Autonomic Brokerage Service for an End-to-End Cloud Networking Service Level Agreement

Mohamad Hamze, Nader Mbarek, Olivier Togni
Le2i Laboratory, UMR 6306 CNRS, University of Burgundy
Dijon, France
E-mail: {Mohamad.Hamze, Nader.Mbarek, Olivier.Togni}@u-bourgogne.fr

***Abstract*—Today, cloud networking which is the ability to connect the user with his cloud services and to interconnect these services within an inter-cloud approach, is one of the recent research areas in the cloud computing research communities. The main drawback of cloud networking consists in the lack of Quality of Service (QoS) assurance and management in conformance with a corresponding Service Level Agreement (SLA). In this paper, we propose a framework for self-establishing an end-to-end service level agreement between a Cloud Service User (CSU) and multiple Cloud Service Providers (CSPs) in a cloud networking environment using brokerage service. We focus on QoS parameters for NaaS and IaaS services. This framework ensures a self-establishing of the proposed SLA using autonomic cloud managers. Moreover, videoconferencing is a very demanding application in terms of bandwidth, delay and CPU resources that we have to guarantee. Therefore, we propose to use our cloud networking architecture and framework to provide the cloud videoconferencing application with QoS guarantee. We simulate the corresponding scenario using the CloudSim toolkit to evaluate its performance. The results reveal a high scalability, and a good video conferencing delay.**

Keywords-Cloud Networking; Autonomic Computing; Service Level Agreement; Quality of Service; Videoconferencing.

I. INTRODUCTION

Cloud computing is a promising technology for the realization of large, scalable on-demand computing infrastructures. Many enterprises are adopting this technology to achieve high performance and scalability for their applications while maintaining low cost. Furthermore, an important differentiation element between cloud environments is the Quality of Service and Service Level Agreement.

In general, a Cloud Service User (CSU) requires for his services an end-to-end Quality of Service (QoS) assurance, a high level reliability and a continued availability. However, there are limitations when users and/or applications rely on guaranteed QoS between deployed cloud infrastructures that are spread across geographically distributed sites or domains.

Therefore, inter-cloud systems are unavoidable as it is very difficult for a single Cloud Service Provider (CSP) to satisfy his customer requirements. In addition, if a cloud is out of resources, it could not meet the new demands of its users. Thus, geographically distributed data centers offer better end-to-end performance between CSU and CSP and improve reliability when failure occurs.

On the other hand, the inter-cloud should be designed as a multi-vendor environment with the ability to migrate services from one provider to another and to locate the best resources not only in terms of computing power and storage, but also connectivity, bandwidth and delay. Therefore, the networking aspect of cloud computing is a critical factor for adopting this approach. In this context, cloud networking is defined as the ability to connect the user to his cloud services and interconnect services within an inter-cloud. It is one of the recent research areas in the cloud computing research communities. It is built upon two main concepts. The first concept is the integration of the networking resources onto existing data center. The second one is the deployment of distributed computing and storage resources in the network to allow better end-user experience and lower dependency on network capacity [1].

However, to achieve the cloud networking model, there are a number of open issues, such as the formality of a language for SLA between CSUs and CSPs, the interoperability of data formats, and communication interoperability using standard interfaces. In addition, there are major challenges on how to establish an SLA in cloud networking and how to choose suitable CSPs during the execution based on SLA and QoS requirements. Furthermore, the development of an autonomic cloud control aims to simplify the complexity and maximize efficiency while minimizing cost and user interactions.

Moreover, recent advances in broadband communication and computing technology have accelerated the proliferation of multimedia conferencing services in large-scale enterprises, and a high quality multimedia conferencing is considered as an urgent requirement. However, the corresponding products cannot support a large number of participants [2]. Meanwhile, cloud computing becomes the most fashionable technology and developing trend in IT industry now. In addition, CSUs and the companies in particular, accepted leaving private IT assets inside and migrating less sensitive data and operations to the public cloud. Therefore, in order to allow a cost-effective scaling and best performance of videoconference systems, the use of cloud computing resources appears as a natural approach, since they provide computing resources available on demand and the ability to pay for use of computing resources [3]. However, the use of cloud computing resources from a single provider has several disadvantages as shown in [3, 4]. Moreover, video streaming in videoconferencing systems is a

978-0-7695-5168-5/14 $31.00 © 2014 IEEE
DOI 10.1109/NCCA.2014.17

IEEE computer society

very demanding application concerning bandwidth resources. Thus, the use of network and computing resources from different providers according to a self-established SLA can help us providing a service with better performance, lower cost and avoiding or at least mitigating most of the problems of cloud computing.

In this paper, we propose a framework for self-establishing an end-to-end service level agreement for cloud networking. We propose at first a cloud networking architecture in an inter-cloud broker scenario. We describe the calculation and splitting of SLA parameters and the interaction between different entities to establish the SLA. Then, we present the SLA self-establishing framework for our proposed cloud networking architecture using autonomic cloud managers. In addition, we enable a videoconferencing application to take full advantage of our cloud networking architecture.

The remainder of this paper is organized as follows: in Section II, we present a brief overview on the state of the art for cloud computing and networking. Section III highlights the most relevant research works and trends in this area. In Section IV, we describe the proposed architecture for establishing an end-to-end SLA in cloud networking. In Section V, we describe our self-establishing framework. Section VI presents the videoconferencing usage case and its evaluation. Lastly, Section VII concludes the paper and points out future work.

