Ann. Telecommun. (2010) 65:713–721
DOI 10.1007/s12243-010-0194-y

Network virtualization for cloud computing

Fabio Baroncelli · Barbara Martini · Piero Castoldi

Received: 18 September 2009 / Accepted: 15 July 2010 / Published online: 29 July 2010

Abstract Cloud computing enables a transparent access to information technology (IT) services such that the users do not need to know the location and characteristics of the relevant resources. While IT resource virtualization and service abstraction have been widely investigated, data transport within the cloud and its efficient control have not received much attention in the technical literature. In fact, connectivity is, itself, a service that contributes to the overall performance of the cloud. This paper introduces a novel classification of the Network as a Service (NaaS) such that it can be orchestrated with other cloud services. Then, it proposes a network virtualization platform (NVP) as the mediation layer able to provide NaaS to cloud computing by exploiting the functionality provided by control plane (CP)-enabled networks. In particular, the proposed NVP maps the end-point addresses and perceived Quality of Service parameters of a NaaS requests in the parameters characterizing the connectivity as viewed by transport networks using the information obtained from the CP at the boundary of the network. The NVP uses these parameters to fulfill connectivity requests to the CP. Finally, this paper presents a complete design from both the software implementation and network signaling perspective of two use cases in which NaaS is involved as stand-alone facility for the connectivity service provisioning or is combined with other cloud services for a storage service provisioning.

Keywords Network virtualization · Cloud computing · Networking

F. Baroncelli (✉) · B. Martini
CNIT,
Pisa, Italy
e-mail: fabio.baroncelli@cnit.it

B. Martini
e-mail: barbara.martini@cnit.it

P. Castoldi
Scuola Superiore Sant'Anna,
Pisa, Italy
e-mail: castoldi@sssup.it

1 Introduction

Cloud computing is a novel paradigm to share resources (e.g., servers, storage devices, desktops, and applications) over the Internet. It is based on the concept of resource virtualization, i.e. on the hiding of implementation-specific details of resources and on their exposure to end users in an integrated way [1]. Currently, a cloud can provide, broadly speaking, three different classes of service: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS provisions hardware, software, and equipments to deliver software application environments with a resource usage-based pricing model. PaaS offers a high-level integrated environment to build, test, and deploy custom applications. SaaS delivers special-purpose software that is remotely accessible by consumers through the Internet, again with a usage-based pricing model [1].

Although some economic benefits of cloud computing are already universally accepted, the success of this paradigm is bound to the level of accessibility, control, reliability, and security assured to its users. In particular, the cloud should be conceived as a multi-provider environment able to offer the capability to migrate a service from one provider to another, to localize the best resource not only in terms of computational or storage capacity but also of connectivity bandwidth and relevant connectivity delay. In fact, advanced IT services require a strong integration with

connectivity services offered by the network infrastructures, in particular when users and service providers are connected through a wide area network, where data transfer performance is hard to be guaranteed. Network services, offering transport network connectivity with a level of virtualization suitable to be invoked by cloud users, are named in this paper Network as a Service (NaaS).

NaaS is unable to be directly provided by current network infrastructures without the mediation of an additional network functionality named in this paper network resource provisioning system (NRPS). In fact, although networks may automate the connection provisioning to client nodes, thanks to the functionality offered by their control plane (CP) via user-to-network interface (UNI), NaaS requests use network resource parameters expressed in a different grain and level of abstraction as regards the parameters characterizing UNI-based connectivity requests.

Several NRPSs have been proposed in literature. These NRPSs generally follow two different approaches named directed signaling, where applications talk directly to the network CP, and undirected signaling, where applications talk to a centralized network entity acting as a service broker [2].

Since standard CPs do not support advanced reservation and negotiation functionalities, the directed approach generally requires an extension of the UNI. Examples of directed signaling NRPS are the architectures proposed by the MUPBED (http://www.ist-mupbed.org) and the FEDERICA (http://www.fp7-federica.eu) projects. In MUPBED, grid applications request Quality of Service (QoS)-enabled connectivity to the generalized multiprotocol label switching (GMPLS) CP thanks to a new adapter interface named grid UNI. The FEDERICA project proposes an infrastructure for future Internet research that creates "slices", i.e., a set of (virtual) network and computing resources monitored and managed by a dedicated CP. The CP integrates and extends the CPs deployed at the different layers of the network stack. The applications request to the CP the services provided by the slices via a dedicated UNI.

The undirected approach is more suitable for multi-domain and multi-provider environment where not all the domains are CP enabled. Examples of undirected signaling NRPS are the architectures proposed by the CARRIOCAS (http://www.carriocas.org) and the G-lambda (http://www.g-lambda.net) projects. The CARRIOCAS proposes a centralized entity named scheduling reconfiguration virtualization as a functional block that mediates between the connectivity request issued by grid application and the network management system. G-lambda establishes a standard web services interface (GNS-WSI) between grid resource manager and network resource manager provided by different network operators.

