic changes in users' Service Conditions. *AcMes* are independent management entities orchestrating operation

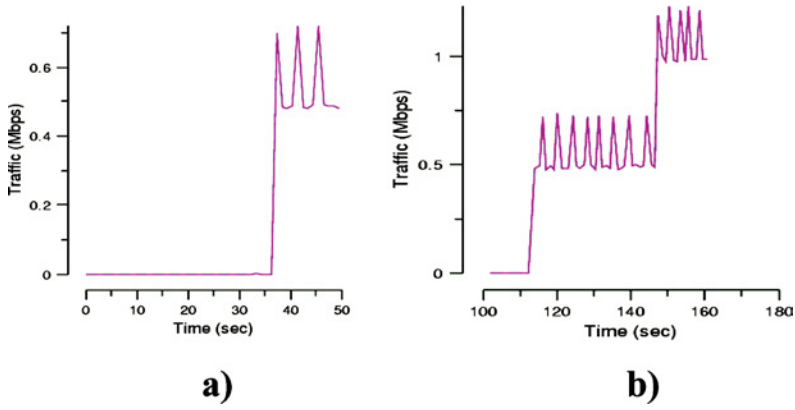


Fig. 7. Experimental results.

of a single network domain. In this way, end-to-end service management is done with limited scalability problems. An experimental testbed to check basic functionality has been implemented as a proof of concept. A more complex testbed, together with precise performance evaluations of proposed solution, will be the subject of our future studies and work. Anyway, we would underline here the importance of framework presented in terms of scalability. The *AcMe* architecture limits scalability issues because message volume is independent of the route hop counts or the number of transit domains on the path. We considered a solution based on a per-domain management better scalable than centralized ones since the state information must be saved only with respect to single domains. In our ongoing work, we are considering the interaction among multiple *AcMes* across a complete, multidomain end-to-end path.

ACKNOWLEDGMENT

Research outlined in this paper was initially funded by the IST project CADENUS IST-1999-11017 "Creation and Deployment of End-User Services in Premium IP Networks." Currently, it is under the financial support of the "Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR)" in the framework of the FIRB Project "Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)."

REFERENCES

1. G. Cortese, R. Fiutem, P. Cremonese, S. D'Antonio, M. Esposito, S. P. Romano, and A. Dioconescu, CADENUS: Creation and deployment of end-user services in premium IP networks, *IEEE Communications Magazine*, Vol. 41, No. 1, pp. 54–69, January 2003.

2. S. Giordano, M. Potts, M. Smirnov, and G. Ventre, Advances in QoS, *IEEE Communications Magazine*, Vol. 41, No. 1, pp. 28–29, January 2003.
3. A. Iera, A. Molinaro, and K. Nahrstedt, QoS in next-generation wireless multimedia communications systems, *IEEE Wireless Communications*, Vol. 10, No. 3, pp. 6–7, June 2003.
4. R. Chakravorty, J. Crowcroft, I. Pratt, and M. D'Arienzo, A framework for dynamic SLA-based QoS control for UMTS, *IEEE Wireless Communications Magazine*, October 2003, Special Issue on Merging IP and Wireless Networks, Vol. 10, No. 5, pp. 30–37, October 2003.
5. J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, and A Sastry, The COPS (Common Open Policy Service) Protocol, RFC 2748, January 2000.
6. S. Giordano, S. Salsano, S. Van den Berghe, G. Ventre, and D. Giannakopoulos, Advanced QoS provisioning in IP networks: The European premium IP projects, *IEEE Communications Magazine*, Vol. 41, No. 1, pp. 30–36, January 2003.
7. K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith, COPS Usage for Policy Provisioning (COPS-PR), Standards Track RFC 3084, IETF, March 2001.
8. M. Mellia, C. Casetti, G. Mardente, and M. Ajmone Marsan, An Analytical Framework for SLA Admission Control in a DiffServ Domain, INFOCOM 2003.
9. X. Wang and H. Schulzrinne, RNAP: A resource negotiation and pricing protocol, In *Proceedings of (NOSSDAV)*, pp. 77–93, June 1999.
10. J. C. Chen, A. McAuley, V. Sarangan, S. Baba, and Y. Ohba, Dynamic Service Negotiation Protocol (DSNP) and Wireless DiffServ, ICC'02, New York City, April 2002.
11. K. V. Nagarajan, G. Awyzio, P. Vial, Modelling and simulation of an alarm based network management system for effective SLA monitoring and management, SCI2003, Orlando, FL, August 2003.
12. K. Czajkowski, I. Foster, C. Kesselman, V. Sander, and S. Tuecke, SNAP: A protocol for negotiating service level agreements and coordinating resource management in distributed systems, *Proceedings of the 8th Workshop on Job Scheduling Strategies for Parallel Processing*, Edinburgh, Scotland, UK, July 2002.
13. D. Raz and Y. Shavitt, Active networks for efficient distributed network management, *IEEE Communications Magazine*, Vol. 38, No. 3, pp. 138–143, March 2000.
14. D. Raz and Y. Shavitt, Towards efficient distributed network management, *Journal of Network and Systems Management*, pp. 347–361, September 2001.
15. I. W. Marshall, J. Hardwicke, H. Gharib, M. Fisher, and P. Mckee, Active management of multiservice networks. In *Proceedings of NOMS*, IEEE, Hawaii, pp. 981–982, 2000.
16. I. W. Marshall and C. M. Roadknight, Provision of quality of service for active services, *Computer Networks*, Vol. 36, No. 1, pp. 75–87, June 2001.
17. I. W. Marshall and C. M. Roadknight, Adaptive management of an active services network, *BT Technology Journal*, special issue on "Biologically inspired computing," Vol. 18, No. 4, pp. 78–84, October 2000.
18. M. D'Arienzo, M. Esposito, S. P. Romano, and G. Ventre, Dynamic SLA-Based Management of Virtual Private Networks, In Proceedings of IWDC 2001, LNCS 2170, September 2001.
19. G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran, Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies, W3C Working Draft, March 25, 2003. (http://www.w3.org/mobile/CCPP/, accessed october 2003)
20. IETF Audio-Video Transport Working Group, RTP: A Transport Protocol for Real-Time Applications, IETF Proposed Standard RFC1889, January 1996.
21. R. Boutaba and A. Polyrakis, Projecting advance enterprise network and service management to active networks, *IEEE network*, pp. 28–33, January/February 2002.
22. C. Cetinkaya, V. Kanodia, and Edward W. Knightly, Scalable services via egress admission control, *IEEE Transactions on Multimedia*, Vol. 3, No. 1, p. 69, March 2001.