II. State of The Art

A. Cloud Computing

In 2011, the NIST [5] defined cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This model consists of five characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. It contains four deployment models: private cloud, community cloud, public cloud, and hybrid cloud. It consists of three service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

In addition, most IaaS cloud providers offer a set of general-purpose VM classes with generic resource configurations. For example Amazon EC2 supports 11 types of VMs, each one with different options of processors, memory, and I/O performance.

B. Cloud Networking

The term cloud networking is introduced in a multi-administrative domain scenario, where network and data center domains must interact through defined interfaces to provide a service to CSU. Cloud networking allows specifying a service through high-level objectives or goals, rapid deployment and management of cloud services across data centers and operator networks, and autonomous optimization and management of underlying resources to provide and maintain requested QoS [1]. Cloud networking extends network virtualization beyond the data center to bring two new aspects to cloud computing: (a) connecting the user to services in the cloud, and (b) interconnecting services that are geographically distributed across cloud infrastructures. Cloud networking users would be able to specify their needed virtual infrastructure and the desired networking properties to access these resources [6].

There are three kinds of networks in cloud environment [7]:

1) Intra-cloud network: this network connects local cloud infrastructures and comprises three kinds of networks: (a) between VMs at the same server, (b) between servers and storage systems, and (c) between VMs at different servers. The data centers require different traffic models to guarantee latency, packet loss and bandwidth requirements.

2) Core transport network (WAN/MAN): this network can be used by customers to access and consume cloud services deployed within the cloud provider's data center.

3) Inter-cloud network: it can be used to interconnect cloud infrastructures together. These cloud infrastructures may be owned by the same cloud provider or by different ones.

In addition, cloud networking services that provide network connectivity in the three networks above are named Network as a Service (NaaS). NaaS allows tenants to customize the services they receive from the network and the provider to decide how resources are allocated and shared between tenants. NaaS requirements include QoS, security and availability.

Indeed, the performance of data transfer in core transport and inter-cloud networks is difficult to guarantee. The most convenient solution is to use enough bandwidth to carry information in these networks. This bandwidth is generally provided by any entity that has the ability to provide network connectivity and reservation, e.g. Network Service Provider (NSP). Furthermore, the NaaS service offered by these NSPs is the Bandwidth on Demand (BoD). BoD is the ability to perform on-demand changes (increase, decrease) and instant provisioning of bandwidth on the particular links via standardized interfaces. The BoD supports the establishment of an end-to-end SLA between CSU and CSP [7]. It provides an economical solution for QoS and interoperability of Clouds.

C. Inter-Cloud Service Broker

Inter-cloud computing allows on-demand assignment of cloud resources, including computing, storage and network, and the transfer of workload through interworking of cloud systems [7]. The communication between different clouds is established through the cloud networking that can provide a NaaS service (BoD). CSPs can interwork thanks to Inter-Cloud Service Broker (ISB) that provides indirect interconnection between two or more CSPs and brokering services to CSUs or CSPs. The ISB provides and executes services of three categories [7]: (a) Service intermediation: the ISB enhances a given service by improving some specific capability and providing value-added services to CSUs. (b) Service aggregation: ISB combines and integrates multiple services into one or more new services. (c) Service arbitrage: it is similar to service aggregation except that the services being aggregated are not fixed. In this case, the ISB has the flexibility to choose services from multiple sources.

III. Related Work

The research project SAIL (Scalable & Adaptive Internet Solutions) [8] describes a cloud networking architecture and focuses on security, but it does not consider the QoS and SLA. The research project Foundation of Self-governing ICT Infrastructures (FoSII) [9] is proposing solutions for autonomic management of SLAs in the cloud. It is investigating self-governing cloud computing infrastructures necessary for the attainment of established Service Level Agreements (SLAs). In addition, the research project Contrail [10] aims to vertically integrate an open-source distributed operating system for autonomous resource management in IaaS environments and PaaS systems. It monitors SLAs at all levels of the federation cloud. However, these two projects do not consider network QoS parameters and NaaS services. We consider our work to be very much in alignment with the objectives of these projects. But, our research work is innovative by considering the self-establishing of SLA in cloud networking environment, and by focusing on QoS guarantee for IaaS and NaaS services.

From standardization perspective, IEEE Cloud Computing formed the Inter-Cloud Working Group (ICWG). It announced the launch of two new standards development projects in April 2011: P2301 [11], a guide for Cloud Portability and Interoperability Profiles (CPIP) and P2302 [12], a Standard for Intercloud Interoperability and Federation (SIIF). Open Grid Forum (OGF) is active in the definition of the Open Cloud Computing Interface (OCCI) [13] for the interoperability between clouds. GICTF (Global Inter-Cloud Technology Forum) [14] studies the standard Inter-Cloud interfaces to improve the reliability of the Clouds, and presents SLA metrics, including security attributes for Inter-Cloud environment. IBM recently presented CloudNaaS in [15], a cloud networking platform for enterprise applications. In our research work we propose to develop a cloud networking framework and we aim to enable communication between, not only CSPs, but also broker, NSPs and CSPs. In addition, this communication is self-established using autonomic cloud managers and through agreed-upon interfaces. For that purpose, we use web services standard technologies.

