Application-aware and Dynamic Security Function Chaining for Mobile Networks

Guanglei Li*, Huachun Zhou, Guanwen Li, and Bohao Feng
Beijing Jiaotong University, Beijing, 100044 China
{15111035, hchzhou, 16111011, bohaofeng}@bjtu.edu.cn

Abstract

Mobile networks have urgent demands of fine-grained, cost-effective and flexible service provision for diversified user traffic. To cope with these demands, researchers have proposed various Service Function Chaining (SFC) solutions with the rise of Software Defined Networking (SDN) and Network Function Virtualization (NFV) technologies. However, most of them are performed based on MAC address and/or OpenFlow protocols without Network Service Header (NSH) support, having drawbacks in complexity, scalability and flexibility. NSH-based approaches are more promising for mobile networks, since they support metadata-based packet information sharing and policy enforcement. Moreover, a hierarchical SFC (hSFC) architecture is proposed to alleviate the scalability and management problems in large-scale networks. Nevertheless, how to realize application awareness and on-demand service provision has not been investigated thoroughly in the hSFC environment. Thus, in this paper, we propose a proactive-based branching approach for application-aware and dynamic security function chaining, where application features are analyzed at first, and then carried in the metadata of NSHs for subsequent processes by the relevant security functions. In this way, the data plane is able to redirect traffic based on metadata without the participation of control plane. Besides, we verify the proposed approach through our prototype system via two typical use cases, the application-aware traffic control and lawful interception, and the related experiment results confirm its feasibility and elasticity.

Keywords: Mobile Networks, Security Function Chaining, Application awareness

1 Introduction

Service Function Chaining (SFC) is an ordered set of Service Functions (SFs) that handles the traffic of delivery, control, and monitoring of a specific service/application [11]. Mobile networks have large demands for network functions or chains, including common functions such as NAT and DPI, and value added service such as parental control, malware detection and elimination and lawful interception, and some other functions such as TCP optimization, video optimizers and HTTP header enrichment [4]. The traditional SFC approaches used in SGi-LAN of LTE networks are usually coarse-grained and static, where each function has to process all the traffic and introducing new SFs costs much time and expenditure, against today's operators' urgent demands of fine-grained, cost-effective and flexible service provision for diversified user traffic.

With the rise of Software Defined Networking (SDN) and Network Function Virtualization (NFV), there have been many brand-new SFC approaches to realize policy-driven, fine-grained and dynamic SFC for mobile networks. However, most of them are based on MAC address and/or OpenFlow protocols without the Network Service Header (NSH) support [11]. Compared with NSH, these approaches have drawbacks in scalability and complexity, as the number of flow tables will grow dramatically with

Journal of Internet Services and Information Security (JISIS), volume: 7, number: 4 (November 2017), pp. 21-34
*Corresponding author: School of Electronic and Information Engineering, Beijing Jiaotong University, No.3 Shangyuancun, Haidian District, Beijing, 100044, China, Tel: +86-188-1038-8516

the fine-grained traffic classfication, and most of them do not consider policy enforcement. NSH is a special header for SFC and can use VxLan-GPE as the encapsulation protocol, carrying identifiers of service paths (Service Path Identifier, SPI) and service function index (Service Index, SI). It can also attach metadata (type 1: mandatory fixed four context headers; type 2: optional variable length context headers) which carries context information about the packets. Thus, it supports multi-tenancy/isolation, context awareness, dynamics and policy enforcement by nature [15]. Moreover, the NSH enables network functions to be composed on-demand according to metadata, thus, a service function chain can be any kind of directed graph instead of linear and non-related blocks [13].

The NSH protocol is promising to be used widely and the IETF SFC WG discusses a lot of use cases in datacenters and mobile networks [4, 6]. Further, as the IETF SFC is expected to be deployed in large-scale networks, a hierarchical SFC (hSFC) architecture, which divides a SFC domain into a top-level (Top-domain) and independent sub-levels (Sub-domains), is proposed to alleviate the problems of complex management and orchestration, as well as challenges of multi-technology and multi-administration [2]. The hSFC is believed as a nature step of SFC evolution and can be adopted by mobile operators. Focusing on path configuration problems, the authors in [17] give an implementation of hSFC based on OpenDaylight SFC Project [14].

