Network-as-a-Service in Software-Defined Networks for End-to-End QoS Provisioning

Qiang Duan
Information Sciences & Technology Department
Pennsylvania State University Abington College
Abington, PA 19001
Email: qduan@psu.edu

Abstract—End-to-end Quality of Service (QoS) provisioning for supporting diverse application requirements is a challenging problem in the Internet. Network-as-a-Service (NaaS) in the Software-Defined Networking (SDN) paradigm offers a promising approach to addressing this challenge. In this paper, the author first presents a framework for applying NaaS in SDN that enables network service orchestration for supporting inter-domain end-to-end QoS. Then a high-level abstraction model for network service capabilities is proposed and a technique for determining required bandwidth in network services to achieve QoS guarantee is developed. Network calculus is employed in the proposed modeling and analysis, which makes the developed techniques general and applicable to networking systems consisting of heterogeneous autonomous domains.

I. INTRODUCTION

The Internet, as a large scale networking system consisting of heterogeneous autonomous systems, faces some special challenges to support various computing applications. End-to-end Quality of Service (QoS) provisioning across multiple network domains is one the challenges. The network architecture of current Internet lacks the flexibility for rapid development and deployment of network services for supporting the wide spectrum of application requirements. Inter-domain QoS provisioning in the current Internet, which typically requires a global agreement on QoS metrics and policies across autonomous systems, is particularly difficult to achieve.

Software-Defined Networking (SDN) is one of the latest revolutions in the networking field that has attracted interest from both industry and academia. SDN architecture decouples the network control and forwarding functions and enables the network control to become directly programmable and the underlying infrastructure to be abstracted for applications [1]. Key features of SDN include separated control and data planes, logically centralized network controller, programmability of the control plane, and standard application programming interfaces (APIs). SDN is expected to have a significant impact on future networking through enabling an open programmable network platform that provides great flexibility for supporting various applications. Separation of control and data planes and centralized controller in SDN also offer a promising approach to facilitating inter-domain end-to-end QoS provisioning in the future Internet.

In the past few years exciting progress has been made in research and development of SDN technologies, including network operating systems such as NOX [2], centralized network controllers and hypervisor like Floodlight [3] and Flowvisor [4], and protocols and interface between data and control planes (southbound interface) such OpenFlow [5]. However, research on SDN is still at an infant stage and there are many issues that must be further investigated before this new networking paradigm can be widely deployed, especially in carrier networks [6]. Carrier-grade end-to-end QoS provisioning in SDN, which involves inter-domain collaboration (east-westbound interfaces), is an important problem that has not been fully studied.

Compared to data center networks, the public Internet must address some special challenges for end-to-end QoS provisioning. Inside a Cloud data center, the applications, Cloud controller, network controller, and data forward platform all belong to the same administration domain; therefore knowledge of underlying network infrastructure, such as network topology and link states, can be made available to upper layer controllers easily. However in the public Internet, service users (computing applications) and network service providers belong to different domains; therefore detailed information of network internal states is not visible to applications. In addition, end-to-end communication paths in the Internet often traverse multiple autonomous systems operated by different carriers. Such a heterogeneous networking scenario requires higher-level abstraction for information exchange between service users and providers and loose-coupling interactions among the autonomous systems involved in end-to-end service provisioning.

In this paper, application of the service-orientation principle in SDN is proposed to address the challenging problem of end-to-end QoS provisioning. The Service-Oriented Architecture (SOA) [7] offers an effective mechanism to enable flexible interactions among autonomous systems to meet diverse service requirements. SOA has been widely adopted in various distributed computing areas including Cloud computing for offering elastic on-demand services. Application of the SOA principle in networking leads to the Network-as-a-Service (NaaS) paradigm, which makes the future carrier networks look more like Clouds. In the NaaS paradigm networking resources, such as transmission bandwidth and control functions,
are utilized by users on-demand as services through a standard interface, much like computational resources are utilized as services in Cloud computing.

NaaS enables abstraction of autonomous networking systems into network services, which can be composed into composite network services by a network orchestration module to offer end-to-end services. NaaS supports a high-level abstraction of network infrastructure in form of services and allows flexible collaboration among autonomous systems via loose-coupling interactions between the services. Therefore, NaaS may greatly facilitate end-to-end QoS provision in the future Internet.