Finally, there are many related research works on QoS, e.g. [16, 17], but QoS mentioned in these works is for SaaS, PaaS or IaaS. In addition, several research works present SLA for cloud computing only: the project Mycloud [18] proposes Cloud Service Level Agreement (CSLA) and the authors in [19] propose to use Web Service Level agreement (WSLA) [20] in a cloud computing context. Moreover, several recent research works on cloud videoconferencing [2, 21, 22] consider only QoS parameters without establishing the corresponding SLA. Therefore, our paper considers, in addition, NaaS services, QoS attributes and an SLA for autonomic cloud networking environment.

IV. Proposed Cloud Networking Architecture

In order to deliver the service level expected by CSU, the proper design, construction and management of a cloud networking architecture is very critical. Therefore, we propose to design an inter-cloud broker based architecture (Fig. 1). We assume an environment with multiple CSPs connected through several NSPs. The CSP can offer resources from one or several datacenters (DC). Each entity (broker, CSP, NSP) has a repository that contains information about resources and corresponding service levels. Within this cloud networking architecture, we must ensure consistency between the QoS requirements requested by the CSUs, and SLA proposed by CSPs to allow multiple CSPs working together to meet the CSU requirements. This architecture will be validated for a videoconference service as described in section VI.

A. Cloud Networking Architecture

In the proposed cloud networking architecture (Fig. 1), the cloud broker is emerged as an intermediate entity between a CSU and CSPs to help the establishing of a service level that meets the CSU requirements and to simplify secure integration and lower prices for CSU. It can help to move data between different CSPs. Thus, we propose the NSP as part of our architecture, providing BoD network service (e.g. network operator) and playing the role of a NaaS CSP. Therefore, CSPs do not need to have a visibility on this network. They will see a service provided by NSP. In addition, the CSP provides IaaS and NaaS services. The IaaS service concerns VMs and storage resources and the NaaS service concerns network datacenter resources that connect local IaaS resources.

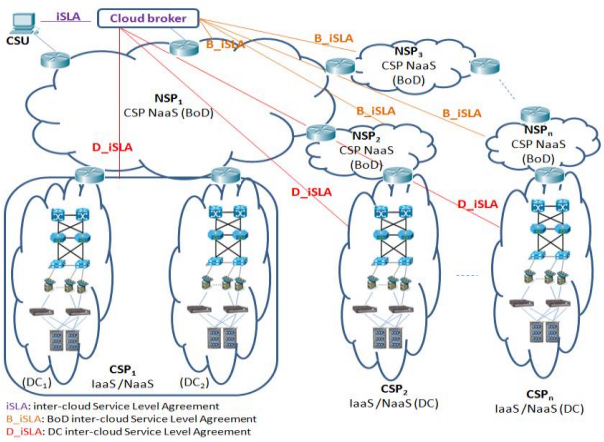

Fig. 1. Cloud Networking Architecture

B. SLA description

In cloud environment, a Service Level Agreement (SLA) is a contract between CSU and CSP. It contains details of shared information and service level guarantees that are offered by CSP and required by CSU. An SLA contains both technical and non-technical parameters. The technical parameters constitute the negotiable part of the SLA, and are grouped together in a specification called Service Level Specification (SLS). The SLS is a set of parameters such as QoS parameters and their corresponding values which allow the definition of the service level offered to CSU. The SLA can be structured by: SLA Identifier, Validity Period, Obliged Parties, Service Identification, Guarantees offered on the Services, and Business Attributes (price, violation and penalties).

To the best of our knowledge, several works have been done on the SLA establishing in cloud computing, but not in cloud networking. In addition, most of the current CSPs are limited only to the availability of resources and do not take into account many other important parameters such as latency and bandwidth [23]. In our cloud networking broker architecture we propose three types of SLA:

1) inter-cloud Service Level Agreement (iSLA): It is a contract between a CSU and a cloud broker with QoS parameters for NaaS and IaaS services. These QoS parameters could be quantitative or qualitative. The most important quantitative QoS parameters of an iSLA are shown in Fig. 2.

2) BoD inter-cloud Service Level Agreement (B_iSLA): It is a contract between a cloud broker and NSPs connecting CSU to CSP. It is for NaaS (BoD) services. QoS parameters are the same as iSLA NaaS QoS parameters.

3) Datacenter inter-cloud Service Level Agreement (D_iSLA): It is a contract between a cloud broker and a CSP for NaaS and IaaS services. QoS parameters are the same as specified within the iSLA Performance Guarantees attribute.

