

Baguette: Towards end-to-end service orchestration in heterogeneous networks

Charalampos Rotsos*, Arsham Farshad*, Nicholas Hart*, Alejandro Aguado†, Sarvesh Bidkar†, Kyriakos Sideris†, Daniel King*‡, Lyndon Fawcett*, Jamie Bird*, Andreas Mauthe*, Nicholas Race*, David Hutchison*

* School of Computing and Communication, † Department of Electrical and Electronic Engineering, ‡ British Telecom
Lancaster University Bristol University

Abstract—Network services are the key mechanism for operators to introduce intelligence and generate profit from their infrastructures. The growth of the number of network users and the stricter application network requirements have highlighted a number of challenges in orchestrating services using existing production management and configuration protocols and mechanisms. Recent networking paradigms like Software Defined Networking (SDN) and Network Function Virtualization (NFV), provide a set of novel control and management interfaces that enable unprecedented automation, flexibility and openness capabilities in operator infrastructure management. This paper presents *Baguette*, a novel and open service orchestration framework for operators. *Baguette* supports a wide range of network technologies, namely optical and wired Ethernet technologies, and allows service providers to automate the deployment and dynamic re-optimization of network services. We present the design of the orchestrator and elaborate on the integration of *Baguette* with existing low-level network and cloud management frameworks.

I. INTRODUCTION

Innovation in network operators enables unprecedented connectivity capabilities for enterprise and commercial end-users, radically transforming all aspects of the global economy, as well as everyday life. At the core of this innovation lies the concept of network services, which allow network operators to generate profit and introduce added-value in their infrastructures. The service portfolio of modern operators covers a wide range of functionalities, including dedicated connectivity, intelligent traffic management and traffic optimization services. Network service innovation has been extensive, but network operators currently face significant challenges to support the growing end-user and application connectivity requirements, while ensuring profitability for their business model. Network device numbers, traffic volumes and access speeds followed an exponential increase for the last decade, while continuously emerging network applications introduce stricter service delivery requirements. Unfortunately, the predominant network management and control protocols have not kept abreast, designed to fulfill radically revised functional requirements, they underwent through limited development in the recent years.

These limitation in production management and control standards force network operators to follow static and inflexible service deployment and management mechanisms. Due

to limitation in dynamic network resource control, delivery guarantees are met through link over-provision and service resilience is achieved through redundant path provision. Furthermore, service management is highly manual, relying on network manager intervention for device configuration, and follows a workflow-oriented approach, coordinated by an Operation Support Systems (OSS). This management model results to extensive OPEX increase (estimated at approximately 27% of total revenue [1]), and high service delivery times (on average dedicated connectivity services have a delivery time of one month [2]), as the size of the network and the number of services increase.

To address these challenges, academia, operators and vendors have engaged in an extensive exploration for new technological paradigms, architectures and protocols which improve management flexibility, scalability and resiliency for network services. Software Defined Networking (SDN) is a recent networking paradigm, introduced in late 2000, proposing clean separation between the control and the data plane of a network, in an effort to improve flexibility in the forwarding policy definition and enable logical control centralization. To enable these functional goals, SDN defines new control interfaces which converge control across a wide range of network technologies and enable timely reconfiguration of the forwarding and resource allocation policy of a device. Similarly, Network Function Virtualization (NFV) explores the applicability of commodity servers to implement network functions (NF), thus enabling data plane programmability and improving elasticity in NF deployment.

SDN, NFV and similar networking paradigms provide low-level primitives that allow functional evolution of a network. To exploit these novel capabilities, we argue in this paper for the introduction of a service orchestration layer in the management and control stack of the network infrastructure, responsible to transform high-level service requirements into concrete infrastructure configurations. To achieve this we present *Baguette*, a novel orchestration framework tailored to the requirements of network services. *Baguette* provides *coordination* and *resource provision* across a diverse set of network technologies (computational, storage, packet and optical resources) and *automation* in the deployment and re-

optimization of network services. Unlike existing function orchestration frameworks, Baguette is designed to support a service-oriented control abstraction.

Baguette design is motivated by popular networks services and aims to fulfill their functional requirement (§ II). We present the architecture of Baguette and elaborate on its ability to converge control between two popular communication technologies; wired Ethernet and optical (§ III). Finally, the paper discusses future directions in the design of service orchestrators (§ IV).

