Dynamic architecture based on network virtualization and distributed orchestration for management of autonomic network

Guy Saadon
Telecom ParisTech,
Paris, France
Jerusalem College of Technology,
Jerusalem, Israel
guys@g.jct.ac.il

Yoram Haddad
Jerusalem College of Technology,
Jerusalem, Israel
haddad@jct.ac.il

Noemie Simoni
Telecom ParisTech,
Paris, France
simoni@telecom-paristech.fr

***Abstract*— In network management architectures of 5G and IoT networks, standardization groups often consider the network resource virtualization layer between the physical network and the SDN controller, as a means to allow deployment and placement of network services with their virtual network functions. However, the following question arises: is this layer enough to react to real-time changes originating from customers or the network without interrupting the service? We consider that a dynamic architecture should allow different and evolving assemblies to be provisioned during a session, in order to meet modification requests without requiring total redesign of the network service. Therefore, in this study, we propose an enhanced architecture. This novel architecture adds a network virtualization layer above the SDN controller with its associated orchestrator. Then, efficiently distributing orchestration among the different layers ensures network autonomy. In this context, we show how real-time service modifications and network failures are handled without losing the existing services and how network management gains additional dynamicity and flexibility.**

***Keywords*— Architecture; Autonomic network; Orchestration; SDN; Service dynamicity; Virtualization**

I. Introduction

5G and IoT telecommunication networks are required to manage critical, considerably diverse, and constantly changing demands. For critical machine communication network for transportation such as autonomous cars, working groups propose an additional and separate secured network slice with its own capacity, extremely low latency, and availability requirements within the future network [1]. Orchestrators are defined to program autonomic behaviors to support the permanent changes emanating from applications and the network.

Scientific literature and several forums and institutes (also referred to as standard development organization (SDO)) have proposed different models to support dynamic and autonomic network requirements. In this context, the following questions arise: do these current models and architectures fit all dynamic requirements? How should we define and organize orchestration to answer the need for autonomic behaviors at different layers of the network? How should the different functions interact to allow a continuous service flow without disruption?

We believe that dynamicity must even support the changes of application requests during a session. By autonomic requirement support, we intend to reach a zero-touch network, by using an effective organization of orchestrators. In this study, we focus on the aforementioned problems by proposing a novel architecture using the following features: adding a network virtualization layer above the SDN controller with its associated orchestrator to improve dynamicity and, distributing the function of orchestration to enhance network autonomy. In Section II, we analyze the work realized in related research and the SDOs and present the resulting problem. In Section III, we detail our proposition: a novel architecture that allows dynamic changes during a session, an organizational model that describes the distribution of the orchestration to allow improved flexibility, scalability, and performance, and a functional model that describes the service lifecycle to ensure network autonomy. We show how our proposition can solve the aforementioned questions with the help of use cases. In the conclusion section, we summarize the advantages of this architecture.

II. Related work and problem

A. Scientific literature

In current telecom networks, network management (NM) encounters several problems with respect to service request (SR) constraints at the application level and its end-to-end (E2E) constraints, and at the network resource level, simultaneously. Numerous researches focus on this problem. Some works emphasize the application E2E service constraints. Area particularly interesting investigation was done by Millnert [2]. This investigation uses mathematical models that take into account the physical deployment of service elements in closed proximity in order to limit the E2E delay. Reference [3] proposes a mathematical model to optimize the number of machines needed for virtual network function (VNF) to guarantee the E2E constraints for network service (NS) setup by considering buffer and computing virtual machine (VM) constraints. However, there is no off-line procedure or scheduling model to determine the best E2E service and compare it with the chosen calculated one. T. H. Tan, et. al. [4] focus on improving local resource constraints. They notably reduce the number of options for web services at the level of local constraints by effectively discarding the service candidates that cannot satisfy global constraints. A. Sheoran et. al. [5] propose to limit the E2E delay constraint at the resource level by placing VNFs by affinity using micro-service aggregates within containers in the datacenter. These important contributions improve performance. However, they do not consider dynamic changes of services including component modifications.