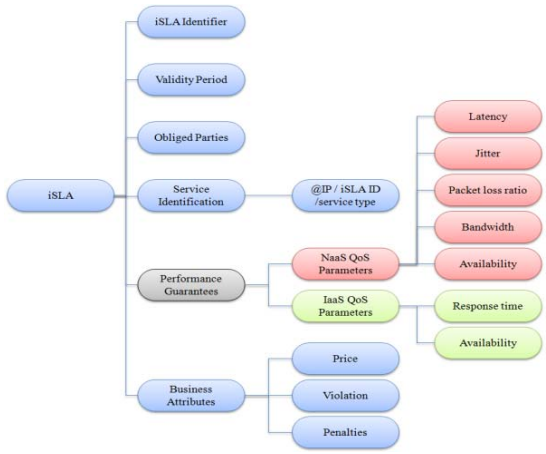

Fig. 2. Inter-cloud Service Level Agreement (iSLA) Structure

C. Interaction between entities

The interaction between CSU, Cloud Broker, CSPs and NSPs to achieve an end-to-end SLA is described in Fig. 3. At first, CSPs and NSPs describe IaaS and NaaS services for their available resources with different service levels and send this information to the cloud broker. In addition, if any changes occur within CSPs and NSPs, they send these changes to the cloud broker which updates its repository. Next, the CSU sends service requirements to the cloud broker to construct an iSLA specifying a service level that contains QoS parameters for NaaS and IaaS services. The cloud broker consults its repository and compares the CSU requirements with the different services and the corresponding service levels offered by CSPs and NSPs to select the appropriate CSPs and NSPs that meet the CSU QoS requirement. Afterward, the cloud broker sends a request to establish a D_iSLA (IaaS, NaaS) and a B_iSLA (NaaS BoD) with respectively selected CSPs and NSPs. After this, the concerned CSPs reserve and configure VM and network resources to deliver IaaS and NaaS services, and the concerned NSPs reserve and configure network resources to deliver NaaS (BoD) service. Then, they update their repository with these changes and send confirmation to cloud broker. Finally, the cloud broker updates its repository with these changes and establishes the iSLA with the CSU.

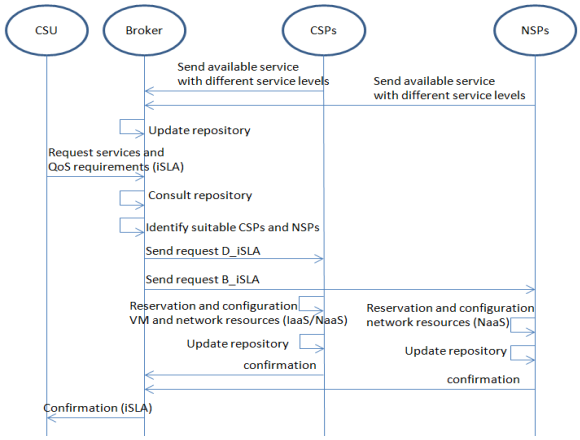

Fig. 3. MSC for Inter-cloud Broker Scenario

D. CSP and NSP Selection Algorithm and QoS Splitting

The selection by the cloud broker of the best CSPs and NSPs according to QoS splitting for a given iSLA is a multi-objective optimization problem (QoS offered, total price, reputation…). Thus, we can split the iSLA and select CSPs and NSPs for a particular CSU request based on proposed Algorithm 1 and equations (1) to (7).

Let $C = \{c_1,...,c_n\}$ represent the set of CSPs, n (number of CSPs offering a service to CSU with different service levels), $N_j = \{N_{1j},...,N_{mj}\}$ represent the set of NSP networks between CSU and c_j, ($\forall j,\ 1 \leq j \leq n$), m (number of these NSPs), S represent the user service requirements {VMs Number and Characteristics based on desired Response time}, U represent the user QoS parameters {L (Latency), J (Jitter), P (Packet loss ratio), B (Bandwidth), A (Availability), and R (Response time) / U_1 is from CSU_1 to c_j and U_2 is from c_j to CSU_2, and if a CSU sends a workload for itself, $U_2=\varnothing$ }, T represent selected CSPs and NSPs, Q and V represent temporary sets.

Algorithm 1 CSP / NSP Selection and VM allocation

```
Q ← ∅, T ← ∅, V ← ∅
for each c_j ∈ C do
  if equations (1) to (7) are met then
    Q.Enqueue(c_j, N_j)
  end if
end for
Q.Sort_descending_order_based_QoS_parameters(U)
while S ≠ ∅ ∧ Q ≠ ∅ do
  {c_i, N_i} ← Q.Dequeue()   //i.e. select best CSP and NSP_s
  V ← (allocate required VMs available in c_i)
  S ← S − V                  //i.e. calculate remaining VMs
  V ← ∅
```

13: *T. Enqueue(c_i, N_i)*
14: **end while**
15: **if** $S \neq \varnothing$ **then**
16: reject user request
17: **for** each $c_i \in T$ **do**
18: release allocated VMs
19: **end for**
20: return $\varnothing$
21: **else**
22: accept user request and establish iSLA, D_iSLA and
23: B_iSLA
24: return T
25: **end if**

1) NaaS QoS parameters splitting equations:

$$L(iSLA) \geq \sum_{i=1\ldots m} L_1(N_{ij}) + L_1(c_j) + L_2(c_j) + \sum_{i=1\ldots m} L_2(N_{ij}). \quad (1)$$

$$J(iSLA) \geq \sum_{i=1\ldots m} J_1(N_{ij}) + J_1(c_j) + J_2(c_j) + \sum_{i=1\ldots m} J_2(N_{ij}). \quad (2)$$

$$P(iSLA) \geq 1 - \prod_{i=1\ldots m} \left(1 - P_1(N_{ij})\right) \times \left(1 - P_1(c_j)\right) \times \left(1 - P_2(c_j)\right) \times \prod_{i=1\ldots m} \left(1 - P_2(N_{ij})\right). \quad (3)$$

$$B(iSLA) \leq \min\left(\min_{i=1\ldots m} B_1(N_{ij}),\ B_1(c_j),\ B_2(c_j),\ \min_{i=1\ldots m} B_2(N_{ij})\right). \quad (4)$$

$$A_{NaaS}(iSLA) \leq \min\left(\min_{i=1\ldots m} A_1(N_{ij}),\ A_1(c_j),\ A_2(c_j),\ \min_{i=1\ldots m} A_2(N_{ij})\right). \quad (5)$$