However, besides path configuration, there are other problems needed to concern for fine-grained and dynamic service chaining. Specifically, how to realize application awareness and on-demand SF provision has not been investigated thoroughly in the hSFC environment. Thus, in this paper, we propose a proactive-based branching approach for application-aware and dynamic security function chaining, where application features are analyzed at first, and then carried in the metadata of NSHs for subsequent processes by the relevant security functions. In this way, the data plane is able to redirect traffic and create branching based on metadata without the participation of control plane. Besides, to enable dynamic branching and ensure proper end-to-end SFC operation, we propose a metadata-based packet information sharing and a context header allocation scheme. On-demand service function composition is also considered for avoidance of repeated packet inspection and effective policy enforcement. Besides, we design two different methods for branching in Sub-domains and the Top-domain of the hSFC environment. Finally, we verify the proposed approach through our prototype system via two typical use cases, the application-aware traffic control and lawful interception, and the related experiment results confirm its feasibility and elasticity.

In summary, the contributions of this paper are listed as follows.

1. We implement the hSFC based on OpenDaylight SFC project. Particularly, based on the NSH mandatory type-1 context header, we design metadata allocation for packet context information sharing in such multi-domain network. Based on DPI and well-known identifier values stored in the context header, we realize application awareness and information sharing among SFs in all domains.

2. Further, path branching approaches are proposed to realize dynamic on-demand service function scale out. We give two different proactive-based branching approaches according to characteristics of the hSFC architecture. We confirm the effectiveness of our design and implementation by two use cases.

The rest of this paper is organized as follows. In Section II, we present related work about SFC, and relevant researches on the corresponding field of SDN and NFV. In Section III, we describe our design and implementation. In Section IV, we show the use cases and experiment results. In Section V, the conclusion of this paper is given.

2 Related Work

SDN technologies make it facile for operators to manage and configure networks. Logical centralized control enables automated network configuration and dynamic traffic steering. As the most popular southbound protocol, OpenFlow protocol provides flexible packet switching based on rich matching fields in Layer 1 to Layer 4 defined in OSI model. NFV enables virtualized network functions, and it provides a new way to deploy and manage network services. As NFV realizes flexible and cost-effecitive network service provision based on virtualization, the combination of SDN and NFV makes each other more compelling.

Finer granularity of perception and control to flows has got much attention from researchers for purposes such as traffic management [5, 8], service delivery [10] or security [7, 20]. Based on application information from Layer 7 and context information of traffic, different types of network services can be deployed adaptively and dynamically in SDN and NFV scenarios, to satisfy high-level policies (QoS policy, traffic manage policy, security policy, etc.) defined by operators and managers. Relying on dynamic components in control plane [5, 7, 8, 10] or metadata sharing and prefetched countermeasures in data plane [15, 20], a network itself is able to automatically deal with network events (congestion, overload, failure, security threat, etc.). Thus, network operators and managers are released from countermeasures for specific network events and complex manual configurations.

SFC is a hot research topic since the rise of SDN and NFV, and it is promising to realize policy-driven and dynamic network service management. The authors in [11] and [16] conclude advantages and disadvantages of SFC approaches proposed by academic researchers, industry, standards and open source communities. Although NSH-based approaches are run as overlays on the top of IP networks and has drawbacks in header overheads, the NSH in IETF SFC Working Group is promising better than most MAC-address-based approaches, as they do not support policy enforcement and have the problem of scalability.