The PHOSPHORUS (http://www.ist-phosphorus.eu) and DRAGON [3] are examples of projects adopting a hybrid approach. PHOSPHORUS demonstrates on-demand service delivery across multi-domain/multi-vendor research network test-beds on a European and worldwide scale. It integrates application middleware and transport networks using three planes: the service plane that exposes API to applications to invoke connectivity services (undirected approach), the network resource provisioning plane that adapts connectivity requests to the existing NRPSs and the CP that enhances the standard GMPLS to provide optical network resources to grid (directed approach). In DRAGON, the directed signaling is accomplished using a modified version of OSPF-TE that incorporates domain abstraction and hierarchical routing techniques to provide inter-domain traffic engineering (TE) routing. For each administrative domain, the undirected signaling is carried out using a management agent named network-aware resource broker (NARB).

Both the directed and undirected approaches require applications to request connectivity among the NRPS nodes that generally are not end user hosts. This implies the applications to be aware of the NRPS address space and attributes. In a public network's scenario, these information are generally hidden to applications.

For this reason, this paper proposes the network virtualization platform (NVP), developed by the authors in their previous works [4–6], as the mediation layer able to provide NaaS to cloud computing by exploiting the functionality provided by CP-enabled networks. The NVP adopts a different approach that may be called undirected signaling to the CP. In fact, an application requests end-to-end (e2e) network services in terms of end user addresses and perceived QoS parameters (e.g., the transfer delay and the guaranteed bit rate) to the service elements controlling the edge node of the transport network serving its access networks without any knowledge of the transport network infrastructure. Thanks to a service signaling protocol, a distributed set of service elements collaborates for collecting and correlating network status information about both the transport and the access networks. The collected information are used to map the applications connectivity requests into a set of CP directives.

Moreover, the paper introduces a novel classification of the NaaS and presents two use cases for proof of concept conceived to show how the NaaS is composed and orchestrated to provide connectivity on demand to cloud users (Connectivity Service Provisioning use case) or to enhance existing cloud services (Storage as a Service use case). The flowchart of each use case shows how the mapping process between an application connectivity request and the set of CP directives, performed by the NVP, is hidden to the cloud and, at the same time, does not require any modification to the network infrastructure.

The rest of the paper is structured as follows. Section 2 introduces the NVP requirements and its architecture. In Section 3, NaaS characteristics and functionalities are de-

scribed. Sections 4 and 5 present two use cases that illustrate a connectivity and storage as a service provisioning respectively. Section 6 discusses the main architectural advantage of the proposed approach. Section 7 provides some conclusions.

2 Network scenario and architectural issues

The network scenario treated in this paper is represented in Fig. 1. A set of users residing in different access networks are connected to the Internet through access points (APs). The cloud infrastructure is composed by a set of servers (e.g., media server, storage server, and application server) interconnected over the Internet.

The cloud services are provisioned to end users, thanks to dedicated connections among APs with assured QoS (bandwidth, delay, etc.). The type of connectivity is related to the type of service provided by the cloud. In particular, the connectivity may be point-to-point (p2p), for example, in the case of a Storage as a Service between user A and the storage server; point-to-multipoint (p2m), for example, in the case of a television on-demand service that involves the media server and users B and C; and multipoint-to-multipoint (m2m), for example, in the case of a video conference among users A, B, and C.

To provide connectivity on demand, a specific entity, namely the NVP, is considered within the cloud computing architecture, as shown in Fig. 2. The NVP design is driven by the following requirements [7]:

- It should hide the implementation details about the network infrastructure;
- It should not cause any change in the hardware or software of end-systems;
- It should not change the Internet infrastructure;
- It should minimize the number of network nodes involved in the NaaS provisioning;
- It should provide QoS continuity along multiple provider domains; and
- It should be incrementally deployable.

For the design of NVP for clouds, we deploy the service oriented network architecture model [4] that has been widely validated also through an experimental prototype. The NVP prototype was used to deploy connectivity on-demand and video on-demand services in a grid and NGN context. Relevant technical details and numerical results are presented in [5, 6]. In particular, in [4] it has been demonstrated that the taken approach is scalable since the network size does not significantly degrade the performance of the system, especially the service provisioning latency.