2) IaaS QoS parameters splitting equations:

$$R(iSLA) \geq R(c_j). \quad (6)$$

$$A_{IaaS}(iSLA) \leq A(c_j). \quad (7)$$

V. Autonomic Cloud Networking Framework

The availability of a reliable system for self-establishing the SLA reduces complexity and minimizes the user interaction. In this section, we propose to establish the iSLA, B_iSLA and D_iSLA corresponding to our cloud networking architecture autonomously using Autonomic cloud Managers (AMs).

A. Autonomic Cloud Networking Architecture

In general, a domain refers to resource collections involving hardware and software managed by a single entity, e.g. cloud data center, or communication network. If we manage this domain in an autonomic manner, we call it Autonomic cloud Domain (AD). In this context, we present the proposed cloud networking architecture (Fig. 4) with several ADs (cloud broker, CSPs and NSPs). Each Autonomic cloud Domain, is under the authority of an inter-cloud Autonomic Manager (iAM). iAM communicates with other iAMs to achieve an agreement on a service level. In addition, it controls one or more AMs to configure resources in conformance with an agreed on service level.

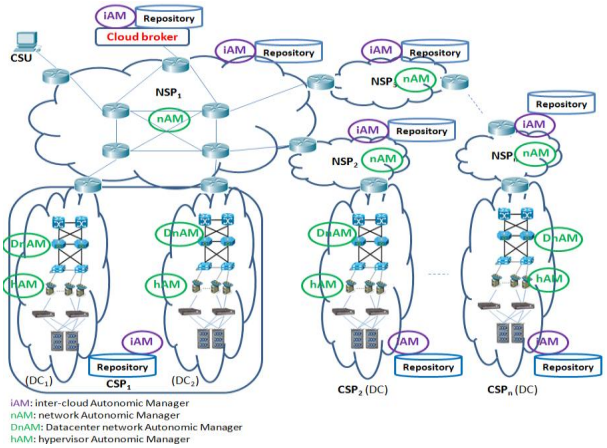

Fig. 4. Autonomic Cloud Networking Architecture

These Autonomic cloud Managers are playing different roles within our autonomic cloud networking architecture:

1) network Autonomic Manager (nAM): it is responsible for creating and managing NSP virtual networks connecting CSU to CSP or connecting CSPs in conformance with the agreed B_iSLA, and monitoring workload and performance.

2) Datacenter network Autonomic Manager (DnAM): it is responsible for creating and managing virtual networks for the CSU within the cloud data center in conformance with the NaaS part of the D_iSLA, and regularly monitoring workload and performance.

3) hypervisor Autonomic Manager (hAM): it is responsible for creating, monitoring and controlling virtual machines (VMs) and storage space. Therefore, the CSP can consequently decide the allocation or deallocation of VMs to maintain the performance as close as possible to performance objective in conformance with the IaaS part of the D_iSLA.

We provide these autonomic cloud managers with the capability to achieve an agreement between the Autonomic cloud Domains. This agreement covers QoS aspects for different service models such as IaaS and NaaS with different service levels. In addition, each entity must be proactive in notifying the consumer when the terms of the SLA are breached for critical data and applications. Moreover, the iAM entities use a repository to store resource management information and to facilitate their interactions with other AMs.

B. Autonomic Cloud Manager Description

The Autonomic cloud Manager (Fig. 5) (iAM, nAM, DnAM, or hAM) manages a single resource or set of resources (AM, VMs, network resources…) controlled by sensors and effectors. The sensors provide mechanisms to collect information concerning the state of a managed cloud element.

Effectors are mechanisms that change the state of that element. After the human administrator integrates its policies in the knowledge base, the autonomic cloud manager begins with the monitoring phase to ensure data collection, aggregation, filtering and reporting from managed resources thanks to the sensor interfaces. Then, the collected data is passed to the analysis phase to correlate these data in accordance with the knowledge base. Then a request for a change is sent to the planning phase to indicate actions needed to achieve specific objectives in accordance with specified policies. Finally, these actions are sent to the execution phase that allows changes to be made in the managed cloud resource by the effectors. In addition, the changes are checked to update the knowledge base by the monitoring phase. Our Autonomic cloud Manager is in conformance with the IBM architecture [24].

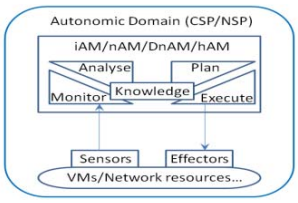

Fig. 5. Functional Details of the Autonomic Cloud Manager

C. Autonomic Cloud Manager Lifecycle

We present the lifecycle of the iAM autonomic cloud managers proposed within our SLA self-establishing framework thanks to the following FSM (Finite State Machine) (Fig. 6).