II. SERVICE ORCHESTRATION CHALLENGES

Network services are represented as directed network graphs, where nodes describe NFs and edges describe connectivity requirements between them. NFs are abstract network elements capable to perform specialized traffic manipulation operations at high rates and with low processing overheads. Their processing capabilities span from simple packet manipulation operations like NATing, firewalling and routing, to advanced flow-oriented processing operations, like WAN optimization and application proxying. Service graph nodes and edges are typically associated with service delivery guarantees that must be fulfilled by the operator during the lifecycle of a service. NF delivery guarantees include both low-level metrics, like bandwidth and per-packet latency bounds, and high-level application-oriented metrics, like guaranteed object delivery latency. Connectivity resource guarantees include a combination of bandwidth, jitter, latency and physical-layer technology requirements. A key challenge for the deployment of network services is NF ordering. Directionality in the service graph defines the order in which packets must traverse NFs.

A *service orchestrator* is a control module which manages the computational and network resources of an operator infrastructure and enables seamless deployment, management and configuration of a network service. To identify the requirements for the service orchestrator, this section elaborates on the model of two popular network service types (Figure 1). Motivated by these services, we define a set of requirements for future service orchestrators.

A. Content Delivery

Content delivery is one of the predominant traffic classes, based on volume, in the current Internet traffic mix [3]. The radical increase in the volume of HTTP traffic carrying static web objects has motivated the development of a new end-to-end service type, named Content Delivery Networks (CDN). CDN services are third-party services which assist Internet applications to improve performance and scalability by on-loading static content distribution.

CDN architectures commonly employ multi-layer architectures to achieve scalability and flexibility. As depicted in Figure 1a, CDN architectures contain predominantly two node types: cache and storage nodes. Cache nodes are distributed across the Internet and serve content to nearby end-users. Because served objects do not fit in the memory and disk

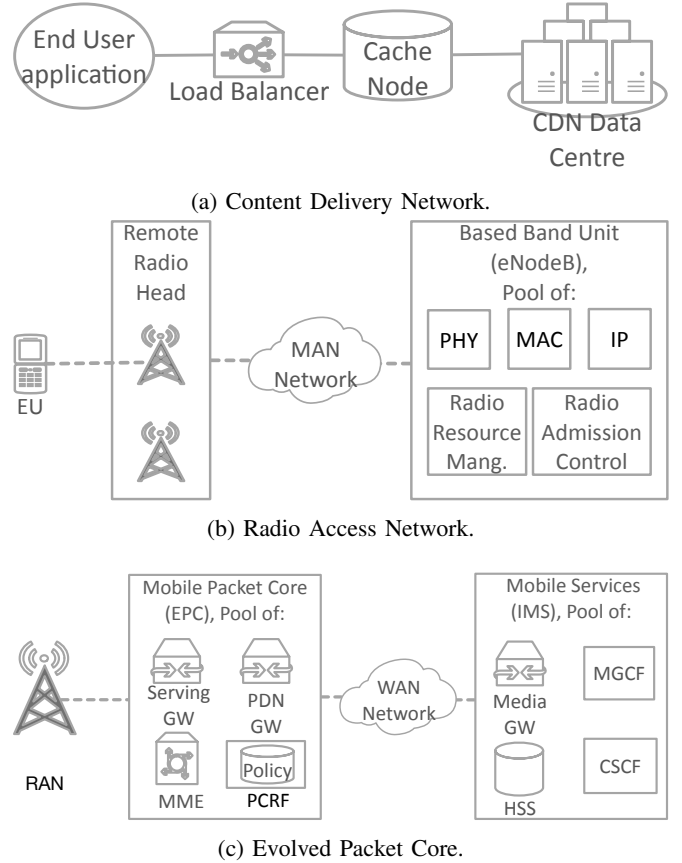


Fig. 1: Example service graphs

resources of each cache node, CDN services maintain a small number of well-provisioned content storage datacenters which host all available objects. On a user request, if an object is not available on the selected cache node, the cache node is responsible to fetch the object from the appropriate storage node in the CDN datacenter. In addition to the storage and cache nodes, CDN services maintain a management plane, responsible to monitor node load and object popularity and decide caching and client steering policy.

