A Framework for SLA Establishment of Virtual Networks based on QoS Classes

Rafael L. Gomes, Luiz F. Bittencourt, Edmundo R. M. Madeira
Institute of Computing (IC), University of Campinas (UNICAMP), Brazil
Email: {rafaellgom,bit,edmundo}@ic.unicamp.br

***Abstract*—The Internet has become the primary means of communication, where many companies have multiple Internet service providers to establish a Service Level Agreement (SLA). Within this context, this paper shows a framework for virtual networks establishment based on QoS classes, where traffic classification techniques are used to forward the traffic to the provider defined in the SLA. It is expected to ensure QoS requirements for each traffic class, and also to reduce the company costs. The experiments show the capacity of the framework to classify the traffic performing a feasible traffic differentiation according to the defined QoS classes.**

I. Introduction

The Internet has become the base for many companies to offer their services. To guarantee Quality of Service (QoS), companies establish Service Level Agreements (SLA) with their Internet Service Providers (ISP). Currently, there is a consensus that the Internet needs to be updated, creating the so-called Future Internet [1]. Along with this, the Network Virtualization (NV) approach has arisen as one of the most important technologies for the Future Internet.

Applications have different quality requirements, where some applications need a complex network infrastructure to experience good QoS. On the other hand, other applications require just a basic network infrastructure to ensure a good QoS. Tied to this fact, the provider charges more to offer a high quality infrastructure. So, the companies end up paying a high value to have a complex infrastructure, when only part of the existing traffic requires such level of quality, generating unnecessary expenses.

Within this context, this paper proposes a framework for SLA Establishment of Virtual Networks (VN) based on QoS Classes. The framework aims to: define some QoS traffic classes, negotiate the desired virtual networks with the ISPs (the SLA negotiation), bind the QoS classes with the negotiated VNs, and forward the client's traffic to the appropriated VN according to the class of the traffic. The idea of the framework is to reflect the particular needs of each client, with the corresponding cost for each network configuration.

The virtualization allows the framework to establish the VN through a full negotiation: it negotiates not only *resources* (measurable parameters), but also the *features* (parameters behaviors) of the virtual network with the provider.

For example, a given client may need to establish a network with resource reservation and traffic engineering (complex network), while for another client a simple OSPF with Ipv4 network may be enough.

This paper is organized as follows. Section II presents related work regarding frameworks for SLA and/or virtual networks. Section III describes the proposed framework, as well as the modules in the client and providers. Section IV details the traffic classification approach used. Section V summarizes the SLA negotiation process. Section VI presents and explains some results of the developed forward schema, and Section VII concludes the paper and presents future work.

II. Related Work

In this section we describe some important works related to SLA frameworks.

Venticinque et al. [2] present a component of mOSAIC framework, called Cloud Agency, which is a broker of Cloud resources from different providers that tries to fulfill the requirements of the applications. According to the available offers, it will generate a SLA that represents the resource negotiation Cloud providers.

Comuzzi et al. [3] show a technical architecture for a multilevel SLA management framework. The work discusses the fundamental components and covers the core SLA management life cycle, including negotiation, provisioning, monitoring, and adjustment.

Zou et al. [4] propose an architecture called Virtual Automated Trust Negotiation (VATN) to centralize ATN policies and credentials for multiple virtual machines in a physical node into a privileged virtual machine.

Zaheer et al. [1] show V-Mart, an open market model for automated service negotiation in network virtual environments for Virtual Network Providers (VNP) and Infrastructure Providers (InP). For InPs, V-Mart fosters an open competition through auctioning, and the VNP can disseminate a request for quotation when it desires to set up a virtual network.

None of the papers found in the literature focus on the development of a framework that performs the establishment of virtual networks based on QoS classes by applying traffic classification techniques, which is the proposal of this work.