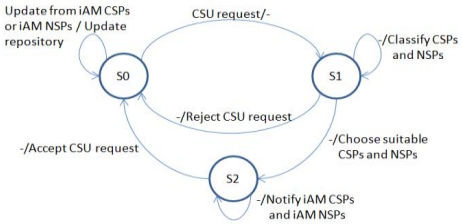

Fig. 6. FSM for iAM Broker Lifecycle

The FSM includes three states. In the first state S0, the iAM broker receives available services with different service levels or any changes from the CSPs and NSPs in the alliance. Then, it updates its repository. After receiving CSU service requirements to construct an iSLA, the iAM broker goes to the second state S1. In this state, the iAM broker classifies the appropriate CSPs and NSPs according to Algorithm 1. If the iAM broker does not find any CSPs that meet the CSU requirement, it rejects the CSU request and goes to the initial state S0. Else, the iAM broker chooses the suitable CSPs and NSPs, and goes to the state S2. In this state, the iAM broker establishes a D_iSLA and B_iSLA with respectively iAM CSPs and iAM NSPs. Finally, the iAM broker accepts the CSU request, establishes the iSLA, and goes to the initial state S0.

D. Autonomic Cloud Manager Interaction

To provide our cloud networking architecture with autonomy while offering an end-to-end QoS guarantee, two kinds of interactions could take place between autonomic cloud managers (Fig. 7). Thanks to the first one, an iAM initiates a peer to peer communication process with the corresponding iAMs in a horizontal interaction to achieve an agreement on a service level. This communication between iAMs is based on Web services (WS) technologies.

In addition, each iAM is responsible of the service level guarantee within the corresponding Autonomic cloud Domain. This guarantee will be possible thanks to a second kind of interaction. Indeed, the iAM controls one or more Autonomic cloud Managers (nAM, DnAM, or hAM) thanks to a standardized manageability interface (effectors and sensors) using Web Services technologies to achieve this service level guarantee. These AMs are responsible of service level guarantee within their respective local ADs thanks to the same kind of interactions. Therefore, iAM provides them with the corresponding service level (B_iSLA or D_iSLA) in a vertical interaction, so that they use a similar interaction to allocate, release, or modify the configuration of their managed resources (MRs: router, switch, server, VM, database...) according to the received service level parameters values.

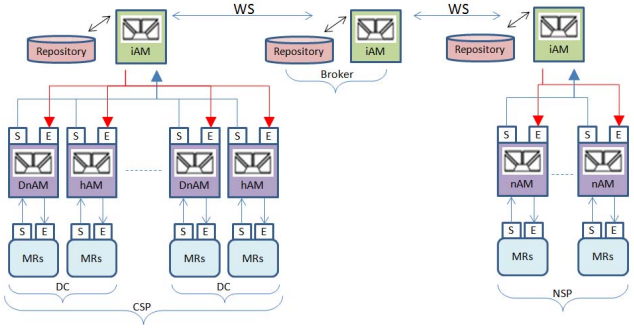

Fig. 7. Autonomic Cloud Manager Interaction Framework

The manageability sensor interface (S) enables iAM, nAM, DnAM, and hAM to request or receive a monitoring information notification from the entities under their control. The manageability effector interface (E) gives these autonomic cloud managers the means to perform actions for behavior modification of managed resources. These manageability interfaces enables vertical interactions between an iAM and their AMs but also a nAM, DnAM or hAM and the corresponding MRs like VMs and network resources.

VI. Usage Case and Validation

To take full advantage of the outstanding performance of our proposed cloud networking framework, we present a usage case for the efficient establishing and management of large-scale videoconferencing service. We test the corresponding scenario as a proof of concept and evaluate its performance by conducting a set of performance evaluation studies using the CloudSim toolkit [25].

A. Usage Case

A videoconference system is a real-time exchange of media data among several parties, which breaks the limit of human communications due to the geographical location of

participants. It allows a great number of users per conference, multiple simultaneous conferences, different client software, and provides an automatic recording service. Typical videoconferencing scenario includes several videoconference clients that are connected through a Multipoint Control Unit (MCU), Flash or SIP. In all cases data flow transcoding is necessary to render all videos and encode them with the corresponding codec. We need also a Real Time Messaging Protocol (RTMP) server for the flash clients and a SIP server for the SIP clients. In addition, server partition is widely used in videoconferencing systems, where we have the video encoding, transcoding and recording servers (CPU intensive), MCU and RTMP streaming servers (bandwidth intensive), and recorded conferences in a NFS server (storage intensive).

From the perspective of both bandwidth requirement and end-to-end delay constraints, multi-party videoconferencing may be one of the most demanding multimedia applications [22]. Therefore, we use NaaS BoD service to take full advantage of inter-datacenter networks in our cloud networking architecture. In addition, we use IaaS services to provide high VM performance for video conferences, while maintaining acceptable delays. Moreover, to be more scalable, we believe that some video flows from different participants in their respective conferences can be aggregated.