B. SDOs related work

Several SDOs [6] contribute significantly to future NM. Among them, a few worth citing are: the ONF forum is active at

978-3-903176-24-9 © 2019 IFIP

the lower NM layers and has specified the OpenFlow™ protocol [7-9]. The MEF forum mostly works at the E2E service level with the third network and LSO [10-12]. MEF encourages active cooperation with forums such as ONF or TMF to converge to standard interfaces. The TMF forum expresses the need for network and resource slicing for an automated cross-layer orchestration of those resources per domain, inter-domain, and cross-partners, depending on the business model [13-16]. The ETSI Institute has performed important work in the network function virtualization (NFV) domain [17,18]. By decoupling service functions from infrastructures, NFV provides benefits such as simplified and more flexible new services development and introduction and lower costs. Open sources initiatives such as open network automation platform (ONAP) and central office re-architected as a datacenter (CORD) also propose a complete architecture implementation [19-20]. A careful reading reveals some gaps such as unclear frontier between the OSS and the orchestration or monolithic orchestration (except for TMF). The aim of these NM architectures is to satisfy every type of SR. However, the following questions arise: once supported, is this still available for any dynamic changes within the same session? Are these SDO architectures capable of reaching a zero-touch network?

C. Problem description

Our proposed architecture is motivated by the following problem. According to the scientific literature and main SDOs, the way SR constraints are considered at both the application and resource levels, is partially covered, especially in case of dynamic changes. When, during a session, the applicative components should be added or migrated dynamically (on demand), the actual implementations cannot assure service continuity, and autonomy can be threatened. The service can be lost and must be redesigned, provoking delay and QoS problems. During a VoIP conversation, an end-user may want the operator to physically E2E encrypt their call. For instance, they want to transmit their credit card number for payment. This is an on-demand request for dynamic modification of a call that cannot be realized in the same session at the network level by current architectures. This may be demanded only at the application layer if the two persons who communicate have the same application. It is worth mentioning that with an architecture such as ONAP [19] that is based on ETSI MANO, the closed control loop and other real-time monitoring modules allow calculating another path and modifying the running service. However, the addition of another VNF function such as encryption, with its queuing and processing delays, may generate serious service degradation. By using a control loop, there is no guarantee that the E2E QoS constraints will be respected. At least several iterations of the control loop will be necessary.

III. Proposed Architecture

The objective is to provide enhanced architecture to solve the E2E and local constraints while creating or dynamically modifying a service and maintain network autonomy. In this context, we consider the following questions: at which levels of the architecture can these problems be identified? Which modules must be added or modified? What are their interactions? Can this avoid resetting the complete service?

A. Architecture

Our architecture tries to synthesize some research concepts regarding NFV and network slicing in the context of application and resource constraints along with the work realized in SDOs. TMF among several proposed architectures, details in [13] every NM layer and separates the orchestration into four distinct levels (see Fig. 1): user, applications and services, network resources, and technology orchestrators. They can communicate with each other via API like REST. With this functional distribution, each orchestrator is responsible for one NM level. Once the work of an orchestrator is complete, it does not have to be re-done at a lower one. Further, each orchestrator trusts the work and decisions of the orchestrator above it. For instance, the second orchestrator selects the application server. The following ones will go on building the SR accordingly.

To understand if this distribution in four Orchestrators is enough to answer our concerns about dynamicity and autonomy, we analyze the roles and responsibilities of these four TMF orchestrators as follows. Process distribution usually allows improved flexibility, scalability, and performance. In addition, orchestration should allow efficient distributed decision-making process according to the different NM layers. This distribution led us to understand to deal separately with E2E constraints and network resource constraints: one at the network orchestrator and the other at the resources' orchestrator. The need for this separation is also mentioned in [21] by pointing the problems in combining SDN and NFV into a unique architecture framework for service provisioning. To this end, this study proposes a two-dimensional abstraction model where SDN is related to split into planes concept and NFV to the five network layers concept. It also proposes an additional dimension of abstraction to decouple service functions and network infrastructures. Therefore, we realize the necessity of a multi-layer NS to allow user service transparency from one side and network resource continuity from the other side: a type of virtual NS composer (planning component). This allows the creation of alternative virtual service if required but, without removing the running service until the modified one is ready. Consequently, after the creation of this new virtual service, the running service can switch to it. This explains the need for an additional virtual layer.