III. Proposed Framework

The Internet has been growing and many companies use it as a basis for their services. To utilize a service available through the Internet, clients usually establish with the desired provider a SLA that specifies properties which must be maintained during service provisioning. Currently, there is no guarantee

978-3-901882-50-0

of service level on the Internet. Hence, there is a consensus that it needs to evolve to a Future Internet.

Based on this scenario of Future Internet and the needs of the users towards the resources provided by ISPs, the proposed framework aims to provide QoS for the client in a multiprovider context. The framework aims to execute all the steps to establish a virtual network between the client and an ISP, comprising: the definition, negotiation, deployment, and usage (traffic differentiation and forwarding) of the virtual networks.

The framework performs a full negotiation of the virtual networks, i.e., it negotiates with the provider not only network resources (bandwidth for example), but also the protocols of the virtual network. Besides this, classification techniques are used to forward the traffic to the virtual network negotiated to be the most suitable for the defined class.

Using the framework, the client expects to meet the requirements of each defined traffic class, since each class has different networking requirements to achieve QoS, such as delay, jitter, and others.

Next, the modules defined in the client side and provider side are shown, where a description of the functionality of each module is presented.

A. Client Modules

Below, the tasks for the Client are listed:

- Define the QoS Classes;
- Generate the mapping between the QoS Classes and the defined Virtual Networks;
- Specify the SLA proposal;
- Perform the negotiation;
- Decide with which provider to deploy the virtual network;
- Classify the flows;
- Forward the traffic according to the negotiation and the flow classification.

Therefore, the client needs the modules described as follows, which are presented in Figure 1.

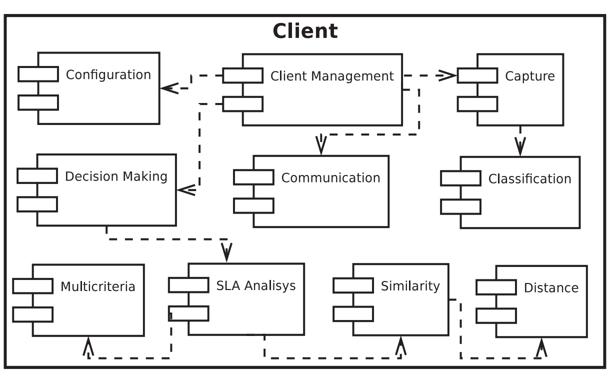

Fig. 1. Client Components

1) Client Management: This module manages the general aspects and issues related to the SLA, first specifying it, and then using the other modules to perform the negotiation of the virtual networks with the providers.

2) Client Configuration: The *Configuration* module handles the data regarding: the providers for the negotiation process, the general information, the priority of each criterion on the negotiation process, and the configuration for the similarity calculation between the protocols.

3) Communication: This module is responsible for the message exchange between the Client and the providers.

4) Capture: The *Capture* module performs the packet capture and flow management. The flow management is to classify identified flows, verifying which flow is active. To perform the flow classification, the *Classification* module is used.

5) Classification: The *Classification* module classifies the flows according to the following process (Detailed in Section IV): packets are classified individually until the final decision, based on five packets or two equal classifications in a row, in order to avoid extra delay to the flow.

6) Decision Making: This module chooses the provider that best fits the requirements in the SLA for each virtual network based on the similarity values and the price charged by the provider. The *Multicriteria* module is used in the decision.

7) Multicriteria: This module defines the Multicriteria Decision Making (MCDM) method used to evaluate the criteria and performs the decision, choosing the most suitable provider to deploy a specific virtual network of the SLA. Any MCDM method existing in the literature can be used in this process.

8) SLA Analysis: The *SLA Analysis* module compares two SLAs, evaluating the virtual networks requested in the SLA. It is used to make a counter-proposal for a SLA request or to evaluate the counter-proposal received with the original SLA.

Regarding the protocols negotiation, when a protocol requested by the client can not be offered by the provider, it must send another option (if available) in the SLA counter-proposal. To choose the most suitable option for the counter-proposal, the provider uses the modules *Similarity* and *Distance*.