CDN services follows two primary approaches to provision connectivity with eyeball ASes, in order to ensure service reachability with low-latency and high-bandwidth. In the first approach, a CDN provider deploys dedicated links from its datacenters to popular IXP infrastructures, thus achieving high peering degree on the AS graph. Eyeball ISPs must provision a similar link to ensure that end-users can access CDN services [4]. An alternative approach, that has gained popularity in the recent years, relies on cache node deployment in the operator datacenters [5]. This way, the operator minimizes the volume of traffic traveling outside its infrastructure and reduce OPEX.

The operation of CDN services in the current Internet architecture is significantly impaired by the static and vertically-integrated nature of modern network infrastructures. On the

one hand, CDN service providers face significant challenges towards an efficient mechanism to redirect users to their nearest cache node. Furthermore, the limited elasticity of resource allocation mechanisms in operator infrastructures, limits the ability of CDN services to gracefully handle high fluctuation in user demands. On the other hand, operators face significant challenges to monetize CDN service connectivity and efficiently manage CDN traffic steering changes. Typically, when a bottleneck appears in the operator network, CDN services try to dynamically steer traffic requests to different cache nodes, radically changing the traffic matrix of the ISP. In response to such major traffic matrix changes, routing protocols try to adjust the forward policy to accommodate link load changes, and result in transient forwarding policy oscillation, connectivity degradation and packet loss.

B. Mobile Network Virtualization

Mobile networks have been widely adopted by end-users in the recent years, currently providing support for 50% of the global population [6]. Since the introduction of the 4G standards, mobile network operators have converged all of their basic network services under a common packet-oriented design with strong packet latency guarantees. The recent effort in the development of the new 5G standards, have introduced new stronger service delivery requirements; three orders of magnitude greater access speeds and device connectivity support, two orders of magnitude more bandwidth per user, sub-millisecond latency and lower energy consumption and OPEX [7].

Mobile networks follow a dual-layer hierarchical architecture, consisting of the *Radio Access Network (RAN)* and the *Evolved Packet Core (EPC)*. Since the introduction of the 3G standard, the mobile RAN is separated in two functional blocks: the Remote Radio Head (RRH), which is responsible for wireless signal reception, transmission, transformation and amplification, and the Base Band Unit (BBU), which manages the MAC layer between the base station, the connected end-devices and the neighboring cells (Figure 1b). Connectivity between the RRH and the BBU has high bandwidth and ultra-low latency requirements, thus the two components are typically co-located. This design choice increases significantly the RAN deployment and operational costs. The cooling requirements of a BBU increase significantly the cost and power consumption of a base station [8].

Recent trends in RAN design have proposed the separation of the two components and the deployment of BBUs in the central office of the operator; an architecture commonly termed as Cloud-RAN (C-RAN). C-RAN significantly reduces deployment and operational costs and improves elasticity and resilience of the RAN. In parallel, the centralization of multiple RRHs under the control of a single BBU improves resource utilization, cell handovers, and minimizes inter-cell interference. Currently a number of interfaces, architectures and testbeds provide the technological capabilities to run and test C-RAN systems [9], [10], while vendors currently provide production-ready virtualized BBU appliances [11]. In addition,

novel control abstractions have been proposed to converge the RAN control abstraction with the underlying transport technologies and enable flexible deployment strategies [12].

A challenge for C-RAN architectures is the multi-Gb bandwidth and sub-millisecond latency and jitter requirements for the connectivity between the RRH and the data-center [13]. These requirements significantly vary over time, reflecting fluctuating geographical location popularity. To improve the efficiency of the RAN front-haul connectivity, operators require dynamic resource management over a wide range of transport and packet networks. In addition, to improve elasticity and resiliency, operator require greater flexibility in BBU pool resource management. Addressing these challenges is a core goal for the 5G Public Private Partnership (5G PPP) standardization body [14].

Evolved Packet Core (EPC) is an architecture for the core network of mobile operators, introduced in the 4G standards, which allows convergence of voice and data traffic under a single IP-based network. The functional blocks contained in an EPC are presented in Figure 1c. The design of the architecture contains a large number of individual components, each having diverse bandwidth and latency requirements, while implementations tend to exhibit limited interoperability and openness.

Multiple studies have argued for the softwarization of the key EPC functional blocks and the introduction of programmability in the EPC network control [15]. SoftAir [16] is a software-defined architecture for next generation mobile networks using network and function virtualization paradigms for both EPC and RAN. Open5GCore [17] provides an open and programmable LTE protocol stack for EPC components. In parallel, network vendors provide carrier-grade IMS VNF products [18].