Toward realizing the NaaS paradigm in SDN for supporting end-to-end QoS, this paper first presents a framework of integrating NaaS and SDN that allows abstraction and orchestration of network services in SDN. Then some key technologies for realizing such a framework are discussed. Specifically a model for high-level abstraction of network service capabilities is proposed and the model is also applied to network service composition. Based on such an abstract model, the author develops a technique for determining the required amount of resources that must be acquired in network services to achieve a given end-to-end QoS requirement. Network calculus theory is employed in the modeling and analysis, which makes the developed model and technique general and applicable to various networking scenarios consisting of heterogeneous autonomous systems.

The rest of the paper is organized as follows. The proposed framework of NaaS in SDN is presented in Section II. The high-level abstraction model for network service capabilities and its application to composite network services are given in Section III. Bandwidth allocation in network services for achieving end-to-end QoS guarantee is analyzed in Section IV. Numerical examples are provided in Section V to illustrate application of the developed technique. Section VI gives conclusion remark and discusses possible future work.

II. NETWORK-AS-A-SERVICE IN SOFTWARE-DEFINED NETWORKING

The SOA can be described as an architecture within which all functions are defined as independent services with invokable interfaces that can be called in defined sequences to form business processes [7]. A service in SOA is a module that is self-contained (i.e., the service maintains its own states) and platform-independent (i.e., the interface to the service is independent with its implementation platform). Services can be described, published, located, orchestrated, and programmed through standard interfaces and messaging protocols. A key feature of SOA is “loose-coupling” interaction among heterogeneous systems, which allows entities to collaborate with each other while keep themselves independent. This feature makes SOA very effective architecture for coordinating heterogeneous systems to support various application requirements.

Application of the service-orientation principle in networking provides a promising approach to addressing some challenges in the future Internet. Such a service-oriented networking paradigm is referred to as Network-as-a-Service (NaaS), in which networking resources are abstracted and utilized in form of network services. In principle, a network service may represent any type of networking component at different levels, including a domain consisting of a set of networks, a single physical or virtual network, or an individual network node. Multiple network services can be combined into one composite inter-network service through a service composition mechanism.

NaaS can be applied in the SDN paradigm to complement and enhance SDN. A system framework for integrating NaaS in SDN for inter-domain service provisioning is given in Fig. 1. In this framework, SDN data plane comprises forwarding infrastructure that may belong to various autonomous network carriers. In the control plane, autonomous network domains may adopt different SDN controllers, for example one domain implements its controller following the Floodlight specification while the other domain may use a Cisco OnePK controller. The control plane has a network service orchestration module on top of the set of SDN controllers. Networking functionality of each domain is abstracted as a network service through a NaaS interface between the SDN controller of the domain and the orchestration module. The orchestration module takes requests from upper layer applications and coordinates utilization of the available network services offered by underlying network domains for providing end-to-end services to applications. For requests that need inter-domain communications, the orchestration module composes network service components, which are abstractions of different network domains, into composite network services. Resource allocation is an important aspect in network service composition in order to meet QoS requirements. The orchestration module must determine the required amount of bandwidth in each involved network service and acquires the bandwidth through the NaaS interfaces with different service components in order to meet application performance requirements.

Fig. 1. Integration of NaaS in SDN for end-to-end QoS provisioning
The proposed integration framework combines advantages of NaaS and SDN. NaaS provides a high-level abstraction of autonomous networking systems and enables flexible interaction among them; thus allowing dynamic and scalable network service management in SDN. On the other hand, the separated control/data planes and centralized programmable controllers in SDN greatly facilitate realization of the NaaS paradigm, which allows a homogeneous control mechanism over the heterogeneous underlying network infrastructure. Therefore, integration of NaaS and SDN offers a promising approach to end-to-end QoS provisioning in the future Internet.

Some key technologies are needed for realizing integration of NaaS in SDN. The rest of the paper particularly investigates high-level abstraction model of network service capabilities and bandwidth allocation for composite network services, which are two important technologies for supporting end-to-end QoS provisioning.