Figure 1 depicts an overview of the IETF SFC architecture used in SGi-LAN of mobile networks. The Classifier (CF) is responsible for classification of traffic into different chains and encapsulating/de-capsulating NSHs for packets. The Service Functions Forwarder (SFF) is responsible for steering traffic to SFs according to SPI/SI in the NSH, and SFs will decrement the SI after processing. If a NSH-unaware SF is used, a SFC proxy must help it encapsulate/de-capsulate the NSHs. The NFV, SDN and

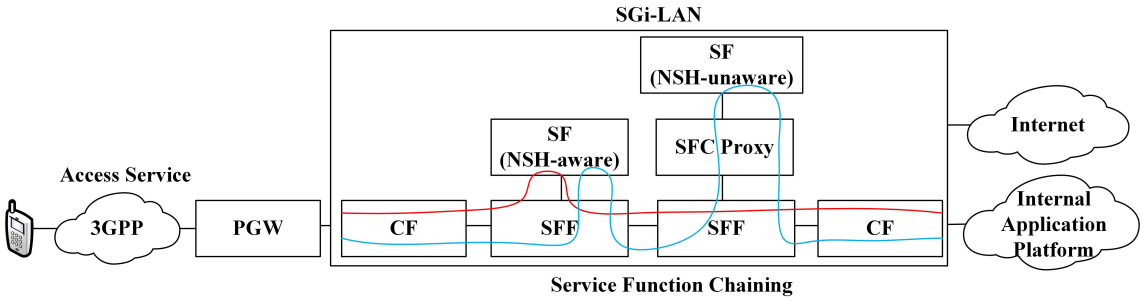


Figure 1: Overview of IETF SFC in mobile networks

SFC technology can remove the limitations of traditional SGi-LAN service approaches, which chain all functions into linear non-related blocks, and its drawbacks include that each function has to process all the traffic, that a packet may suffer unnecessary and repeated inspection by different functions and that a failure of any single SF causes interruption of the all service.

The hSFC architecture is proposed further by the IETF SFC Working Group. The hSFC architecture divides a large network into multi-domain networks and has multi-levels. As the example depicted

in figure 2 shows, a Top-domain is the entire network domain and includes CFs, SFFs, SFs and Sub-domains, Sub-domains are regarded as lower levels and each of them probably acts as a subset of the total paths in the higher-level domain. The control and management of different domains are independent, hence an Internal Boundary Node (IBN) is proposed to bridge packets between higher and lower layers.

The authors in [17] indicate that the hSFC architecture brings many benefits. For example, it can reduce management complexity in large datacenters, realize flexible service decomposition/composition, connect datacenters and enable distributed multi-domain NFV over a wide range of regions, and simplify load-balancing by distributing SFs in Sub-domains and pooling Sub-domains in a high-level region. In a typical SGi-LAN, the hSFC architecture can realize service decomposition/composition and simplify load-balancing by deploying functions in different Sub-domains. In the Top-domain, the operator can steer traffic based-on load-balancing policy and use coarse-grained classification approaches such as network prefixes or subscriber groups. The Sub-domain could use fine-grained classification approaches such as TCP, UDP ports or application protocols.

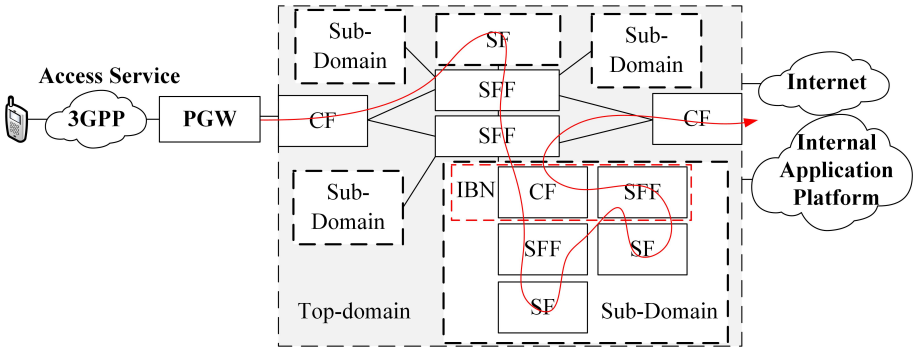


Figure 2: Hierarchical SFC

Based on the hSFC architecture and OpenDaylight SFC project, we give an implementation of application-aware and dynamic security function chaining for mobile networks. We use the popular opensource DPI software nDPI [12] to inspect application protocol and the application protocol identifiers in nDPI are used as the well-known identifier values shared among SFs.