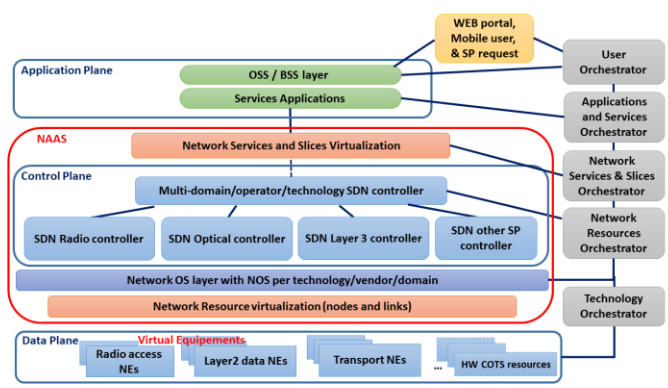

Fig. 1. Proposed architecture.

That's why, in the Network-as-a-Service (NaaS) part, we introduce a fifth orchestrator: the "network service and slice (NSS) orchestrator", related to service and slice attachment. Service attachment means service assignment without resource allocation, i.e. a virtual service deployment. This orchestrator operates at the network level, and it is responsible for SR (re-)design. This SR "design" includes the main components of a service (nodes (VNFs), links, and endpoints) with its application's QoS constraints. In MEF, the multi-domain aspect between cross-partners is covered, at the network level, at the service orchestrator using Interlude interface. Likewise, the NSS orchestrator will ensure this role. This orchestrator is a split-off from the network resource one. The latter works directly with the general SDN controller (multi-domain /operator /technology), which is responsible for SR placement using the network resource virtualization.

This new NSS orchestrator cooperates with an additional virtualization layer: the "network services and slices virtualization," located at the northbound interface of the NaaS. This virtualization layer is an abstraction of the network resources. For instance, a group of switches in a business building can be represented as one entity with its general capacity, delay, and constraints. As explained in [22], tight coupling between service provisioning and network infrastructure quickly becomes a barrier to rapid and flexible service deployment. With the introduction of this network virtualization layer, the SR attachment is created in two phases. First, at the service level it is built with its requirements coming from applications: nodes (applications) links (transactions), end-points, and E2E constraints. Then, this built SR distributes its component functions over the network abstraction considering nodes (VNFs), links, and local constraints. This selection gives rise to a negotiation. Every service type and VNF are dimensioned and calibrated with their properties and constraints: throughput, maximum error ratio, and transport time. These service components are registered in the NS catalog that includes E2E service constraints and the VNFs catalog with more local constraints like ETSI MANO architecture [6]. For instance, a call service cannot overcome delay of 150 msec., or a virtual NodeB must be in the building closest to the antenna it serves. By introducing this abstraction layer, we draw a parallel between the cloud computing and the networking worlds. SaaS, PaaS, and IaaS are parallel to virtual network VNaaS, VNPaaS, and VNIaaS respectively. The new layer corresponds to the VNPaaS and allows a horizontal approach, where the SR is first attached at the PaaS and VNPaaS and then physically allocated at the IaaS and VNIaaS. Therefore, in this attachment phase, the service is virtually built and optimized, by ensuring that the aggregation of the local constraints does not exceed the global E2E service limitation. If a physically allocated VNF does not respect its local constraints, it can be physically migrated to another VM. Consequently, virtual deployment and physical placement are separated and entrusted to well-identified and independent managers, each having its own constraints to respect. This "virtual deployment" respects the global SLA regardless of the physical resources.

Then, the attached SR is sent to the general SDN controller and VNF manager for physical placements. The SDN controller controls different domain SDN managers that can be viewed as virtual networks with their own controllers [21]. Therefore, this network virtualization layer can impart more dynamicity to the network. This virtualization layer, unlike the previous upper layer, does not consider the service provider (SP) services and server locations; it is centralized on the network itself. In this layer, each network slice request is independent of the others. In case no alternative slice is found, this can be reported to the OSS layer for planning a network extension. BSS receives statistics from the network for billing and business applications. At the application plane, OSS still appears in main SDOs architectures with a limited functionality such as setup or long-term functions [6]. By creating a virtual service, we express the requested QoS constraints for the IT resources. We express the local and E2E networking constraints for this SR, i.e., the offered QoS NSs with the help of the NSS virtualization layer. The attached service is a result of the negotiation between the requested QoS constraints and the offered QoS NSs. While addressing the SDN controller, the SR is already virtually deployed with its E2E constraints. Therefore, we can summarize our contribution as follows: we add an NSS virtualization layer for NS virtual deployment thereby, allowing loose coupling between SR attachment and physical placement. We separate the resources orchestrator into two functional entities.