As shown in Fig. 2, the NVP is composed by a set of distributed entities (DEs) and a centralized entity (CE). In particular, the CE implements a database with the customer profiles and relevant service level agreement (SLA) for the end user authentication and relevant network service authorization. Since network resource information, e.g., the availability of a storage server, changes dynamically, an efficient resource discovery mechanism is a fundamental requirement for the NVP. There are three important scale parameters to consider when designing a resource discovery service: (1) the rate of updates to the database, (2) the state required to be held by the mapping service, and (3) the

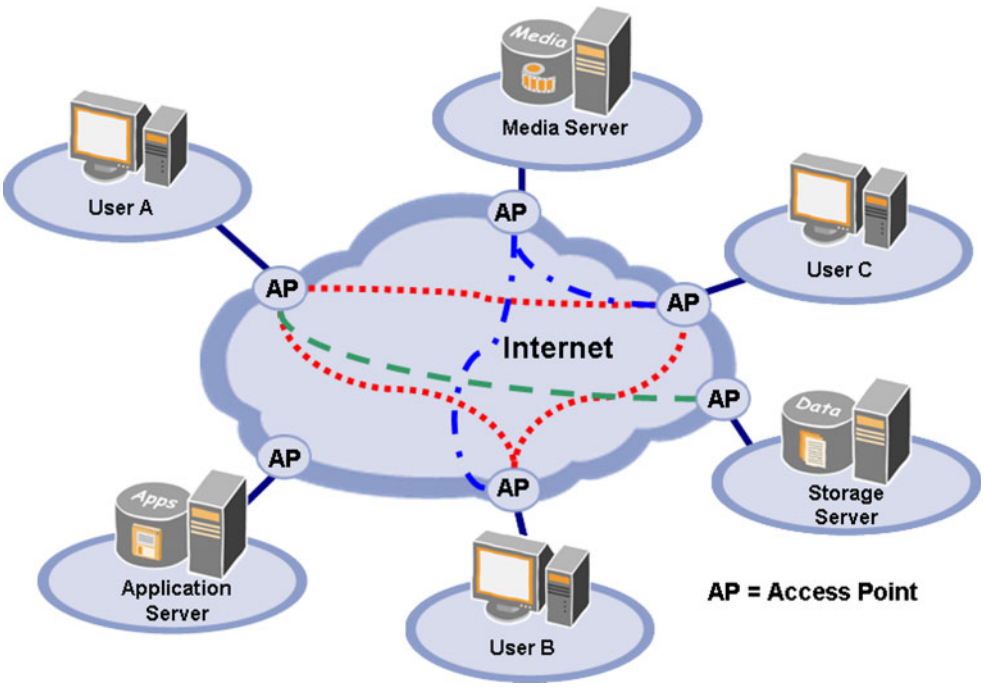

Fig. 1 Cloud scenario

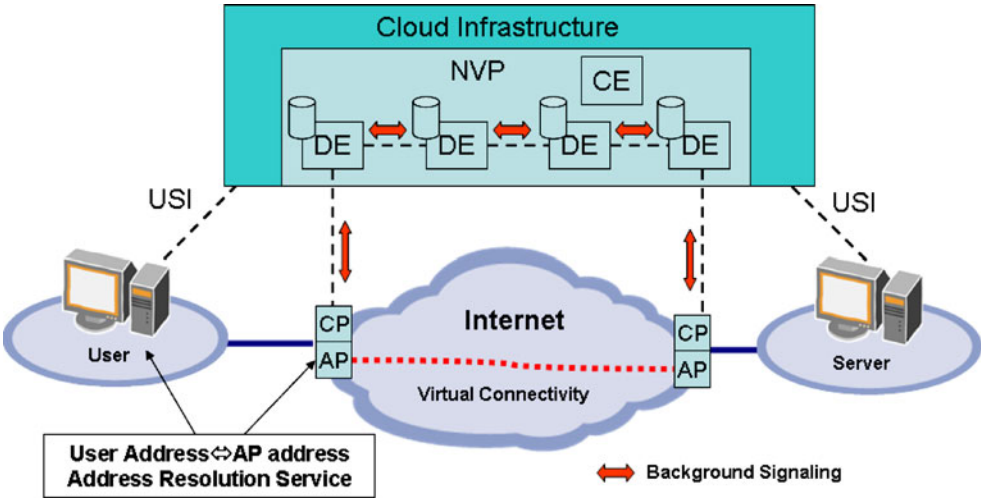

Fig. 2 NVP architecture

latency incurred during database lookup. Various kinds of resource discovery solutions have been developed, including the centralized and hierarchical information server approach (http://www.ist-phosphorus.eu). However, the aforementioned approaches have limitations as regards to scalability, fault tolerance, and network congestion.

Following the architecture proposed in [4], NVP is based on a decentralized resource discovery mechanism, named *background signaling*, for the automatic collection of the network topology information. In particular, the background signaling is based on messages exchanged at regular intervals between the DEs that directly control the CP of the network APs for recovering the network status information. Each DE distributes to the other DEs that information, in order to perform the network resource virtualization logic and thus the mapping between NaaS requests and relevant set of CP directives. Each DE stores the received resource attribute and status information in a local network resource database that is used also by the network address resolution service.