III. An Abstraction Model for Network Service Capability

NaaS in SDN needs a high-level abstraction model for network service capabilities. Such a model should be general and flexible so that it is not only applicable to heterogeneous networking systems but also can be composed to specify service capabilities of end-to-end network services.

In general network service capability includes two aspects at a high-level: the connectivity offered by the network service, which can be described by enumerating the pairs of sources and destinations between which the network transports data; and the capacity of data transportation between each pair of source-destination. Therefore, for a given network service with \( m \) ingress points and \( n \) egress points, a high-level abstraction of its service capability can be specified by a matrix

\[
\mathbf{C} = \begin{pmatrix}
  c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\
  c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  c_{m,1} & c_{m,2} & \cdots & c_{m,n}
\end{pmatrix}
\]

where \( c_{i,j} \) is defined as

\[
c_{i,j} = \begin{cases}
  0 & \text{no connection from } i \text{ to } j \\
  P_{i,j} & \text{the service provides connection from } i \text{ to } j
\end{cases}
\]

and \( P_{i,j} \) is called the capacity profile for the connection from \( i \) to \( j \). Therefore, each non-zero element \( c_{i,j} \) of the matrix shows that the network offers data transport service from ingress \( i \) to egress \( j \), and the transport capacity of such a connection is described by the profile \( P_{i,j} \). The service curve concept from network calculus theory [8] is employed here to define a general capacity profile. The service curve in network calculus is defined as follows.

Let \( R(t) \) and \( R^*(t) \) respectively be the accumulated amount of traffic of a flow that arrives at and departs from a system by time \( t \). Given a non-negative, non-decreasing function, \( S(\cdot) \), where \( S(0) = 0 \), we say that the system guarantees a service curve \( S(\cdot) \) for the flow, if for any \( t \geq 0 \) in the busy period of the system,

\[
R^*(t) \geq R(t) \otimes S(t)
\]

where \( \otimes \) denotes the min-plus convolution operation defined in network calculus as

\[
h(t) \otimes x(t) = \inf_{s \geq 0} \{ h(t - s) + x(s) \}.
\]

A service curve model for typical networking system is the Latency-Rate (LR) server [9], which guarantees each flow a service curve \( \beta_{r,\theta}(t) = r(t - \theta) \), where \( \theta \) and \( r \) are respectively called the latency and service rate provided by the server to the flow.

In the proposed abstraction model, the service curve guaranteed by the network service for data transport from \( i \) to \( j \) is used as the capacity profile \( P_{i,j} \). Since a service curve is a general data structure that is independent with network implementations, it is flexible enough to describe service capabilities of various networking systems. In a network where a connection from ingress \( i \) to egress \( j \) can be modeled by a LR service curve \( r_{i,j}(t - \theta_{i,j}) \), the matrix element \( c_{i,j} \) can be represented by a data structure with two parameters \( [r_{i,j}, \theta_{i,j}] \).

An end-to-end Internet connection traverses multiple networking systems, each of which can be abstracted as a network service. The service curve-based capability model supports network service composition. Known from network calculus, the service curve \( S_i(t) \) guaranteed by a series of tandem servers \( S_i \), \( i = 1, 2, \cdots, n \), which respectively guarantees the service curves \( S_i(t), i = 1, 2, \cdots, n \) to a flow, can be obtained through the convolution of all the service curves; that is,

\[
S_n(t) = S_1(t) \otimes S_2(t) \cdots \otimes S_n(t).
\]

Therefore, the capacity profile of an end-to-end connection provided by a composite service can be obtained from the convolution of the profiles of all links offered by individual network services involved in end-to-end data delivery. Fig. 2 shows the NaaS-based abstraction of end-to-end service provisioning and composition of service capacity profiles to an end-to-end profile.