3 Design and Implementation

In this section, we give our design and implementation of application-aware and dynamic security function chaining. Firstly, we present our implementation of hSFC and environment setup. Then, we give our design and implementation of application awareness and policy enforcement. The metadata allocation in context headers is given and the application awareness ability is based on nDPI. Finally, we conclude the passive and proactive dynamic SFC methods and design two different service path branching approaches for the Sub-domain and Top-domain based on proactive context-based forwarding.

3.1 Hierarchical SFC Setup

In the hSFC architecture, as figure 3 shows, the Top-domain usually steers traffic based on coarse-grained policy and the Sub-domain will reclass the traffic based-on fine-grained policy and generate more service paths. The path configuration of IBN is the key to bridge packets in Top-domain and Sub-domains. When packets enter in a Sub-domain from the top-domain, the IBN will replace the SPI/SI using the SPI/SI in the Sub-domain, and when packets leave the Sub-domain and renter the Top-domain, the SPI/SI of Top-domain should be restored.

We follow the path configuration approach in [17]: store SPI/SI of the Top-domain in the third mandatory context header (MCH-3) of NSH and the IBN consists of the Sub-domain ingress CF and the last SFF of a service path. When Top-domain traffic enters the Sub-domain, the IBN decreases SI for the Top-domain and stores SPI/SI in the MCH-3. When the traffic has been processed by all SFs and reenters the Sub-domain from the last SFF of its path, the last SFF restores the SPI/SI from the MCH-3. We further add a new action "dec_nsh_nsi" for Openflow and decrease the SI for Top-domain traffic in the Sub-domain ingress CF.

Based on our OpenStack-based network emulation platform (EmuStack) [9], we build a hSFC environment using kernel-based virtual machines (KVM) and virtual networks (VN). Figure 4 presents three connected Sub-domains using VN1, VN2 and VN3 respectively and connected by VN4. In each domain, the OVS-2.6.1 with yiyang' NSH patch [19] is utilized as the CF and SFF. The SFs are based on docker containers and co-located in the same KVM with the OVS-based SFF. Host1 and Host2 access the top-domain via VN5 and VN6 and are used to generate traffic. The number of SFF and SF in each domain is scalable.