In addition to the background signaling, two additional signaling, independent of the CP signaling of the controlled network, named *service provisioning signaling* and *inter-domain signaling* are conceived among the NVP entities [4, 5]. The service provisioning signaling is responsible for the actual NaaS provisioning by performing the needed settings on the network nodes. Referring to Fig. 2, cloud users request NaaS by sending messages to the NVP via the user-to-service interface (USI). In the NVP prototype presented in [6], the USI signaling is implemented as XML-based messages exchanged over transmission control protocol (TCP) sockets. In particular, the XML technology allows to perform easily data processing in service mapping and structure checking in message validation. When a cloud user requests a NaaS to a DE (e.g., the creation of a virtual private network (VPN)) with given end-point addresses and perceived QoS parameters (e.g., the bandwidth, the delay, the jitter), the DE maps them in the parameters characterizing the CP connectivity as viewed by the transport network (e.g., the edge label switch router addresses, the effective bandwidth comprising the transport overhead) using the information stored in its network resource database and obtained from the background signaling. Then the DE distributes these parameters to the DEs controlling the edge nodes involved in the service provisioning (as determined during the mapping process). In turn, each DE uses these parameters to issue a set of directives to the controlled CP element at the boundary of the network. These directives may include buffer management, queuing and scheduling, packet filtering, traffic, classification, marking, policing, shaping, gate control, and firewall. Upon receiving connectivity directives, each CP element triggers a signaling session within the network CP for establishing the part of service under its competence.

The inter-domain signaling occurs among the NVP DEs deployed in different domains to provide edge-to-edge connectivity in a multi-domain context. In particular, each domain elects a DE as the entity that exchanges at regular intervals with its peers, located in different domains, a subset of the information stored in the network resource database relevant to the reachability of the access networks and of the inter-domain connectivity among edge nodes. The DE is enhanced with security and authentication capability according to the policy of the service providers. The exchanged information are used to fulfill inter-domain connectivity request to the CP that in turn computes the path and initiates the network reservation process according to the characteristic of the controlled networks and to the capability of the GMPLS external network-to-network interface that, to date, is still under standardization process.

3 NaaS characteristics and functionalities

The NVP is logically placed alongside (i.e., at the same level of abstraction of) the cloud virtualization platform, and it accepts on-demand NaaS requests from the user via a dedicated interface as shown in Fig. 3. The NVP implements NaaS by composing and coordinating the IP-based or sub-IP-based connectivity services provided by the actual network resources.

Following the proposed approach, NaaS may be invoked on demand by end users or may be combined with IaaS, PaaS, and SaaS to provide richer or advanced services, as it will extensively illustrated in the section dedicated to the use cases.

The resulting NaaS abstraction is a novel class of services for cloud computing that provides virtualized connectivity to end users at different levels of reliability, traffic QoS and transparency in a flexible and scalable way. Flexibility represents the possibility for the end user to dynamically change QoS parameters, like the bandwidth or the delay time, and to monitor the bill, the number of access, and their durations, according to its SLA. Scalability refers to the opportunity for the user to add or remove network virtual nodes easily without affecting its traffic.

The parameters that characterize a NaaS can be classified in two categories: the perceived QoS parameters and the user address parameters. The perceived QoS parameters refer to the connectivity attributes that an end user can perceive and monitor, for example, using software or hardware probes (e.g., the bandwidth, the packet delay, and the packet jitter). The user address parameters are relevant to the address space domain of the end user (e.g., the IP addresses of the user's hosts).

NaaS can be classified in the following service sets: network access service, virtual connectivity service, virtual topology service, virtual node service, and network cost estimation service as sketched in Fig. 4.

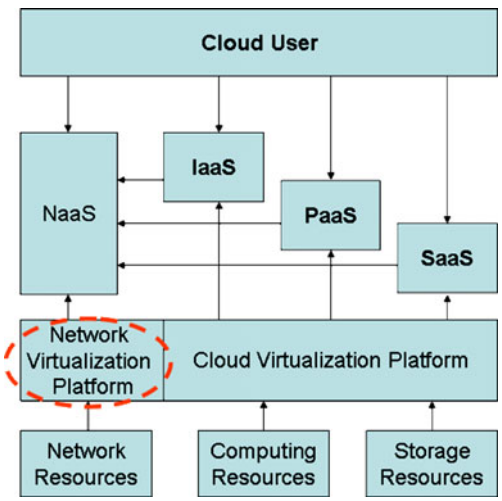

Fig. 3 Cloud architecture extension

The network access service is composed by the authentication service and the authorization service. The former regards the user identification following a service request in order to grant the access to the network service. The latter is related to service acknowledgement and provisioning according to user's SLA.

The virtual connectivity service offers the monitoring and management of the virtual connections established among users. It is composed by the connection creation service that allows a connection with the specific attributes to be created between a pair of user addresses, the connection deletion service that allows an existing connection to be deleted, the connection Monitoring service that provides the status of certain connection parameters, and the connection modification service which allows to modify parameters of an already established connection.

The virtual topology service is composed by the topology monitoring and the topology management, respectively for monitoring and managing virtual connectivity information such as the available bandwidth, then packets delay and jitter, and the restoration time.