C. Service Orchestration Requirements

A key challenge for service innovation is the significant operational costs in infrastructure management. Effectively, production network technologies and protocols exhibit limited flexibility to support the required dynamicity of complex network services. Motivated by this observation, vendors and academia have engage in the development of innovative network and server products, based on new paradigms like SDN and NFV, which virtualize the lower layers of network infrastructures and enable programmability. To fully exploit these new network properties, operators require a high-level orchestration layer designed to fulfill network service requirements. We identify the following functional requirements for a service orchestrator:

Coordination: Operator infrastructures exhibit high heterogeneity, both in terms of the control abstractions, as well as in terms of protocols. Typically, the design of an operator network follows a three-layer hierarchical model, where each layer uses a different mix of network technologies. Each network technology has a unique low-level control abstraction, like the wavelength abstraction of optical technologies, which requires specialized device configuration in order to map path resource requirements into appropriate configurations.

Management heterogeneity is equally significant across NF products. NFs provide a wide range of network functionalities, spanning across all the layers of the network, and thus expose specialized management and configuration interfaces. To improve the flexibility of the underlying infrastructure, the orchestrator must unify the management and control interfaces of the different technologies under a common service-oriented abstraction.

Automation: Operators infrastructures containing devices from multiple vendors or network technologies operate multiple standalone control systems, for each vendor and network layer. These individual controllers do not provide any interoperability and service deployment relies on manual intervention by network managers to propagate the required configurations between them, typically orchestrated through complex workflows organized by an OSS system. Network programmability, based on the SDN paradigm, and NF virtualization, based on the NFV paradigm, provide low-level abstractions that can provide easy management and configuration across all layers of the network. A key goal for an orchestrator is to capitalize on this low-level interfaces and synthesize new service-oriented abstractions that minimize human interaction and provision service in the order of hours.

Resource and function virtualization: The limited flexibility of the predominant infrastructure management technologies, introduces topological, technological and resilience dependencies between network service deployment strategies and the underlying infrastructure [19]. As a result, network paths are over-provisioned to ensure delivery guarantees during peak operation, while idle back-up paths between end-points are established to ensure resiliency. Furthermore, because NFs are not ubiquitous in the network, the network policy is responsible to guarantee that the NF ordering is fulfilled. As the number of services and the size of the network increases, the resource fragmentation, generated by the existing model, escalates and increases the OPEX of the operator. The network orchestration is responsible to take advantage of the resource and forwarding control interfaces, currently available in SDN implementations to provide flexible and timely network configuration in order to meet service delivery guarantees.

III. BAGUETTE ARCHITECTURE

Baguette is a novel service-oriented orchestration framework. This section presents the design of the Baguette orchestrator and elaborates on the capabilities of the current implementation and its integration with existing control mechanisms in the operator infrastructure. Figure 2 presents the architecture of Baguette. Its functionality exploits the control capabilities of the SDN and NFV paradigms to converge control across network infrastructures, using off-the-self control frameworks. Its primary functionalities are to provide coordination between the network and function control systems and to enable flexible resource and forwarding control.

External applications and OSSes can use a RESTful API to deploy and monitor their network services over the orchestrated infrastructure. To elaborate on the functionality

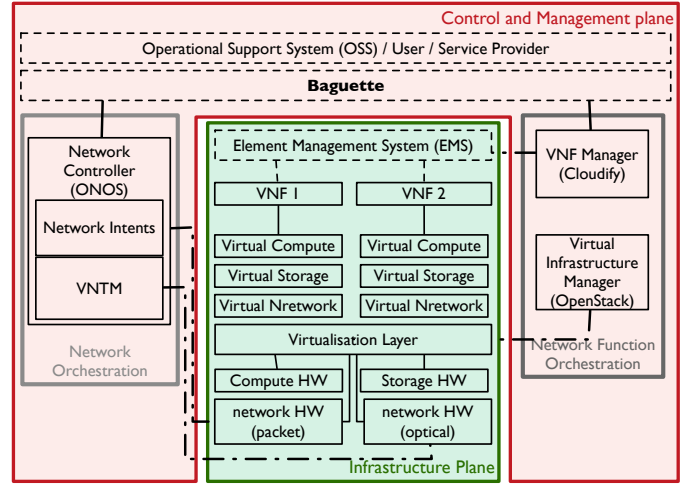


Fig. 2: An architectural model for end-to-end service orchestration.