9) Similarity: The *Similarity* module generates the similarity values between the received and requested protocols.

10) Distance: The *Distance* module calculates the distance between two objects, where any distance method for nominal variables can be used in this process. The distance represents an inverse proportional relation with similarity.

B. Provider Modules

Below, the tasks for the Provider are listed:

- Evaluate the SLA proposal received and send a counter-proposal, based on the available resources;
- If the provider is chosen, it must deploy the virtual network and notify the client that the VN is available.

So, the provider needs the modules described as follows, which are illustrated in Figure 2. The *Communication*, *SLA Analysis*, *Similarity* and *Distance* modules have the same functionality previously described in Section III-A. The definition and functionalities of other modules are as follows.

1) Provider Management: This is the main module of the provider, which controls the provider side as a whole. It manages when a service is provisioned until the virtual network deployment is performed after the negotiation process.

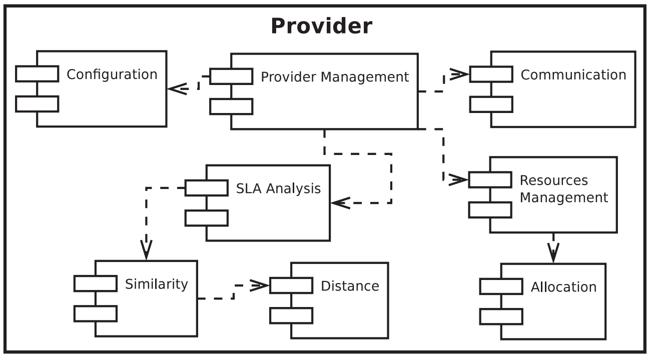

Fig. 2. Provider Components

2) Provider Configuration: The *Configuration* module loads the configuration that is requested by the *Provider Management* module. The *Configuration* must provide the information related to: available resources, the protocol availability, and the provider data for the SLA.

3) Resource Management: The *Resource Management* module is responsible to manage the network resources available in the provider, i.e., to inform how much of a certain resource is currently available for the negotiation process. The deployment and the release of the virtual networks are made by the *Allocation* module.

4) Allocation: This module controls the deployment of the virtual networks, performing the allocation and deallocation of resources for each virtual network in the SLA, as well as the application of the desired protocols in the virtual network.

C. Framework Overview

Generally, to establish the virtual networks using the proposed framework, the following sequence of steps must be performed:

1) Define the QoS classes for the classification agent;
2) Apply the classification module;
3) Perform the match up of the parameters with each provider;
4) Define the parameters of the negotiated virtual networks in the SLA, and the mapping between the QoS classes and the virtual networks;
5) Send the SLA proposal to each provider;
6) Each provider analyzes the SLA proposal and sends a counter-proposal with the available resources and protocols that best fit the client requirements;
7) The client analyzes the counter-proposals and chooses the provider most suitable to deploy each of the virtual networks defined in the SLA;
8) The client notifies each chosen provider;
9) The provider verifies the compatibility between the received SLA and the available infrastructure, and deploys the negotiated virtual network;
10) The provider notifies the client that the virtual network(s) is(are) available;
11) The client receives the notification, allowing the forward of the flows through the framework, and starts to use the negotiated infrastructure.

Figure 3 shows the sequence of steps previously described, where the numbers presented in the figure represent each of the described steps. Some modules were omitted from the diagram for the sake of simplicity and for an easy visualization.

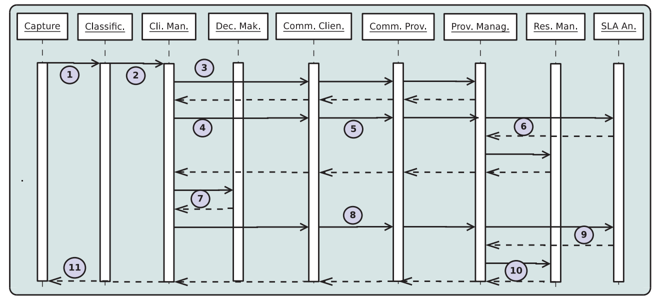

Fig. 3. Steps performed following the Framework.