Since typical networking systems can be modeled as LR servers, composition of LR capacity profiles is particularly interesting to us. Suppose each network service \( S_i \), \( i = 1, 2, \cdots, n \), guarantees a service curve \( \beta_{r_i,\theta_i}(t) = r_i(t - \theta_i) \), it can be proved that the composite service curves is

\[
\beta_{r,\theta}(t) = \beta_{r_1,\theta_1}(t) \otimes \cdots \otimes \beta_{r_n,\theta_n}(t)
\]
where \( r = \min\{r_1, r_2, \ldots, r_n\} \) and \( \theta_S = \sum_{i=1}^{n} \theta_i \). Equation (4) implies that if the service capacity of each link on an end-to-end network route can be described by a latency-rate profile, then the end-to-end connection can also be modeled as a latency-rate capacity profile, whose latency parameter is equal to the summation of all link latency parameters and the service rate parameter is limited by the link with the least service rate.

IV. BANDWIDTH ALLOCATION IN NETWORK SERVICES FOR END-TO-END QOS PROVISIONING

A key to end-to-end QoS provisioning is allocation of sufficient amount of bandwidth in network services. In the proposed framework of NaaS in SDN, the network orchestration module is responsible for determining the required amount of bandwidth based on the QoS requirement specified by an application. Then the orchestration module communicates with network services through the NaaS interface for allocating the required bandwidth. The abstraction model and capacity profile proposed in last section can be employed by the orchestration module for determining bandwidth allocation requirement.

Computing applications with different traffic loads will require different amounts of bandwidth for achieving a certain performance level. Therefore, a method for describing traffic load is needed for bandwidth allocation. Due to the diversity of computing applications supported by future networks and the loose-coupling interaction between applications and network service providers, traffic load specification also requires high-level abstraction. The arrival curve concept in network calculus is employed here to develop a general abstract load profile that is applicable to various applications.

Let \( R(t) \) denote the accumulated amount of traffic generated from an application by time \( t \). Given a non-decreasing, non-negative function, \( A(\cdot) \), the application is said to have an arrival curve \( A(\cdot) \) if

\[
R(t) - R(s) \leq A(t - s) \quad \forall 0 < s < t.
\]

Currently most QoS-capable networks apply traffic regulation mechanisms at network boundaries to shape arrival traffic from applications. The traffic regulators most commonly used in practice are leaky buckets. A traffic flow constrained by a leaky bucket has an arrival curve

\[
A(t) = \min\{pt, \sigma + \rho t\},
\]

where \( p, \rho, \) and \( \sigma \) are called respectively the peak rate, the sustained rate, and the burst size of the flow.

For an application that generates a traffic flow with an arrival curve \( A(t) \) and has a maximum delay requirement \( D \), the required bandwidth at a network service for achieving the delay upper bound \( D \) can be determined by the effective bandwidth of this flow, denoted as \( r_D(A) \). Following network calculus theory, effective bandwidth of such a flow can be obtained as

\[
r_D(A) = \sup_{\epsilon \geq 0} \left\{ \frac{A(s)}{s + D} \right\}
\]

Considering a traffic flow constrained by a leaky-bucket arrival curve with parameters \( (p, \rho, \sigma) \), suppose that the capacity profile of the end-to-end route provided by a composite network service to this flow is a LR profile with parameters \( (r_e, \theta_S) \), then the effective bandwidth that must be allocated in the network service for this flow in order to achieve a required end-to-end delay upper bound \( D_{req} \) can be determined as

\[
r_{D_{req}} = \begin{cases}
\rho & D_{req} \geq D_{max} \\
\frac{\rho D_{req}}{(D_{req} - \theta_S)(p - \rho) + \sigma} & D_{min} < D_{req} < D_{max} \\
\sigma & D_{req} \leq D_{min}
\end{cases}
\]

where \( D_{max} = \theta_S + \sigma/\rho \) and \( D_{min} = \theta_S - \sigma/(p - \rho) \).

Equation (7) gives a technique for the network service orchestration module to determine the required amount of bandwidth for QoS provisioning. Please notice that here the bandwidth allocation is performed by the orchestration module with an end-to-end service vision; that is, the required bandwidth \( r_{D_{req}} \) needs to be offered by the composite network service, which in turn requires acquisition of bandwidth \( r_{D_{req}} \) from each individual network service (i.e., each network domain) involved in end-to-end service provisioning. The bandwidth acquisition can be realized via the NaaS interface between the orchestration module and network controllers, just like acquisition of computing capacity from a Cloud data center through the Infrastructure-as-a-Service (IaaS) interface.