B. *Organizational proposal*

In this section, we describe two scenarios of service setup and dynamic change, to understand how the different modules of our proposal interact and complement each other. So, we can better identify the decision-making process in the different orchestrators and the importance of the NSS virtualization.

The end-user or the SP requests new services or makes changes to existing ones, going through a portal. This user request is accepted when it is authentified, authorized and respects the SLA (step 1, Fig. 2). The role of the user orchestrator is to qualify a user request in a standard and declarative format that is understandable by the second orchestrator with the help of the policy agent. It can also prevent over-subscription via admission control process.

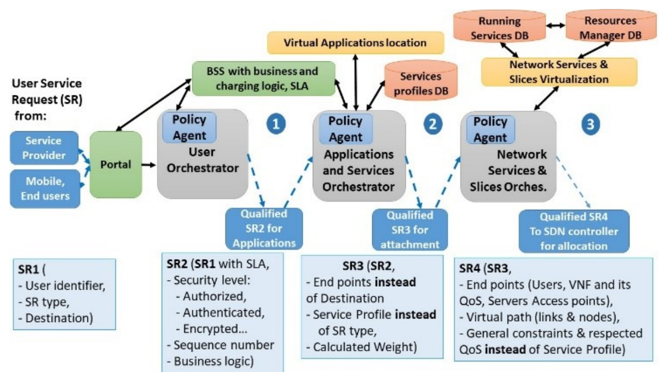

Fig. 2. Organizational proposal at the upper layers.

Once qualified, the SR includes security level, service type, and destination. It is directed to the applications and services orchestrator. Here, we describe an SP network slice request that illustrates the setup phase of a zero-touch network. SP signs an SLA with an operator for a network slice with defined QoS and the connection points to the operators' network. The second orchestrator associates the end-points and their network location and looks for the corresponding service type in the service profile (catalog) database (DB). The request includes QoS parameters such as minimum throughput, maximum delay, and error ratio [23]; list of access points; security level; protection type; and functions associated to the service. The SP request is then addressed to the policy agent that includes decision-making rules related to business logic or regulation. With the help of the BSS and the policy agent, a service weight is calculated according to the customer importance, the signed SLA, and the SR priority. High weight can be reserved for public safety services such as firefighters. The qualified SR is then directed to the NSS orchestrator that calculates the network slice with the help of NSS virtualization (step 2, Fig. 2). The network slice is attached with its components: the VNFs with their properties and constraints (located in the resource manager catalog) and the links of the slice.

In step 3, this request is sent to the general SDN controller for path calculation with respect to QoS at the resource level. If the resources are free, the network slice is allocated. The network resources orchestrator works with the main SDN controller and synchronizes resource applications such as self-healing or path optimization running in background. The running services and resource manager DBs, the Performance monitoring (PM) handler, and the resources orchestrator are updated. The PM handler allows real-time monitoring of the running slices and of the resources. It can update the orchestrators or the OSS if required.

When dynamic changes occur in the network slice, several components may have to be modified, e.g., at a large musical evening event (The SP charging policy must be considered here.). The autonomous network should be able to renegotiate the existing slice with attachment of alternative resources or temporary additional capacity, first at the NSS virtualization layer. Some access points located near the musical event require more throughput, and some VNFs may be migrated or added to other VMs. The NSS orchestrator attaches the virtual components. Then, the SDN controller and the lower layers check and physically allocate the attached slice. The slice is then physically modified. The databases are updated for the period of the event. Using two virtualization layers enables work in parallel and assures network elasticity. The virtual deployment is not limited by the network resource virtualization, i.e., the network slice can be created without every limitation of the resource's virtualization. For example, when a network slice including virtual NodeB is modified, these VNFs do not always exist in the required VM (closer to the event). They can be scaled up or created at the step of the network slice allocation. Open source software such as Docker and Kubernetes that support containers and containers management, cover this domain with micro-services (used in CORD and ONAP). Consequently, this separation into two levels of virtualization in the NaaS assures dynamicity of network slicing. Moreover, network slices stay isolated from each other.