of Baguette, Listing 1 presents a simplified CDN service (Figure 1a) definition. The definition consists of four elements: a *service name*, an *operational budget*, a list of *NFs*, and a list of *NF and network endpoint dependencies*. A NF specification describes the type of the function, along with configuration parameters and performance requirements. A service dependency specifies the connectivity requirements between NFs and network end-points, represented as IP address masks. Each service dependency can contain a set of service delivery requirements. The current Baguette instantiation considers latency and bandwidth parameters exclusively.

NFs can be implemented using a wide range of appliances. To simplify the specification of network services and to improve resource utilization, Baguette decouples the definition of a network service from the specific implementation details. Effectively, the service consumer describes in the service definition the type of NFs contained in its service chain. The orchestrator is responsible at run-time to optimize the mapping between the NF types and the NF implementations of each service chain, given the performance requirements and budget of the service providers and the available resource in the infrastructure. For example, a load balancing function can be realized using: (i) a proactive OpenFlow policy configured on a virtualized SDN switch, (ii) a virtual machine running software like nginx or HAproxy, or (iii) through a virtualized hardware load balancer [20], with each implementation providing different performance and functionality trade-offs. In Baguette, these different appliances are abstracted under the `lb` function type. The end-user of the orchestrator can use the requirements section of the NF template to specify its requirements with respect to the different trade-offs across relevant NF implementations.

Baguette architecture is presented in Figure 3. The system consists of three functional blocks: the *Service Orchestrator*, the *VNF Orchestrator* and the *Network Provisioner*. The Service Orchestrator is a focal control point for the operator

```

1  {
2  "name": "CDN",
3  "budget": 100,
4  "nfv": {
5      "cache": {
6          "type": "cache",
7          "config": {
8              "storage": "16GB",
9              "service": ["10.20.10.1"],
10         },
11         "requirements":
12         {
13             "requests": 4000
14             ....
15         },
16         "load-balancer": {
17             "type": "lb",
18             "requirements":
19             {
20                 "requests": 4000
21                 ....
22             }
23     },
24     "paths": [
25         {"src": "10.10.10.0/24", "dst": "lb",
26          "requirements": {"bw": ".5g"}[...]},
27         {"src": "lb", "dst": "10.10.10.0/24",
28          "requirements": {"bw": "1g"}},
29         {"src": "cache", "dst": "lb",
30          "requirements": {"bw": "1g"}},
31         {"src": "lb", "dst": "cache",
32          "requirements": {"bw": "1g"}},
33         {"src": "10.20.10.0/24", "dst": "lb",
34          "requirements": {"bw": "1g"}},
35         {"src": "lb", "dst": "10.20.10.0/24",
36          "requirements": {"bw": "1g"}},
37     ]
38 }

```

Listing 1: A CDN service chain specification using the service interface of Baguette.

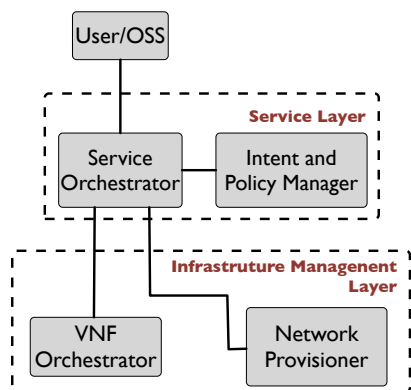


Fig. 3: Baguette functional block diagram.

infrastructure. The system is responsible to aggregate service requests from users and identify a near-optimal deployment plan. The module has complete visibility of available network and computational resources across the operator infrastructure and uses a service embedding algorithm to decide on the deployment strategy for every service request. The current Baguette implementation treats the embedding problem as a

bin packing problem and uses a suboptimal first-fit algorithm to assign path and resources to a service request. The problem of service embedding is a well studied problem, and we aim in future iterations of the implementation to explore better heuristics algorithms [21]. Once the service orchestrator has selected a deployment plan, the service deployment is delegated to the VNF orchestrator and the network provisioner, to configure the NFs and the paths of the service, respectively. The rest of this section discuss the design of Baguette for the control and configuration of the NFs and connectivity establishment.