The virtual node service is composed by the virtual node monitoring and the virtual node management, respectively for monitoring and managing information about virtual nodes. The information provided is strictly related to the type of virtual connectivity established (e.g., VPN, virtual private LAN service (VPLS)).

The NVP performs the virtual topology monitoring and the virtual node monitoring by interrogating, for example via SNMP, the connectivity databases of the CP at each border node. Since the NVP maps the virtual connectivity requests expressed by the cloud in a set of connectivity requests to the CP, the NVP combines its mapping information with the CP network status information, thus offering status information about virtual node and virtual topology.

The network cost estimation service (NCES) is a grid functionality standardized by the open grid forum [8] and implemented within the framework of the EU project DataGrid [9]. NCES provides information on the status and transmission behavior of the network to the grid services that, consequently, may improve their performance by dynamically adapting their behavior to the Grid network status.

The previously presented NaaS needs network-specific information to perform the mapping between a NaaS request and the set of CP-based directives needed to configure the network for the provisioning of that service. In particular, the parameters that characterize the request of a connectivity service provided by a NVP can be classified in three categories: the connectivity QoS parameters, the transport network address parameters, and the network technology

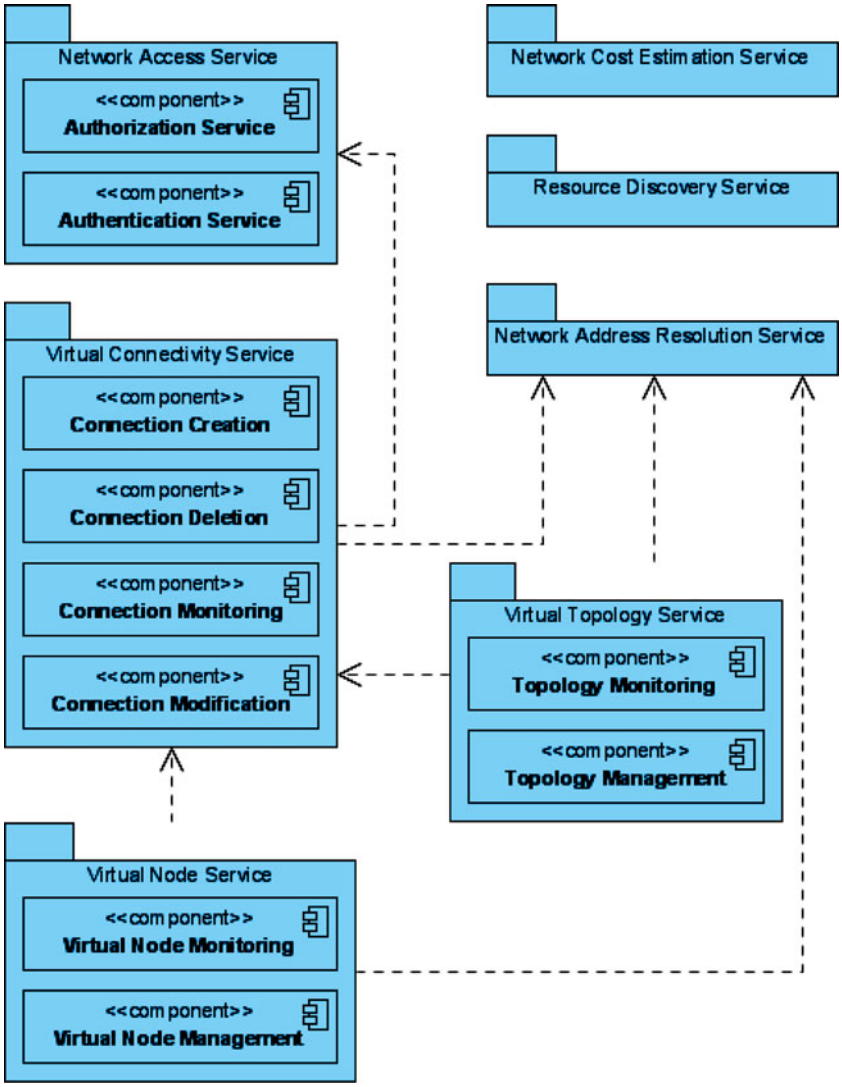

Fig. 4 NaaS classification

parameters. The connectivity QoS parameters consist in the QoS parameters that characterize the connectivity established in the transport network, such as the bandwidth, and the class of service used by a specific network technology (e.g., DiffServ). The transport network address parameters address the edge nodes of the transport network involved in the provisioning of a service, e.g., the network service address point of a transport network optical device. The network technology parameters are the technology-specific parameters needed by the NVP for setting up the network devices.

To perform the mapping from these parameters and the perceived QoS and user address parameters, two basic services should be conceived: the resource discovery service and the network address resolution service. Since these services deal with private network information, their use is only conceived to compose other NaaSs and cloud services in general.

