Application Driven Network: providing On-Demand Services for Applications

Yi Wang†, Dong Lin†, Changtai Li†, Junping Zhang‡,
Peng Liu‡, Chengchen Hu§, Gong Zhang†
†Huawei Future Network Theory Lab, Hong Kong
‡Huawei Wireless Technology Lab, China
§Xi'an Jiaotong University, China

ABSTRACT

Application Driven Network(ADN) is a new paradigm that provides on-demand differentiated services for applications. A physical network in ADN is sliced into various logically isolated sub-networks. Each network slice can have its own network architecture and protocol to serve one application exclusively. ADN enhances the user experience while keeping the resource efficiency by further imposing multiplexing among these logically isolated sub-networks.

CCS Concepts

•**Networks** → **Network architectures; Network services;**

Keywords

Application Driven Network; NFV; Network Slicing; SDN; DiffServ

1. INTRODUCTION

Traditional resource efficient architecture has become a barrier to meet the diverse application requirements, and it is inevitable that the future network should be application driven. Application diversity leads to the varied resource requirements. Although the bandwidth of network increases and the unit cost decreases, current networks are still facing poor user experience and low application satisfactory.

Bandwidth guarantees can be achieved through static reservations [2], which lead to inefficient utilization of network resources as the slice of one application cannot be used by another application. Considering that the average bandwidth utilization of an application is low and the traffic is bursty in nature [3], work-conservation manner provides minimum bandwidth guarantees.

However, applications demand more performance guarantees beyond bandwidth, such as latency, co-flow finish time. In reality, there are many applications that the current network can not well supported due to the diverse requirements. For example, some IOT applications, such as using water meters and electricity meters to automatically upload the usages, have a huge number of terminals. While the bandwidth requirements of these applications are small, they require a significant demand for the control channel and they are very sensitive to the cost. Current architectures can not meet the requirements while maintaining a low cost. Another example is the instant messaging applications (such as Wechat), which cause signaling storm problem. These applications consume a small bandwidth, but need to be constantly refreshed to keep connected, which can be badly supported by existing networks. The former case can be solved by network slicing, and the latter case can also be solved by preserving adequate signaling resources.

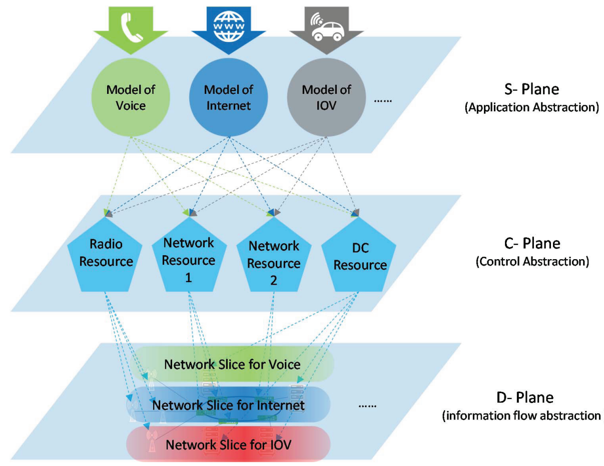

Figure 1: The high-level overview of ADN architecture.

Thus, we aim to propose a new network paradigm that:

- Meets the application requirement efficiency while achieving the resource efficiency.
- Provides minimum performance guarantees.
- Evolves current IP network to provide on-demand QoS for different applications.

In this paper, we propose Application-driven networks (ADN), a solution that achieves our goals. ADN proposes to build networks for applications and provide a logically independent network to satisfy the unique requirement of every application. ADN can be optimized according to the characteristics of various application configurations while maintaining the total resources unchanged. It is shown that the complexity and performance can be improved when the delay requirements of different applications are pre-known [1]. To be more specific, a physical network in ADN is sliced into various logically isolated sub-networks. Each network slice can have its own network architecture, protocol and serves one application exclusively. ADN enhances the user experience while keeping the resource efficiency by further imposing multiplexing among these logically isolated sub-networks. Their weights are dynamically adjusted according to the running states.

ADN evolves current IP network to provide on-demand QoS for different applications by mapping the distinct properties of applications to respective network resources. To abstract the application requirements on the network, ADN analyzes the different requirements of different applications on the network, construct multi-dimensional application abstraction model and arrange network resources to satisfy different applications. Existing approaches such as NFV and network slicing can divide the original unified and unique network (such as wireless air interfaces, link bandwidth, computing power, storage spaces) into multiple logical indepen-

SIGCOMM'16 August 22–26, 2016, Florianópolis, Brazil

ACM ISBN 978-1-4503-4193-6/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2934872.2934903

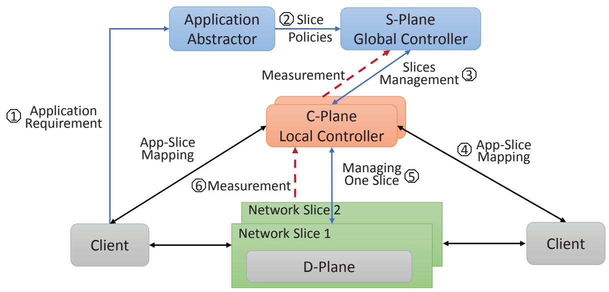

Figure 2: The framework and workflow of ADN.

dent elements. ADN use this basis to further multiplex the network resources to meet the application requirements. Based on the fast/slow neural control theory [4], ADN provides fast and slow control for network resource with respect to time, space and value, etc. A slow controller takes slow varying parameters such as network topology and traffic property as inputs, and determines the optimal network slicing and operating points. A fast controller conducts real-time measurements over fast varying parameters such as the queues in switches and link states, and uses Kalman filter algorithm to operate each slice at the optimal control point with the minimal cost. With both the fast and slow controllers, the global network can operate at the optimal control point, fulfilling the services with respect to applications.

