hBRAS: Towards High-performance Broadband Remote Access Services with Programmable Switches

Yu Zhou★, Wenhua Li†, Jun Bi★, Yaning Ma†, Chen Sun★, Piaoping Chen†

★Institute for Network Sciences and Cyberspace, Tsinghua University

★Beijing National Research Center for Information Science and Technology (BNRist)

†CertusNet Inc.

1 INTRODUCTION

Broadband remote access servers (BRAS) work as crucial elements to provide broadband network services in Internet. The responsibility of BRAS includes routing, ACL, QoS, tunneling, and so on. Traditionally, BRAS is implement with dedicated and closed hardware, and faces dozens of problems, such as vendor lock-in, high cost, and manual management. To handle this dilemma, a recent trend is to implement virtual BRAS (vBRAS) in the network function virtualization (NFV) framework. With the flexibility of NFV, vBRAS can bring a lot of significant benefits, such as being economically implemented on Commercial Off-The-Shelf (COTS) servers, programmable management, considerable scalability, etc.

However, the benefits of vBRAS come with the performance compromises. As the performance of BRAS is important to guarantee the quality of broadband network services and BRAS needs to process heavy network traffic, the performance compromise inevitably prevents vBRAS from large-scale deployment.

The rising of programmable switches (e.g., Tofino [2]) and domain specific languages (e.g., P4 [1] and POF [3]) opens new opportunities for solving above problems. Programmable switches oblige operators with more flexibility to define novel functions and protocols, making fast innovations possible on networked systems. Various P4 programs have been built to enhance many perspectives of networks, such as security, performance, and measurement functions. Further, programmable switches could support in-network computation or caching to improve the performance of upper applications or alleviate network-bound bottlenecks. In this demo, we are considering to improve the performance of BRAS using programmable switches.

To reap the benefits of vBRAS while provisioning high-performance services, we present a solution, hBRAS, aiming to build high-performance BRAS with the acceleration of programmable switches. The key design of hBRAS is to offload network functions on programmable switches and directly process some traffic in programmable switches without the involvement of COTS servers. Obviously, hBRAS does not include all network functions. We thoroughly analyze all network functions in hBRAS to check whether they can be implemented in programmable switches. Further, to guarantee the correctness of network functions in vBRAS and programmable switches, We also provide a design for communications between these components.

We have implemented hBRAS with Tofino whose forwarding capacity is up to 6.4 Tbps. Our evaluation shows that hBRAS could largely outperform vBRAS: hBRAS could keep line rate when processing 128-bit packets of 80Gbps, while vBRAS provides a throughput of merely 40Gbps. The processing latency of hBRAS can be as small as $1\mu s$.

2 DESIGN

The architecture of exiting vBRAS systems commonly comprises two layers, i.e., the data plane and the control plane. Both of them run in the virtual machines. As shown in Figure 1, hBRAS is composed of three layers, *i.e.*, the control plane in virtual machines (VMs), the data plane in VMs, and programmable switches. Compared with the existing vBRAS architecture, we replace legacy switches with programmable switches to offload packet processing. Further, some modifications on the data plane and the control plane make them work in harmony with programmable switches. Next, we will respectively introduce the these three layers.

Control Plane in VM: The centralized control plane is the brain of BRAS and provides management-related functionalities, such as providing north-bound API for operators, management of IP address pool, customer management, and so on. We build a programmable-switch-aware BRAS control plane. As different traffic may need to be processed by different network functions or even chains of network functions, we design an approach that is similar to service function chaining to coordinate network functions residing at different layers.

Data Plane in VM: The data plane resides at the edge of broadband networks and enforces user policies. It takes the responsibility of traffic switching, quality of services, traffic accounting, multicast, access control, keeping PPPoE sessions alive, and so on. Most traffic will only traverse the data plane which is implemented in VMs. Thus the data plane becomes the bottleneck for improving the service quality. To

This demo is supported by the National Key R&D Program of China (No.2017YFB0801701) and National Science Foundation of China (No.61472213). Jun Bi (junbi@tsinghua.edu.cn) is the corresponding author.