The resource discovery service searches for resources that match the user's application requirements. The resources may be both network-specific and IT-specific. The former are relevant to the technology-specific information about the network resource controlled by the NVP (e.g., the available bandwidth, the available ports and channels) that should be utilized by the virtual connectivity service before the establishment of a new virtual connection. The latter are relevant to the information about the connectivity, reachability, and availability of the IT resources controlled by the cloud such as the list of storage servers connected to a specific sub-network or the IP addresses associated to a given cluster.

The network address resolution service bounds a user address to the relevant address of the network access node.

4 Use case 1: connectivity service provisioning

With reference to Fig. 1, we suppose that user A requests to the cloud infrastructure the provisioning of a connectivity

service among itself and users B and C. To this purpose, a *connectivity service request* message is delivered to the NVP by the cloud infrastructure using mechanisms that are out of the scope of this paper. The message contains the information about user A's identity (e.g., its IP address and the port used for receiving the response message) and the parameters relevant to the connectivity requested (e.g., the bandwidth, the list of IP address of the users to be interconnected).

A connectivity service request is the concatenation of the following basic NaaS: the authorization and authentication services, the address resolution service, and the connection creation service.

Referring to Fig. 5, the NVP first performs the authorization and authentication service. These services interrogate the SLA database for the authentication of the user A identity and for the authorization of the service provisioning.

The setup of the requested connectivity requires the knowledge of the APs involved in the connectivity provisioning. To this purpose, the NVP performs the address resolution service that interrogates the network resource database using

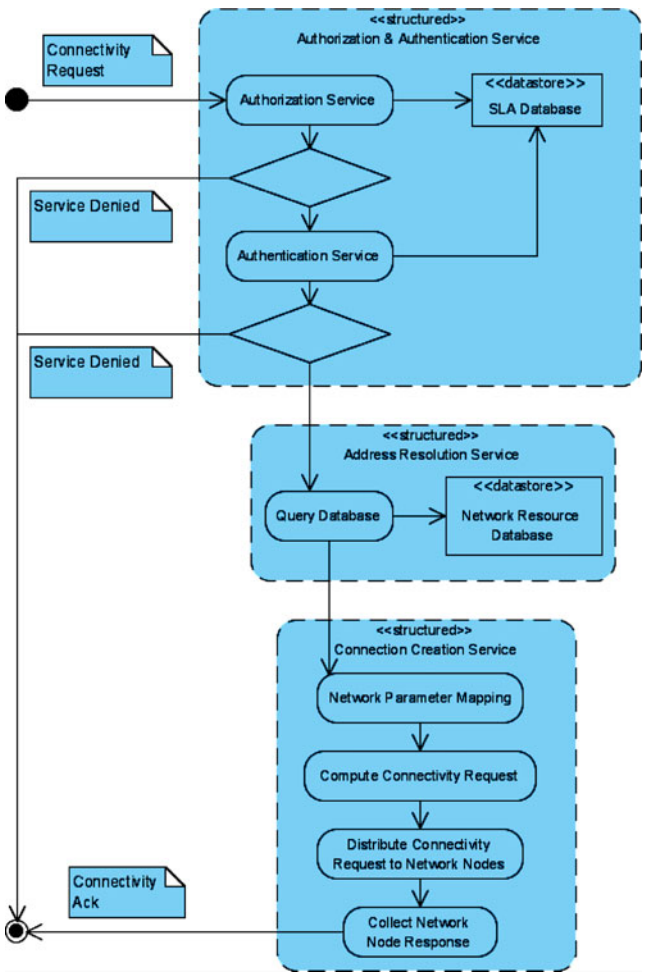

Fig. 5 Connectivity service provisioning flowchart

the IP addresses of users A, B, and C included in the connectivity service request message. Then, the NVP maps the perceived QoS parameters contained in the connectivity service request to the transport technology-dependent QoS parameters needed by the network for the provisioning of that service (e.g., the DiffServ class of service).

The NVP uses these parameters to compute a set of network directives. Typically, the operations needed for the provisioning of a connectivity service such as a VPN or a VPLS in commercial routers are the routing instance configuration, the border gateway protocol (BGP) configuration, and the multiprotocol label switching (MPLS) configuration. The routing instance configuration is needed to create an instance of VPN routing and forwarding (VRF) table, which is the private forwarding tables used to route IP packets within the proper VPN. The creation of the routing instance and the VRFs tables is needed to ensure that the VPN/VPLS remains distinct and separate within shared routers. The BGP is used to flood routes information among the APs in order to fill the proper forwarding tables. The MPLS configuration allows to establish and to configure the label switched path (LSP) that connect two APs.

The directives are distributed to the set of network elements involved in the service provisioning via the management interfaces typically available in commercial network equipments. After receiving and elaborating the directives, each network element sends a set of responses (generally one for each directive) to the NVP. The NVP collects all these responses and elaborates a connectivity acknowledge message containing the connectivity status (established or rejected) that is sent back to the cloud infrastructure.