The videoconferencing scenario considered in this paper includes several SIP based participating clients. Interaction between clients using SIP is in conformance with the RFC 3261 described in [26]. When the participant number increases, the conference room enlarges its size by getting additional resources from the corresponding resource pool within the cloud infrastructure thanks to the brokerage service. When participants leave the videoconference, the resource would be returned to the corresponding pool. In cloud conferencing, the chair or organizer of a conference can pay the cloud broker according to conference scale, duration, function and so on based on the iSLA established between them.

B. Experiments

To validate the videoconferencing scenario within the cloud networking architecture, we use our algorithm of best CSP and NSP selection with QoS splitting. This algorithm enables determining the best delivery paths of videoconferencing streams among the VMs allocated for videoconferencing servers. It is possible that VMs are distributed in different cloud data centers. The objective of the cloud broker is maximizing the total throughput across all conferences, yet without violating end-to-end delay constraints.

CloudSim toolkit supports both system and behavior modeling of cloud system components such as data centers, VMs and resource provisioning policies. Currently, it supports simulation of cloud computing environments consisting of both single and inter-networked clouds and simulation of network connections among the simulated system elements. Moreover, it exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked cloud computing scenarios. CloudSim offers a self-contained platform for modeling clouds, service brokers, provisioning, and allocations policies [25].

```
45          CloudSim.init(num_user, calendar, trace_flag);
46          CloudBroker broker = createBroker("Broker_0");
47          int brokerId = broker.getId();
48          Datacenter datacenter0 = createDatacenter("Datacenter_0");
49          int datacenter0Id = datacenter0.getId();
50          Datacenter datacenter1 = createDatacenter("Datacenter_1");
51          int datacenter1Id = datacenter1.getId();
52          Datacenter datacenter2 = createDatacenter("Datacenter_2");
53          int datacenter2Id = datacenter2.getId();
54          Datacenter datacenter3 = createDatacenter("Datacenter_3");
55          int datacenter3Id = datacenter3.getId();
56          user u1 = new user("user1");
57          int u1Id = u1.getId();
58          user u2 = new user("user2");
59          int u2Id = u2.getId();
60          NSP nsp1 = new NSP("NSP1");
61          int NSP1Id = nsp1.getId();
62          NSP nsp2 = new NSP("NSP2");
63          int NSP2Id = nsp2.getId();
64          NSP nsp3 = new NSP("NSP3");
65          int NSP3Id = nsp3.getId();
66          NSP nsp4 = new NSP("NSP4");
67          int NSP4Id = nsp4.getId();
68          //configure network
69          NetworkTopology.buildNetworkTopology("topology.brite");
```

Fig. 8. Initialization Example of CloudSim Entities with Two CSUs

Firstly, we create our proposed architecture, i.e. four datacenter entities, ten CSU entities, one cloud broker entity, and four NSP entities that connect other entities using BRITE topology for modeling link bandwidth and associated latencies (Fig. 8). Then, all entities are initiated at the beginning of the simulation and the interactions process described in the *section IV.C* is simulated. Note that the videoconferencing initiator is an enterprise (CSU) that specifies requirements and establishes the iSLA with the cloud broker and a Graphical User Interface (GUI) is proposed to collect its preferences (Fig. 9).

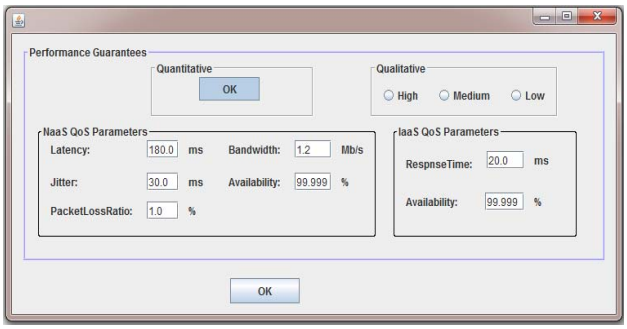

Fig. 9. GUI for CSU Preferences

After selection of best CSPs and NSPs with QoS splitting, the broker allocates required VMs to the CSU. Then, entities start the videoconferencing. Finally, at the end of the videoconference the broker releases allocated VMs.

C. Evaluation

To assess the performance of our proposed framework while considering the videoconferencing service, we evaluate nine videoconferencing scenarios. Each scenario is between two CSUs. We measure the streaming delay from source to destination for different cases corresponding to different CSU geographical locations. Formally, delay is defined as a period of time necessary for a bit of data to travel from one endpoint to another. In our case, the delay (D) is calculated using the equation (8), based on (1) and (6), where c_j is the selected CSP. The delay and the jitter, which is the delay variation, are not critical for non-real time data packets, but they are very

important QoS parameters for real time interactive applications such as videoconferencing. An end-to-end delay greater than 200ms could cause a degraded videoconferencing service.

$$D = \sum_{i=1...m} L_1(N_{ij}) + L_1(c_j) + R(c_j) + L_2(c_j) + \sum_{i=1...m} L_2(N_{ij}). \quad (8)$$

As shown in Figure 10, the results reveal a good streaming delay achieved by our proposal, as compared to a static CSP and NSP selection without using our brokerage service. The end-to-end delays are well controlled to keep queuing delays to the minimum while packets traverse through intermediate datacenters on their paths.