A. Network Function Orchestration

The current Baguette implementation supports exclusively software NFs running over a virtualized server. NF control is separated in two layer: the virtual infrastructure layer and the VNF layer. The infrastructure layer controls resource virtualization across the servers of the infrastructure, while the VNF layer is responsible for the configuration and the control of NFs. To rapidly prototype NF orchestration in Baguette we use the OpenStack [22] and the Cloudify [23] frameworks, to support the respective control layer.

OpenStack is a popular open-source cloud management framework, providing a generic API for the deployment, configuration and control of VMs across a cloud infrastructure. It provides support for all popular server virtualization frameworks, like XenServer, Docker and KVM, and exposes a unified VM management abstraction. OpenStack follows a micro-service architecture and compartmentalizes the control and configuration tasks into standalone services. Baguette uses OpenStack to allocate and monitor computational and storage resources and to deploy NFs across the cloud infrastructure of the operator. OpenStack provides additionally the ability to control the network configuration of individual VMs and virtualized servers. Nonetheless, the design of the OpenStack network service (neutron) assumes that the network is a single broadcast domain, over which the cloud manager has direct connectivity between all VMs. We relax this requirement by assign the network configuration of the network edges (VM and virtualized server) to OpenStack and establish inter-server connectivity by appropriately manipulating network configuration through the Baguette network provisioner.

Cloudify is an open-source cloud orchestration framework, designed to automate the configuration, monitor and control of NFs. The framework provides out-of-the-box OpenStack integration to launch VMs and to configure local networking, storage and security. Cloudify supports the TOSCA modeling standard [24] and provides a YAML-based Domain Specific Language (DSL) to model interaction of the network orchestrator with NF. A service is described through the Blueprint abstraction, a TOSCA-based lifecycle description of the service and its individual NFs. Blueprint files describe the logical representation of a service graph, the components of the service and the integration between NFs. In addition, a Blueprint specifies script executions that occur during the different event of the NF lifecycle.

During a service request to the Baguette RESTful API, the VNF orchestrator is responsible to translate specific NF types into respective NF implementation representations in the TOSCA language. Baguette contains a set of `Blueprint` templates for the specification, configuration and monitor of different NFV instances and their integration. A high-level service specification is transformed by the VNF Orchestrator module into a `Blueprint` script, by stitching together individual NFV templates. The current Baguette implementation contains a small set of templates for stateless NFV appliances (load balancer, cache, switch) and we aim to enhance our codebase with more NFV templates in the near future. The resulting `Blueprint` is submitted to the Cloudify instance for deployment.

B. Network Orchestration

Cloud management infrastructures have been proposed as an effective framework for the control of software NF instances. Nonetheless, a key challenge for the migration of such frameworks in an operator environment, is the significantly different network design. Cloud infrastructures typically contain large layer-2 domains using only wired Ethernet and NF connectivity is implemented using tunneling mechanisms on the edges of the network. Operator networks contain a wider range of network technologies, its infrastructure are distribute across multiple locations, while routing and resource control is more complex and cannot simply rely on edge network configuration.

In order to improve the flexibility in the control of the underlying network to meet the growing connectivity demands of network services, Baguette uses the ONOS network controller. The network provisioner module uses the ONOS RESTful API to discover the network topology and to configure network paths between NFs. Path establishment uses the intent control abstraction. Intents is a novel control abstraction, developed as a core module in ONOS, which provides a mechanism to describe the connectivity requirements between network end-points. Control application express their connectivity requirements as intents and the controller is responsible to define a network policy that fulfills these requirements and appropriately reconfigures the forwarding policy during link failures and bottlenecks.

The ONOS intent abstraction is designed for wired Ethernet. In order to support optical technologies in Baguette, the network provisioner uses a Virtual Network Topology Manager (VNTM). VNTM is a service that stores multilayer information for heterogeneous network and supports sets up or tears down of lower-layer Label Switched Paths (LSPs) [25]. When the network provisioner cannot set-up a path in the wired Ethernet domain, it fetches topological information from the optical domain of the network and explores if network connectivity can be established through the establishment of an optical path. Upon the optical path establishment, the ONOS controller will be able to detect the path also in the packet domain and we be able to compute and establish network paths through the intent abstraction.