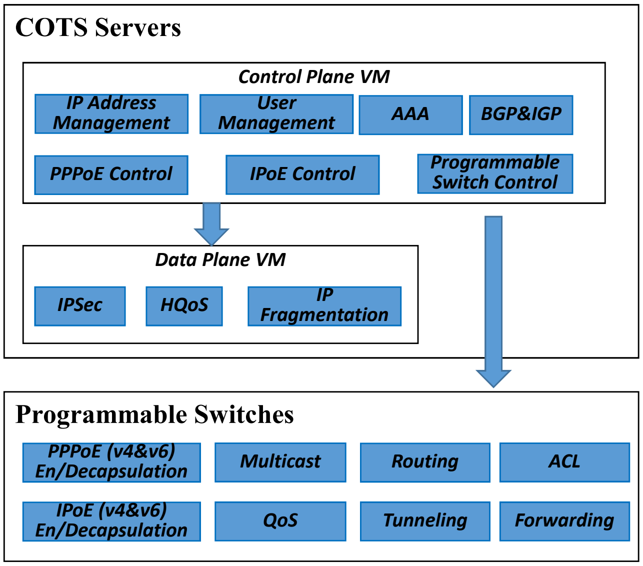

Figure 1: Architectural overview of hBRAS.

this end, we introduce programmable switches to accelerate the data plane, and most functionalities of the data plane can be directly offloaded to programmable switches whose performance is 100 times higher than VM.

Programmable Switches: To provide high-performance broadband remote access services, we introduce programmable switches into BRAS. The incorporation of programmable switches inevitably raises concerns on the constrained programmability and resources (compared with COTS servers). On the one hand, we thoroughly check all network functions in BRAS and have successfully offloaded dozens of functionalities from the data plane to programmable switches, such as traffic switching and quality of services. On the other hand, we develop programmable switches as a cache layer for the data plane and control plane in VMs. If a packet hit a table entry in a programmable switch, then it can be directly forwarded. Otherwise, the programmable switch should transfer the packet to the data plane for further processing.

3 EVALUATION

Evaluation Setup. To demonstrate the benefits of hBRAS, we present a proof-of-concept evaluation in this demo. We deploy hBRAS on a Tofino switch and use a Dell R730 server to run vBRAS. hBRAS are installed with 126K PPPoE sessions and contains 600K IPv4 forwarding rules. The evaluation of vBRAS has the same settings with hBRAS. We use a packet generator with eight 10Gbps ports to test the throughput and processing delay. We use four ports to simulate the traffic from the customers, while the other four ports are used to simulate the Internet access. The experiments last for more than twelve hours to test the stability of hBRAS.

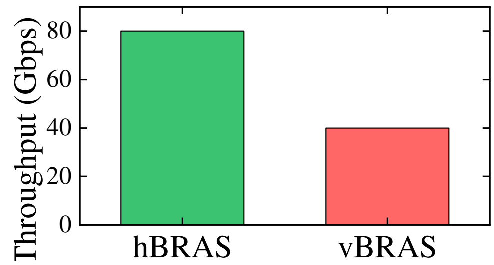

Figure 2: Throughput of hBRAS and vBRAS.

As shown in Figure 2, hBRAS can keep line rate when the packet generator emits 80bps 128-bit packets, but vBRAS cannot achieve the 40 Gbps throughput. Based on the above result, we can draw an elementary conclusion that with the switch offloading technique, hBRAS largely outperforms vBRAS. Further, the average processing delay of hBRAS is as low as 800 nanoseconds, while the maximum delay is around 1.5 milliseconds, which is much smaller than vBRAS.

4 DEMO DESCRIPTION

In this demo, we will present a prototype of hBRAS, which is deployed in a network similar to the evaluation setup. Then, we will respectively show the functionalities of different layers, and how packets are processed in hBRAS. Further, we will demonstrate the performance of hBRAS.

Demo requirement. To conduct the demo, we require power lines, a monitor, a table, and network connectivity.

5 ABOUT CERTUSNET

CertusNet Inc. is a world-leading provider of information platform and application solutions. As a pioneer of Elastic Network and an advocate of Elastic Services concepts, the company is committed to providing Elastic Network services covering cloud, network and terminal for global users, enterprises and government agencies. CertusNet strives to continuously improve user experience, facilitate new services and applications, and help customers generate greater value with its widely trusted Elastic Services.

REFERENCES

[1] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: Programming Protocol-independent Packet Processors. *SIGCOMM Comput. Commun. Rev.* 44, 3 (July 2014), 87–95.

[2] Barefoot Networks. 2018. Barefoot Tofino. Website. (2018). https://barefootnetworks.com/technology/.

[3] Haoyu Song. 2013. Protocol-oblivious Forwarding: Unleash the Power of SDN Through a Future-proof Forwarding Plane. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN '13)*. ACM, New York, NY, USA, 127–132. https://doi.org/10.1145/2491185.2491190