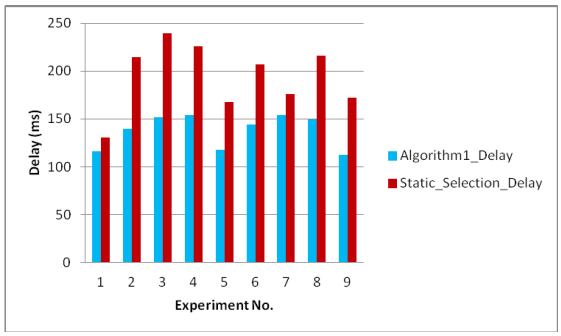

Fig. 10. Comparison of end-to-end Delays

VII. Conclusion

In this paper, we presented a framework for self-establishing a service level agreement within cloud networking environment. At first, we have proposed an architecture of cloud networking. We have focused on QoS parameters for IaaS and NaaS services. In addition, we have presented the calculation and splitting of these QoS parameters then the establishing of an inter-cloud service level agreement (iSLA). Then, we have proposed the use of autonomic cloud managers for the iSLA self-establishment. Finally, we evaluated our framework performance using a videoconferencing application as a proof of concept, and we have obtained good results in terms of end-to-end delay evaluation.

As a future work, we are working on a federation cloud networking architecture that we will compare to the broker architecture specified in this paper in terms of performance evaluation. Then, we intend to enable the established SLA self-management using the MAPE-K control loop while introducing security parameters in our proposed iSLA and study their impact on QoS parameters.

References

[1] P. Murray, A. Sefidcon, R. Steinert, V. Fusenig and J. Carapinha, "Cloud Networking: An Infrastructure Service Architecture for the Wide Area," Future Network & Mobile Summit 2012: 21st Annual Conference Proceedings, 2012.

[2] H. Feng and W. Wu, "Framework and User Migration Strategy of Cloud-Based Video Conference Multi-Gateway System," 19th International Conference on High Performance Computing (HiPC), p.p. 1 - 8, Dec 2012.

[3] M. Armbrust, A. Fox and R. Griffith, "Above the clouds: a berkeley view of cloud computing," Technical Report, EECS Department, University of California, Berkeley, Feb 2009.

[4] Leavitt. N., "Is cloud computing really ready for prime time?," Journal Computer, Volume 42 Issue 1, p.p. 15-20, Jan 2009.

[5] NIST, "The NIST Definition of Cloud Computing," Special Publication 800–145, USA, 2011.

[6] P. Schoo, V. Fusenig, V. Souza, M. Melo, P. Murray, H. Debar, H. Medhioub and D. Zeghlache, "Challenges for Cloud Networking Security," 2nd International ICST Conference on Mobile Networks and Management, pp.298–313, 2010.

[7] ITU-T, "Focus Group on Cloud Computing, TR part 1–3," 2012.

[8] "SAIL Project," http://www.sail-project.eu/.

[9] "FoSII - Foundations of Self-Governing ICT Infrastructures," http://www.infosys.tuwien.ac.at/linksites/FOSII/index.html.

[10] "Contrail-project," http://contrail-project.eu/.

[11] "CPIP IEEE, http://standards.ieee.org/develop/project/2301.html

[12] "SIIF IEEE, http://standards.ieee.org/develop/project/2302.html

[13] "Open Cloud Computing Interface (OCCI)," http://occi-wg.org/.

[14] "GICTF," http://www.gictf.jp/index_e.html.

[15] T. Benson, A. Akella, A. Shaikh and S. Sahu, "CloudNaaS: a cloud networking platform for enterprise applications," In Proc of 2nd ACM Symposium on Cloud Computing, 2011.

[16] I. Goiri, F. Julia, J. Fito, M. Macıas and J. Guitart, "Resource-Level QoS Metric for CPU-Based Guarantees in Cloud Providers," GECON 2010, LNCS 6296, pp. 34–47, 2010.

[17] G. Cicotti, L. Coppolino, R. Cristaldi, S. Antonio and L. Romano, "QoS Monitoring in a Cloud Services Environment: the SRT-15 Approach," Epsilon srl, Italy, pp. 15–24, 2011.

[18] Y. Kouki and T. Ledoux, "CSLA: a Language for improving Cloud SLA Management," In International Conference on Cloud Computing and Services Science, Portugal, April 2012.

[19] P. Patel, A. Ranabahu and A. Sheth, "Service Level Agreement in Cloud Computing," OOPSLA, 2009.

[20] H. Ludwig, A. Keller, A. Dan, R. King and R. Franck, "Web service level agreement (WSLA)," IBM Corporation, 2003.

[21] Y. Wu, C. Wu, B. Li and F. Lau, "vSkyConf: Cloud-assisted Multi-party Mobile Video Conferencing," Second ACM SIGCOMM workshop on Mobile cloud computing, 2013

[22] Y. Feng, Ba. Li and Bo. Li, "Airlift: Video Conferencing as a Cloud Service using Inter-Datacenter Networks," 20th IEEE International Conference on Network Protocols, 2012.

[23] I. Brandic and E. Feller, "Energy and QoS-aware workload management in clouds," Research report of COST group, 2011.

[24] M. Huebscher and J. Mccann, "A survey of Autonomic Computing —degrees, models and applications," CSUR, 2006.

[25] R. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environment and evaluation of resource provisioning algorithms," Software: Practice and Experience, 2011.

[26] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, June 2002.