IV. Classification Agent

The classification agent aims to classify the network packets into QoS classes in real time. So, we defined five traffic classes: *Audio* (online radios, VoIP, and others), *Control* (control and management packets), *Data* (burst traffic using TCP protocol), *Video On Demand* (on demand video traffic) and *Real Time Video* (including live stream transmissions).

We evaluated two Machine Learning techniques: Naive Bayes and Decision Tree [5]. In our previous work [6], we evaluated some ML techniques for Four-QoS class classifier. The Naive Bayes and Decision Tree techniques performed better. Hence, now we evaluate these two techniques comparing the performance evolution from Four-QoS class to the new Five-QoS class approach.

In this new classification approach (Five-QoS class classifier) the *Video* class was divided in two: *Video on Demand* and *Real Time Video*. This division occurs due to the fact that these two kinds of traffic have different QoS requirements, despite both being related to video transmissions.

The approach for training the classifiers follows the same applied in previous work [6]. After trained, we determine the performance of the classifiers through the Accuracy metric [7]. It represents the ratio between the sum of the correct classifications and the total size of the sample.

The Four-Classes Naive Bayes, Five-Classes Naive Bayes, Four-Classes Decision Tree and Five-Classes Decision Tree get the performance of 0.949, 0.951, 0.956 and 0.974, respectively. Based on this values, it is noted that all classifiers get a good performance. The performance increase in the Five-QoS class classifier can be justified by the fact that the division of the *Video* class decreases the intersection that existed between the two new classes and the others.

Furthermore, we evaluate the time for classification of each classifier. In this evaluation, we applied a confidence interval of 95%, which is showed in Figure 4. The Five-QoS class Naive Bayes and the Four-QoS class Naive Bayes have almost the same time for classification. However, the time for classification of the Five-QoS class classifier Decision Tree is lower than the Four-QoS class Decision Tree. This occurs because the tree topology length generated in the training

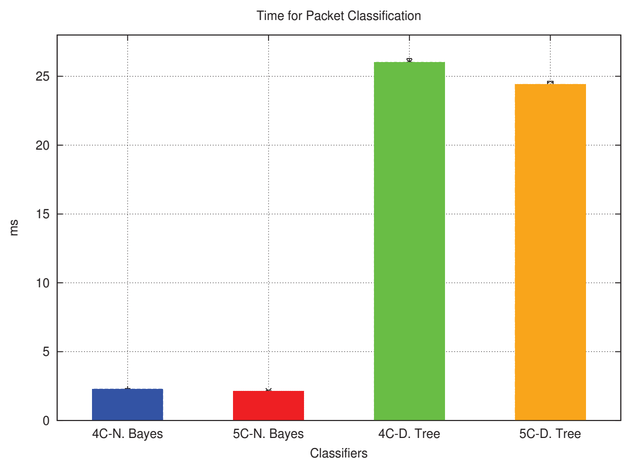

Fig. 4. Time for Classification of One Packet

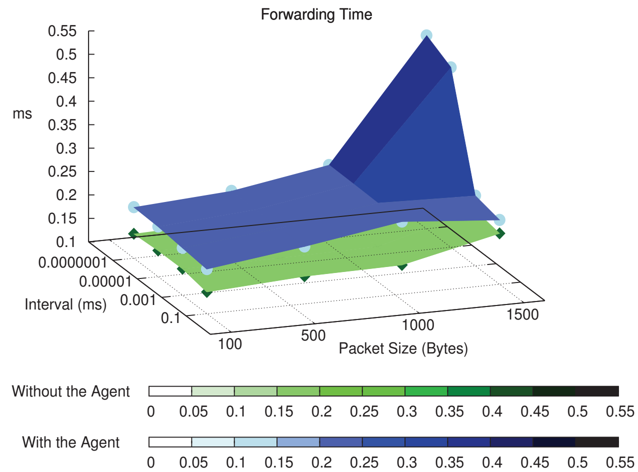

Fig. 5. Forward Time Results

process for the Five-QoS class classifier was smaller than that one for the Four-QoS class classifier due to the division of the *Video* class.