A. Network Function Models

NFs provide a wide range of functional capabilities, which spans from simple packet manipulation to complex flow termination. Due to the high heterogeneity between NFs, control interfaces vary significantly. Interface heterogeneity impairs significantly the flexibility and automation of the orchestrator. Typically, NF management relies on a simple lifecycle model which triggers bash script executions which configure or monitor an active NFV instance. This model increases significantly the complexity to introduce new functions in the network of an operator. The network manager is responsible to analyze and identify the configuration mechanisms supported by a new NF and to implement and test from scratch the required scripts required for integration with the data model of the service orchestrator. Furthermore, the data model of a new NF may contain information which is incompatible or absent in the data model of the orchestrator, thus requiring data transformation and even redefinition of the orchestration logic. Effectively, we require new data models which converge the configuration and monitoring capabilities available in NFs with similar functional capabilities. The standardization of control interfaces between NFs can also enable seamless cooperation between NFs and joint configuration. For example, a load balancing NFV can receive performance measurements from individual replicas and dynamically adapt its hashing function.

Efforts towards service modeling are fairly recent and their outcomes are still limited. The IETF NETCONF Data Modeling Language (NETMOD) WG provides a rich portfolio of model specifications, developed using the YANG [26] data modeling language. The respective models can be classified in two broad categories: network element models and network service models [27]. Relevant to network service modeling are the latter models, but the scope of these models remains limited and primarily focuses on connectivity services. Relevant efforts in cloud computing have delivered frameworks, like Ansible [28] and Chef [29], which simplify the deployment of web service for large scale systems using configuration template. These systems provide cookbooks containing deployment recipes which abstract and automate web service and VM configuration. These approaches should be revisited and adapted in the context of NF management and configuration practices.

B. Network Heterogeneity

The current release of Baguette provides support for two network technologies: wired Ethernet and optical. These technologies vary significantly in their forwarding control abstraction, supporting respectively a packet-oriented and path-oriented abstraction. Nonetheless, their medium access mechanism is coordinated and, thus, resource control can be deterministic. In optical technologies resource control can be translated into appropriate spatial, time and frequency configurations, while performance degradation factors, like optical dispersion, are static. Similarly, wired Ethernet networks

resource control is enforced through the packet scheduling algorithm in network devices, using packet queues.

Nonetheless, network infrastructures contain a wide range of network technologies with shared medium and non-deterministic resource control, like wireless and visible light communication technologies. Resource control for such technologies is subject to the end-user mobility, while access patterns and signal interference can affect network performance. In the wireless domain, the research community has proposed a wide range of specialized control frameworks which virtualize access points and dynamically measure and reconfigure MAC layer parameters [30], [31]. Adding Baguette support for such technologies cannot rely on simple translation of network service requirements to technology specific configurations. Service orchestration requires the development of highly reactive and specialized control system for a subset of network technologies, which control the physical layer properties of the shared medium and guarantee service delivery goals.

C. Monitoring

The growing demand for dynamic resource, functions and connectivity provision in an orchestrated infrastructure can increase network incidents and unregulated network changes. Network orchestrator success depends on its ability to measure the network performance and assess network quality using a very small set of metrics and to provide network diagnosis and root cause analysis during service disruptions. In parallel, the orchestrator must support network resource scheduling which can adapt to near real-time service demands (in-operation) [32].

Nonetheless, to investigate network problems or identify the severity of major network events or interruptions, a network health index or network key performance index (KPI) or key quality index (KQI) is critical. Generating the KPI or KQI would require data collection from various data sources using a set of automated communication processes and transmit them to one or more receiving equipment. This process is known as network telemetry. The data collected from data sources include network performance data, network logging data, network warning and defects data, network statistics and state data, and network resource operation data (e.g., operations on RIBs and FIBs). The process and ability to normalize the data to derive several end-to-end network composite metrics that reflect the network performance and quality from several different perspectives, e.g., network diagnosis, network performance, network QoS, network security. These end-to-end network composite metrics can then be used for in-operation planning.

V. CONCLUSION

This paper elaborated on the challenge in the deployment of end-to-end services in modern network operator infrastructures. Motivated by the analysis of a set of popular service chains, we have identified a set of core functional requirements for the development of future service orchestration

mechanisms. Furthermore, we presented the architecture of the Baguette service orchestration and elaborated on the design of a straw-man implementation. The design of the Baguette orchestrator is still in its early stages and we plan to open source the first stable release of the system.