2. ARCHITECTURE OVERVIEW

ADN consists of a resource orchestrator for mapping the application requirement to the network slice, controls for managing slices, virtualization-capable network devices for providing logical subnetworks. Figure 1 is an overview of ADN architecture.

- S-Plane abstracts the application requirements of networks, such as bandwidth, latency, number of connections, and allocates appropriate sub-networks for applications. In the running time, S-Plane measures the states of sub-networks and adjusts resources among slices dynamically.
- C-Plane controls one sub-network to provide the specific service for the associated application. Similar to the SDN controller, C-Plane is a local controller with the capability to react the sub-network changes rapidly.
- D-Plane is the abstraction of network devices which can provide logically independent slices for applications via network function virtualization.

The framework and workflow of ADN are illustrated in Figure 2. When applications use the network to transfer information, applications can submit their requirements of QoS directly or indirectly to the *Application Abstractor*, which abstracts the demands to network resources and sends slice policies to the *Global Controller* in S-Plane. For applications running in conventional models, i.e., directly using the network without requiring QoS specifically, ADN abstracts the QoS requirements of applications by measuring and analyzing their traffics in a runtime.

The global controller receives slice policies from the application abstractor and then allocates physical network resources including control plane resources in C-Plane and data plane resources in D-Plane to applications. To improve the network performance, the global controller dynamically adjusts the weights of slices according to the network running states which are collected and analyzed from the measurement data.

A local controller manages the network resources in a slice. It measures the running states of a sub-network and conduct flow and congestion controls. Given only partial and local view, the local controller responses to the network dynamics more rapidly and therefore helps to improve the performance of applications in the same slice.

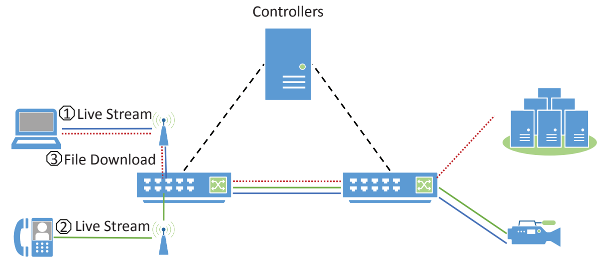

Figure 3: The framework of ADN demo.

NFV plays a critical role in D-Plane providing multiple network slices which can be dynamically created, modified, deleted and multiplexed by the global controller.

3. DEMONSTRATION

In our demo, there are two types of applications with different service level requirements running on the very same network infrastructure at the same time. The overall bandwidth is insufficient yet the bandwidth allocations can be better arranged given the different natures of the applications. As illustrated in Figure 3, the *Live Stream* application requires high bandwidth and is sensitive to the network latency. On the contrary, the *File Download* application is also bandwidth-hungry yet is tolerant to large network latency. Bandwidth and latency are the key metrics in ADN. To satisfy both, ADN leverages complex speed limitation function and queue scheduling on switches to achieve a performance-guaranteed and multiplexing-capable isolation among network slices.

In our demo, each application is assigned to a network slice, while the controller monitors the network dynamics and allocates bandwidths among different slices accordingly. The demo shows the progress that ADN adapts to the applications' requirements and allocations the network bandwidth dynamically such that the user experiences of all applications are maximized.

4. CONCLUSION AND FUTURE WORK

As an evolutionary network architecture, ADN has two key features. i.e., (1) meeting the QoS requirements of different applications in an on-demand manner and in a runtime; (2) 2-layered and hierarchical control architecture that enhances the user experience while keeping the resource efficiency.

We will address the following challenges in the future work. (1) Developing a better solution that models the network resources, abstracts the QoS requirements of applications, and maps the demands to network slices. (2) Seeking for more efficient ways to support a large number of network slices. (3) Designing a scalable distributed and layered cooperative control scheme which achieves higher resource efficiency and meanwhile addresses the actual needs of applications more rapidly. (4) Designing a network sampling mechanism to measure the running states of logical/physical networks efficiently and accurately for the sake of application profiling.

5. REFERENCES

[1] M. Alizadeh, S. Yang, M. Sharif, et al. pfabric: Minimal near-optimal datacenter transport. *SIGCOMM*, 2013.

[2] H. Ballani, P. Costa, et al. Towards predictable datacenter networks. In *SIGCOMM*, 2011.

[3] T. Benson, A. Akella, et al. Network traffic characteristics of data centers in the wild. In *SIGCOMM*, 2010.

[4] J. C. Doyle and M. Csete. Architecture, Constraints, and Behavior. *PNAS*, 2011.