The Five-QoS class Naive Bayes classifier was chosen to be part of the classification agent. Aiming that the proposed agent gets a good packet classification rate with a small impact in the end-to-end delay of the applications.

After the classification process, the class of the packet is placed in the Type Of Service (ToS) field in the IP packet header. It allows the identification of the class of the packet in the forwarding process.

V. Virtual Network Negotiation

Regarding the virtual network negotiation, our previous work [8] proposed a negotiation protocol for the negotiation of flexible parameters, allowing the negotiation of any Virtual Environment (VE), so the protocol can negotiate virtual networks and, for instance, virtual environments in clouds. The negotiation process is only a small part of the framework as a whole, and to detail the negotiation process is not the focus of this paper. Here, we have to keep in mind that the negotiation chooses one of the possible ISPs to deploy the virtual network.

VI. Traffic Differentiation

When the negotiation is finished, the framework is responsible to classify the flows, and according to the class, forward the packets to the more suitable provider. So, the framework must perform this forwarding, minimizing delay for the packets, in order to not compromise the QoS of the applications.

The forwarding process has a cost (delay generation), so we evaluate the capacity of the forwarding process and possible overheads in the packet end-to-end delay. To evaluate it, we compare the time taken for forwarding with the framework against no framework use.

In the experiments, we varied the size of the packets and the interval of sending packets. The results are shown in Figure 5. We note that in cases with small packet size (100 Bytes), regardless of the sending interval, using the agent causes a very small impact, with values for packet forwarding around 0.05 milliseconds.

In extreme cases, packets with the maximum size (1500 Bytes) and very small intervals (10^{-7} ms), the framework generates a maximum delay of 0.3 milliseconds. Thus, these experiments indicate that the framework does not cause significant delays to the flows.

VII. Conclusion

This paper presents a framework for virtual network establishment based on QoS classes, where traffic classification techniques are used to forward the traffic to the provider defined in the SLA.

The experiments showed the effectiveness of the classification process, attesting to the capacity of the framework to classify and forward the flows with a good accuracy in a feasible time. As future work, we intend to extend the framework for an end-to-end scenario.

Acknowledgment

The authors would like to thank FAPESP (2012/04945-7), CNPq and RNP for the financial support.

References

[1] F. Zaheer, J. Xiao, and R. Boutaba, "Multi-provider service negotiation and contracting in network virtualization," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, april 2010, pp. 471 –478.

[2] S. Venticinque, R. Aversa, B. Di Martino, M. Rak, and D. Petcu, "A cloud agency for sla negotiation and management," in *Euro-Par 2010 Parallel Processing Workshops*, 2011, vol. 6586, pp. 587–594.

[3] M. Comuzzi, C. Kotsokalis, C. Rathfelder, W. Theilmann, U. Winkler, and G. Zacco, "A framework for multi-level sla management," in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*.

[4] D. Zou, S. Du, W. Zheng, and H. Jin, "Building automated trust negotiation architecture in virtual computing environment," *The Journal of Supercomputing*, vol. 55, pp. 69–85, 2011.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[6] R. L. Gomes and E. R. M. Madeira, "A traffic classification agent for virtual networks based on qos classes," *Latin America Transactions, IEEE*, vol. 10, no. 3, pp. 1734 –1741, april 2012.

[7] M. Gonen, *Analyzing Receiver Operating Characteristic Curves With SAS (Sas Press Series)*. SAS Publishing, 2007.

[8] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira, "A generic sla negotiation protocol for virtualized environments," in *Proceedings of 18th IEEE International Conference On Networks (ICON 2012)*, 2012.