ACKNOWLEDGMENTS

The authors are grateful to the UK Engineering and Physical Sciences Research Council (EPSRC) for funding the TOUCAN project (EP/L020009/1), which supported much of the work presented in this paper; and to the EPSRC and BT for funding Daniel Kings Industrial CASE PhD research.

REFERENCES

- [1] T. Nolle, "Automated OSS and BSS models needed for new network operations," <http://searchtelecom.techtarget.com/tip/Automated-OSS-and-BSS-models-needed-for-new-network-operations>.
- [2] British Telecom, "BTnet: Market Leading Leased Line Internet," http://business.bt.com/assets/pdf/broadband-and-internet/datasheet/BTnet_target_availability.pdf, Aug. 2015.
- [3] Cisco, "Cisco visual networking index: Forecast and methodology, 2014-2019 white paper."
- [4] "Netflix peering locations," <https://openconnect.netflix.com/en/peering-locations/>, 2006.
- [5] "Netflix open connect," <https://openconnect.netflix.com/>, 2016.
- [6] ITU, "ICT facts and figures: the world in 2015," <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf>, May 2015.
- [7] 5GPPP, "5G Vision: The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services," <https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf>, 2015.
- [8] China Mobile, "C-RAN: The Road Towards Green RAN," 2011.
- [9] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo, "CloudIQ: A Framework for Processing Base Stations in a Data Center," in *Mobicom*. ACM, 2012.
- [10] N. Nikaiein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, Oct. 2014.
- [11] Alcatel-Lucent, "vRAN," <https://www.alcatel-lucent.com/solutions/vran>.
- [12] R. Riggio, K. Gomez, L. Goratti, R. Fedrizzi, and T. Rasheed, "V-cell: Going beyond the cell abstraction in 5g mobile networks," in *IEEE NOMS*, May 2014.
- [13] "Common Public Radio Interface (CPRI); Interface Specification V6.0," 2013.
- [14] 5G-PPP, "The 5G Public Private Partnership (5G PPP)," <https://5g-ppp.eu>.
- [15] ETSI, "Network functions virtualisation (nfv); use cases," *VI*, vol. 1, 2013.
- [16] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: A software defined networking architecture for 5G wireless systems," *Computer Networks*, vol. 85, jul 2015.
- [17] "Open5GCore: The next mobile core network testbed platform," <http://www.open5gcore.org>.
- [18] Alcatel-Lucent, "Rapport cloud communications," <https://www.alcatel-lucent.com/solutions/cloud-communications>.
- [19] P. Quinn and T. Nadeau, "Rfc 7498: Problem statement for service function chaining," 2015.
- [20] "Barracuda Load Balancer ADC," <https://www.barracuda.com/products/loadbalancer>.
- [21] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 1888–1906, Fourth 2013.
- [22] "Openstack," <https://www.openstack.org/>.
- [23] "Cloudify," <http://getcloudify.org/>.
- [24] OASIS TOSCA, "TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0," <http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/csd02/tosca-nfv-v1.0-csd02.pdf>, 2015.

- [25] A. Aguado, M. Davis, S. Peng, M. V. Ivaréz, V. Lpez, T. Szyrkowiec, A. Autenrieth, R. Vilalta, A. Mayoral, R. Muoz, R. Casellas, R. Martnez, N. Yoshikane, T. Tsuritani, R. Nejabati, and D. Simeonidou, "Dynamic Virtual Network Reconfiguration Over SDN Orchestrated Multitechnology Optical Transport Domains," *Journal of Lightwave Technology*, vol. 34, no. 8, pp. 1933–1938, April 2016.
- [26] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," RFC 6020, Oct. 2015. [Online]. Available: <https://rfc-editor.org/rfc/rfc6020.txt>
- [27] B. Claise and C. Moberg, "YANG Model Classification," Internet Draft, RFC, Apr. 2016.
- [28] "Ansible," <https://www.ansible.com>.
- [29] "Chef," <https://www.chef.io>.
- [30] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards Programmable Enterprise WLANS with Odin," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 115–120. [Online]. Available: <http://doi.acm.org/10.1145/2342441.2342465>
- [31] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC: An OpenFlow based architecture for 802.11 MAC layer processing in the cloud," in *2012 IEEE Globecom Workshops*, Dec 2012, pp. 186–191.
- [32] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, and V. Lopez, "In-operation network planning," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 52–60, January 2014.