C. *Functional proposal: sequence diagram and scenarios*

In this section, we use sequence diagrams to describe some scenarios of service provisioning to gain clear understanding of the autonomic aspect.

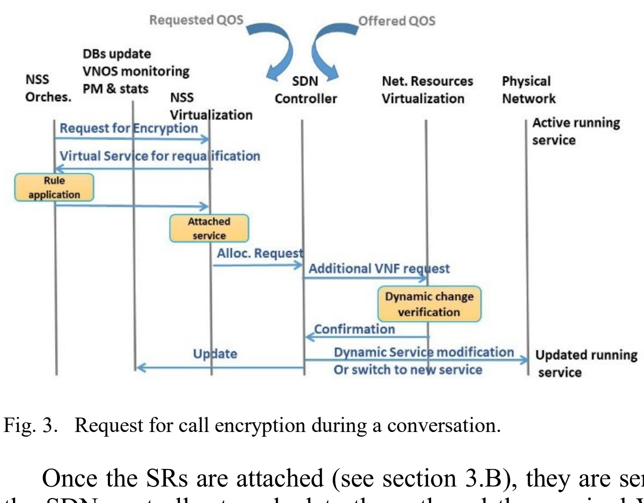

Fig. 3. Request for call encryption during a conversation.

Once the SRs are attached (see section 3.B), they are sent to the SDN controller to calculate the path and the required VNF locations according to the real-time and QoS constraints at the resource level. Then, the request is sent to network resources virtualization, where virtual nodes and vlinks are chosen. If the resources are available, confirmation is sent to the SDN controller, the service is placed and activated. Finally, the two databases and PM handler are updated with the new service, and virtual network OS (VNOS) monitoring [22] starts on this link. VNOS per running service once activated, allows real-time and autonomic service management.

In section 2.C, we present a user who needs to encrypt their voice call during conversation. A request is sent to modify the running service. This request is addressed to the NSS orchestrator (see fig. 3). It looks for two VNFs realizing encryptions in the resource manager, to allow this E2E call encryption. To this end, these two VNFs must be deployed near the end-users. The orchestrator tries to recompose the service at the NSS virtualization. They require more VM computing resources and introduce queuing. Together with the other VNFs of the call, this may overcome the general E2E QoS call constraints. A request to the NSS orchestrator can improve the E2E delay by decreasing, for instance, the number of hops for the conversation. Then, once the SR is attached, it is addressed to the general SDN controller that calculates its path, using the network resource virtualization. If these VNFs already exist in the related VM, the running service is updated or re-allocated and the service is switched transparently to the modified one. If these two VNFs do not exist in the needed VMs, one option is to install these VNFs in the required VMs (If enough CPU and memory). Else, some migration of other VNFs can be done before, if these VNFs can be located in another VM and thus free resources for the required VMs. Then, the databases are updated and the VNOS associated with this service restarts. In today networks, such a dynamic request is not handled: the call must be stopped, and if encrypted call service is available, the call service must be renewed. Despite the complexity of the architecture, we have illustrated how it can efficiently solve specific dynamic use cases and provide enhanced autonomy.

IV. CONCLUSION

Considering the growing complexity of networks and the explosive demand for services, there is a need for a zero-touch network. An architecture that overcomes limitations in the area of orchestration at the northbound interface with the OSS is required. Synergies between the SDOs and open sources would help this to be achieved. Although these groups have performed substantial work in the domain of standardization and adaptability between the control and data planes, these architectures lack some dynamicity. In this study, we proposed an enhanced architecture by introducing a new network virtual layer above the SDN controller to improve dynamicity. With this new layer in place, orchestration is then distributed among different layers. We clearly identified the role of orchestration and the resulting greater autonomy. We showed how this architecture can overcome some limitations such as simultaneously respecting the SLA and QoS constraints at different levels of the network. Using some examples, we illustrated how this architecture provides more dynamicity when a VNF must be migrated or added during a session. Following this, we work now on applicative simulations to demonstrate the validity of our proposition.

ACKNOWLEDGMENT

This research was funded by the Israel Innovations Authority under the Neptune generic research project. Neptune is the Israeli consortium for network programming.