23. A. T. Campbell, H. G. DeMeer, M. E. Kouvanis, K. Miki, J. B. Vicente, and D. Villela, A survey of programmable networks, *Computer Communication Review*, Vol. 29, No. 2, pp. 7–23, April 1999.
24. H. Bos, R. Isaacs, R. Mortier, and I. Leslie, Elastic network control: An alternative to active networks, *Journal of Communications and Networks*, Special Issue on Programmable Routers and Switches, Vol. 3, No. 2, March 2001.
25. K. Cho, Managing traffic with ALTQ, In *Proceedings of USENIX 1999, Annual Technical Conference*, FREENIX Track, Monterey, CA.
26. F. D. Smith, F. H. Campos, K. Jeffay, and D. Ott, What TCP/IP Protocol Can Tell Us About the Web, ACM SIGMETRICS, 2001.
27. F. H. Campos, K. Jeffay, and F. D. Smith, What TCP/IP Protocol Headers Can Tell Us About the Web – Part II Trends, Issuues and More, SIGCOMM Measurement Workshop, 2001.
28. D. Mosberger and T. J. Httperf, A tool for measuring web server performance. (http://www.hpl.hp.com/personal/David Mosberger/httperf.html. hewlettpackard research labs)
29. A. Pescapè, D. Emma, and S. Avallone, D-ITG – Distributed Internet Traffic Generator. (www.grid.unina.it/software/ITG)

Maurizio D'Arienzo has a post-doc position at Computer Engineering and Systems Department of the University of Napoli Federico II. After a Laurea Degree in Electronic Engineering from the University of Napoli Federico II, he got a PhD in Computer Science Engineering from the same University. His main research interests are focused around networking area and in particular with respect to distributed systems, network management, and QoS in IP networks.

Antonio Pescapè is a PhD Student at Computer Engineering and Systems Department of the University of Napoli Federico II. He earned a Laurea Degree in Computer Engineering from the same institution. His research interests are in the networking field with focus on models and infrastructure for QoS over IP networks, models and algorithms for Internet Traffic, and Network Measuring and Management.

Giorgio Ventre is Professor of Computer Networks in the Computer Engineering and Systems Department of the University of Napoli Federico II. He earned a Laurea Degree in Electronic Engineering and a PhD in Computer Engineering, both from University of Napoli Federico II. As leader of the networking research group, he is the principal investigator for a number of national and international research projects. He has coauthored more than 100 publications and is a member of the IEEE Computer Society and of the ACM.