V. NUMERICAL EXAMPLES

Suppose a communication path traverses three networking systems abstracted as network services, and the capacities provided by these network services to the end-to-end path are modeled by three LR profiles. We consider two applications \( A_1 \) and \( A_2 \) with different traffic loads and delay requirements that may utilize the end-to-end path. Denotes the traffic flows generated by the two applications as \( f_1 \) and \( f_2 \). The traffic parameters for \( f_1 \) and \( f_2 \) are respectively \( (p_1 = 5.3\text{Mb/s}, \rho_1 = 1.5\text{Mb/s}, \sigma_1 = 140\text{kbits}) \) and \( (p_2 = 3.2\text{Mb/s}, \rho_2 = 1.1\text{Mb/s}, \sigma_2 = 300\text{kbits}) \) [10].

![Fig. 3. Required amount of bandwidth for end-to-end delay guarantee.](image-url)

The amounts of bandwidth that must be provided by the end-to-end network service to guarantee a set of maximum delay requirements are determined and plotted in Fig. 3, where bandwidth for \( f_1 \) and \( f_2 \) are denoted as \( r_{e1}^1 \) and \( r_{e2}^2 \).
respectively. From this figure we can see that the required amount of bandwidth for both flows increase when the delay requirement decreases. This shows that more bandwidth must be acquired by the network orchestration module from each network service in order to provide a tighter end-to-end delay guarantee.

Fig. 3 also shows that different amounts of bandwidth are required by the two flows for achieving the same level of delay performance. This means that bandwidth allocation is impacted by application traffic characteristics as well as the delay requirement. Comparison between the two bandwidth curves shows that \( r^1_d \) drops faster than \( r^2_d \) does with increasing delay requirement. This implies that applications with different traffic loads require different amounts of bandwidth increment to achieve the same extent of improvement in delay performance.

NaaS enables network service abstraction of multiple networking domains to be composed into one composite network service for end-to-end service provisioning. Network service composition allows resource allocation across multiple domains with an end-to-end service vision; therefore the network orchestration module only needs to determine the effective bandwidth for the composite service then acquires the same amount of bandwidth from each network service components involved in the end-to-end service. Without such a network composition mechanism, each network domain has to determine and allocate the required bandwidth locally. In order to compare bandwidth utilization of these two allocation schemes in this networking case, the required bandwidth in each domain is also determined separately, assuming the end-to-end delay upper bound is equally divided into the three domains. The obtained results are plotted in Fig 4, where the domain-based bandwidth allocation for \( f_1 \) and \( f_2 \) are respectively denoted as \( r^1_d \) and \( r^2_d \).

Fig. 4 shows that the required bandwidth determined by both schemes decreases with increasing end-to-end delay requirement. That is, more bandwidth is required by both schemes to offer a tighter delay guarantee. Another important observation obtained from this figure is that for achieving the same level of delay performance, the allocation scheme for composite network service always requires less amount of bandwidth than what the domain-based allocation scheme requires. This indicates that in this example the end-to-end resource allocation enabled by NaaS-based network service composition achieves higher bandwidth utilization compared with the domain-based allocation scheme of the conventional inter-domain QoS mechanism.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, the author presents a framework for applying the NaaS paradigm in SDN, which enables network service orchestration for supporting inter-domain end-to-end QoS. A high-level abstraction model for network service capabilities is proposed and applied to composite network services for end-to-end service provisioning. Based on this model, a technique for determining the required amount of bandwidth in network services to achieve QoS guarantee is developed. Numerical study of a networking case is provided to illustrate application of the developed model and analysis. Analytical and numerical results show that NaaS-based network service orchestration supports end-to-end QoS with flexible service management and higher bandwidth utilization compared to traditional inter-domain QoS mechanism.

The author believes that integration of NaaS and SDN opens an interesting research direction. The progress toward end-to-end QoS provisioning in SDN reported in this paper could be extended in various aspects. As future work, the author plans to further investigate network service composition mechanisms that fully utilize networking resources while providing QoS guarantees. Efficient and flexible NaaS-based SDN northbound and east-westbound interfaces are also interesting topics for future research.

REFERENCES