REFERENCES

[1] A. Neal, "Study on New Services and Markets Technology Enablers" 3GPP – TR 22.891 rel. 14.2, Sep. 2016.

[2] V. Millnert, E. Bini, J. Eker, "Cost minimization of network services with buffer and end-to-end deadline constraints", ACM SIGBED Review, 14(4), pp. 39-45, Jan. 2018.

[3] M. Di Natale and J. A. Stankovic, "Dynamic end-to-end guarantees in distributed real time systems", in Proceedings of the 15-th IEEE Real-Time Systems Symposium, pp. 215–227, Dec. 1994.

[4] T. H. Tan, C. Manman, S. Jun, L. Yang, A. Étienne, X. Yinxing, and D. Jin Song, "Optimizing selection of competing services with probabilistic hierarchical refinement." In IEEE/ACM 38th International Conference on Software Engineering (ICSE), pp. 85-95., IEEE, 2016.

[5] A. Sheoran, P. Sharma, S. Fahmy, and V. Saxena, "Contain-ed: An nfv micro-service system for containing E2E latency", ACM SIGCOMM Computer Communication Review, 47(5), pp. 54-60, 2017.

[6] G. Saadon, Y. Haddad, and N. Simoni, "A survey of application orchestration and OSS in next-generation network management", Computer Standards & Interfaces, vol. 62, pp. 17-31, 2018.

[7] "SDN Architecture Overview", ONF Technical Reference TR-504, Nov. 2014.

[8] "SDN Architecture Issue 1.1", ONF Technical Reference TR-521, Feb. 2016.

[9] "Orchestration: A more holistic view", ONF Technical Reference TR-540, Jan. 2017.

[10] C. Rotsos, D. King, A. Farshad , J. Bird, L. Fawcett, N. Georgalas, M. Gunkel, K. Shiomoto, A. Wang, A. Mauthe, N. Race, and D. Hutchison, "Network service orchestration standardization: a technology survey", Computer Standards & Interfaces 54, pp. 203–215, 07 Feb. 2017.

[11] A. Mayer and S. Mansfield, "The Third Network: Lifecycle Service Orchestration Vision" MEF, white paper, Feb. 2015.

[12] "Lifecycle Service Orchestration (LSO): Reference Architecture and Framework", MEF-55, Mar. 2016.

[13] D. Bushaus, A. Turner, R. Rich, S. Wray, P. Davis, "Orchestration, Get ready for the platform revolution", ISBN: 978-1-945220-04-3, tmforum, Sep. 2016.

[14] "OSS/BSS futures – Architecture – Preparing the future mode of operation FMO of Hybrid services in digital market places", tmforum, release 15.5.1, Mar. 2016.

[15] D. Bushaus, S. Wray, and P. Davis, "Moving towards the operations center of the future" OpCF, ISBN: 978-1-939303-97-4, tmforum, Mar. 2016.

[16] T. Ben Meriem, T. Nakamura, S. Horiuchi, S. Fratini, D. Milham, and A. Pope, "A Dynamic Network Slices Management and Business Models", tmforum, rel. 17.0.1, Nov. 2017.

[17] "Network Functions Virtualization (NFV); Management and Orchestration", ETSI GS NFV-MAN 001 V1.1.1, Dec. 2014.

[18] R. Mijumbi, J. Serrat, J. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and Orchestration challenges in network function virtualization", IEEE Communications Magazine, 54(1), pp. 98-105, Jan. 2016.

[19] "Open Network Automation Platform (ONAP) Architecture" White Paper, Casablanca release, Nov. 2018.

[20] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, 690 Central office re-architected as a data center, IEEE Communications Magazine 54 (10) pp. 96–101, Oct. 2016.

[21] Q. Duan, N. Ansari, M. Toy, "Software-Defined Network Virtualization: An Architectural Framework for Integrating SDN and NFV for Service Provisioning in Future Networks," IEEE Network Magazine, vol. 30, no. 5, pp. 10-16, Sep. 2016.

[22] Q. Duan and S. Wang, "Network as a service for next generation internet", chapters 3 and 4, IET, 2017.

[23] E. Marilly, O. Martinot, H. Papini, and D. Goderis "SLA: a main challenge for Next Generation Networks" Alcatel IEEE, Aug. 2002.