5 Use case 2: Storage as a Service provisioning

Keeping as a reference the network scenario of Fig. 1, we suppose that user A request a Storage as a Service, i.e., a service that enables users to store data in the cloud, where the exchanged data travels over a reliable connection with assured bandwidth. Storage as a Service is an example of IaaS and, as shown in Fig. 6, may be subdivided in the following basic services: the authorization and authentication services, the address resolution service, the resource discovery service, and the storage service.

User A issues a service request to the cloud infrastructure providing its user identity and the parameters relevant to the service requested (e.g., the bandwidth, the level of reliability, the megabytes to be stored). In turn, the cloud infrastructure requests to the NVP to find a storage server and to establish a reliable connection with assured bandwidth between that server and the user A. The NVP performs the user authentication and the connectivity service authorization as seen in the

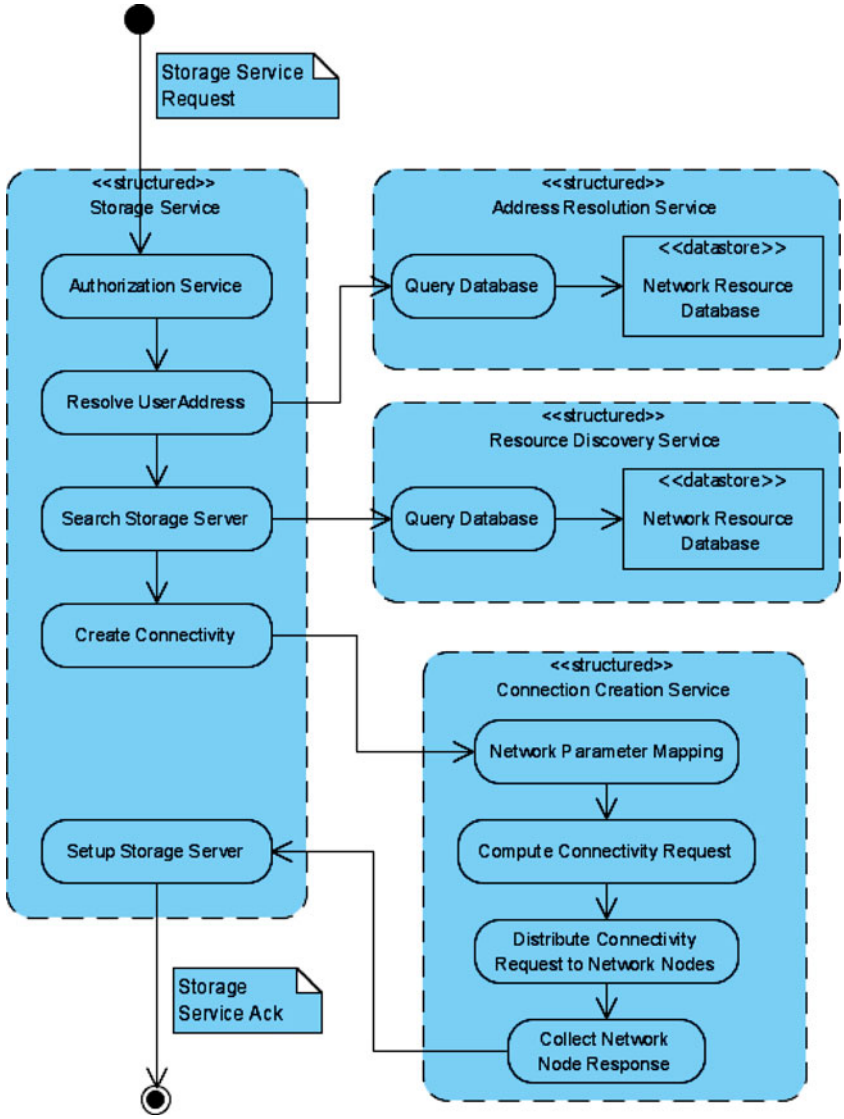

Fig. 6 Storage as a Service provisioning flowchart

use case 1. Still following the flowchart of Fig. 6, before establishing the requested connectivity, the NVP needs to know the network AP of the user A, its address and relevant interface. To this purpose, the NVP performs the network address resolution service that interrogates the network resource database using the user A IP addresses contained in the storage service request message. To find the most suitable storage server based on given policies (e.g., the nearest, the fastest, the less congested at the moment of the request), the NVP performs the resource discovery service. That service interrogates the network resource database, explained in Section 2, to retrieve the IP address and other useful information needed to address the server (e.g., the used TCP port). It is possible that no storage server can match the requirements thus the Storage as a Service may be provided by combining the storage capability of two or more servers.

Then, the NVP executes the connection creation service, explained in Section 3, to connect the user A with the found storage server, using for example an LSP, or storage services, by establishing for instance a VPN.

Finally, the NVP sends a message to the cloud infrastructure that in turn performs the setup storage service that configures the storage server/servers to accept the user A data.

6 Discussion

From the investigation of the presented use cases previously, some features of the proposed architecture can be outlined.

Network service virtualization Bandwidth-greedy cloud services or cloud user applications benefit from the delivery of QoS enabled virtual connectivity provided by the NVP where traffic flows are treated according to different requirements (e.g., the throughput, the maximum delay).

Dynamic service provisioning Cloud services and cloud user applications are able to directly issue network service requests on an end-to-end basis. The NVP performs the mapping between the end-node address and connection requirements in a set of network directives. In particular, this allows the cloud to take no interest in network-specific details and to focus on the application logic.

Reuse of transport control functionalities Cloud service providers benefit from the reuse of CP functionalities of the network domains. This allows developing new services or upgrading the existing ones for emerging bandwidth-greedy applications by preserving existing network infrastructure. In fact, thanks to the separation of transport and service functionalities, the proposed cloud architecture enables the deployment of network services without application-specific control functionalities and decouples the network technologies from evolving service requirements.

Smooth migration The migration from the current cloud computing architecture towards the proposed architecture can be smooth because the NVP may be first implemented in separate devices connected to the cloud devices, and at a later stage implemented as a software extension of the cloud computing equipments.

Scalability The NVP configures the DEs in parallel, i.e., approximately at the same time, thus the number of DEs does not affect the NaaS provisioning time. From the experimental result presented in [4, 5], we can affirm that the service provisioning signaling time is principally determined by the time needed by the DEs for the execution of the CP directives. This consideration allows to conclude that in a single routing area, the architecture is not suffering a structural increase of the service setup time when the number of node is increased. Regarding the number of message and the amount of the information exchanged among DEs, in the background signaling, they grow as n^2 where n is the number of edge nodes of the network, while in the service signaling, they linearly grow with the number of edge nodes involved in the connectivity provisioning independently from the network(s) dimension.

7 Conclusion

We have discussed the abstraction of NaaS, a novel class of cloud computing services that enables cloud users to request on-demand connectivity without any knowledge of the complexity and the technology details of the network. NaaS is enabled by a network virtualization platform (NVP). The NVP has principally two roles: it collects network status at the boundary of the network and maintains a distributed network resource database, and it performs the mapping between the perceived QoS parameters of the NaaS to the transport technology-dependent QoS parameters needed by the network for the provisioning of that NaaS. The mechanism that allows the NVP to collect network topology and technology information has been presented. It is based on a signaling among the NVP DEs and allows to achieve scalability in terms of network dimension and service delivering latency. An NVP can control more than one network domain. In particular, an NVP implementation is provided with functionalities in order to communicate with other NVP for the provisioning of inter-domain NaaS.

We have also provided a classification and a characterization of the principal NaaS and described the relevant functionalities. From the perspective of service providers, new cloud services are possible without significant modification on both the cloud computing and the network infrastructure, as illustrated by the two proposed use cases.

Acknowledgment The work described in this paper was carried out with the support of the Building the Future Optical Network in Europe (BONE) project, a network of excellence funded by the European commission through the 7th ICT-Framework Programme.

References

1. Foster I, Yong Zhao, Raicu I, Lu S (2008) Cloud computing and grid computing 360-degree compared. In: Grid computing environments workshop, 2008. GCE '08, Austin Convention Center, Austin, 12–16 Nov. 2008
2. Audouin et al (2009) CARRIOCAS project: an experimental high bit rate optical network for computing-intensive scientific and industrial applications. TridentCom 2009, Washington, DC, 6–8 April 2009
3. Lehman T, Sobieski J, Jabbari B (2006) DRAGON: a framework for service provisioning in heterogeneous grid networks. Communications Magazine IEEE 44(3):84–90
4. Martini B, Martini V, Baroncelli F, Torkman K, Castoldi P (2009) Application-driven control of resources in multiservice optical networks. IEEE/OSA J Opt Commun Networking 1: A270–A283
5. Baroncelli F, Martini B, Martini V, Castoldi P. A cooperative approach for the automatic configuration of MPLS-based VPNs. International Journal of Grid Computing and Multi Agent Systems (GCMAS) (in press)
6. Baroncelli F, Martini B, Martini V, Castoldi P (2007) A distributed signalling for the provisioning of on-demand VPN services in transport networks, Integrated Network Management (IM) 2007. Germany, Munich
7. Chowdhury NMMK, Boutaba R (2009) Network virtualization: state of the art and research challenges. Communications Magazine IEEE 47(7):20–26
8. Tiziana Ferrari (2007). Grid network services use cases from the e-Science Community. http://www.ggf.org/documents/GFD.122.pdf. Accessed 12 Dec 2007
9. Final report on network infrastructure and services, DataGrid deliverable 7–4, Jan 2004 (https://edms.cern.ch/document/414132). Accessed 4 Feb 2004