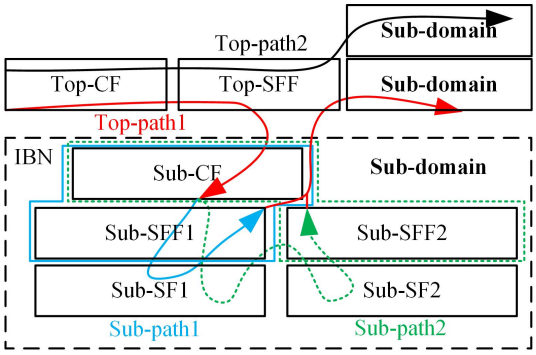


Figure 3: Hierarchical SFC Path configuration

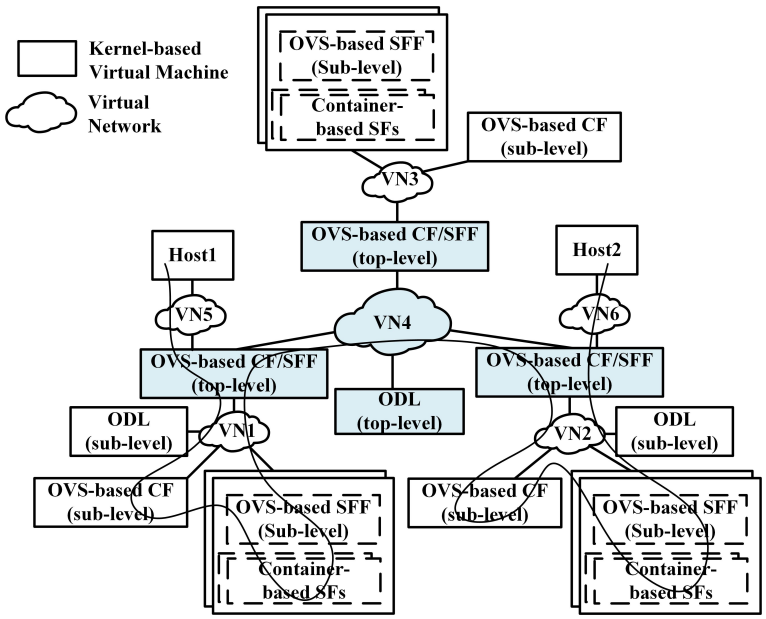


Figure 4: Environment Setup

3.2 Application awareness and Policy Enforcement

3.2.1 Metadata Sharing

The end-to-end performance would descend because of repeated and costly operations in the long-distance service chain, such as DPI, which is a basic process engine deployed in many functions [1]. It's naturally to utilize metadata to share information and avoid some repeated inspection to packets. The hierarchical network consists of multi-domains and their administration are independent, so the definition and meaning of metadata may vary in different domains. In some cases, metadata are shared within Sub-domains and the process to NSH header in Sub-domains are transparent to each other. However, in some cases metadata should be shared among different levels to ensure proper end-to-end SFC operation. This problem is described as "Gluing levels together" in draft [2], and it proposes to use well-known identifier values with global registry or well-known labels to map the actual identifier, which is assigned by control plane system. Hence, proper metadata allocation and appropriate composition of Sub-domains and SFs for specific service/application need to be considered.

In the hSFC environment, if we compose different Sub-domains and SFs properly for specific service or policy, metadata-based packet information sharing can improve efficiency. For example, in a security defense case, the service can be composed of two Sub-domains. The first Sub-domain can realize application awareness and access control. The second Sub-domain performs heavier detection. The application information acquired in the first Sub-domain can be carried in context header, then the IDS in the second Sub-domain can skip the application detection phrase and detect suspicious activity for this application directly.

Figure 5 shows an example of DPI-based application information sharing. The DPI function is deployed in the front of the chain. It provides Layer 7 information of packets for the subsequent Firewall (FW) and Intrusion Detection System (IDS). The composition of DPI and a Layer 4 Firewall can realize a Layer 7 Firewall, and the IDS does not need to have a DPI engine. To avoid becoming the bottleneck, we deploy the DPI in Sub-domains rather than the ingress of Top-domain. The top-domain can steer traffic to different Sub-domains to balance load.

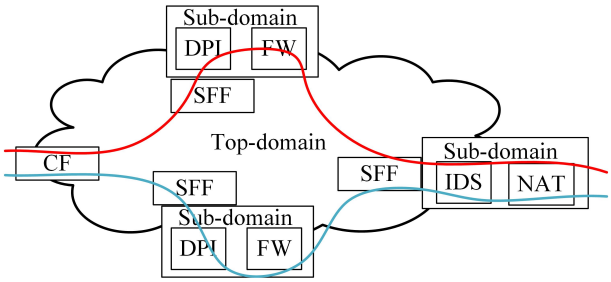


Figure 5: An example of DPI-based metadata sharing

3.2.2 Context Header Allocation

The NSH has two kinds of context header, the type 1 is mandatory fixed four 4-byte context headers and the type 2 is optional variable length context headers for vendor-specific SFs. As the OVS and OpenDaylight support the type 1 and we have followed the path configuration approach in [17], our current implantation relies on type 1: the mandatory fixed four 4-byte context headers.

According to our use cases and the hSFC architecture, the context header allocation scheme is presented in figure 6. MCH-1 and MCH-3 are occupied by the hSFC implementation to bridge packets in the Top-domain and Sub-domain. MCH-1 stores the address of Top-domain SFF which is directly attached

to the Sub-domain, as when packets reenter into the Top-domain the destination should be retrieved from the packet itself. MCH-3 stores the SPI/SI of the Top-domain. To share application information, we use MCH-2 to store the application protocol identifier and the value follows the definition in nDPI. The MCH-4 is used to store security context, the DPI and IDS function can write values in this header to present different events such as Blacklisted Host, TCP SYN flood and SQL scan. The values of different application identifiers and events are well known by assigning them in the control plane system.

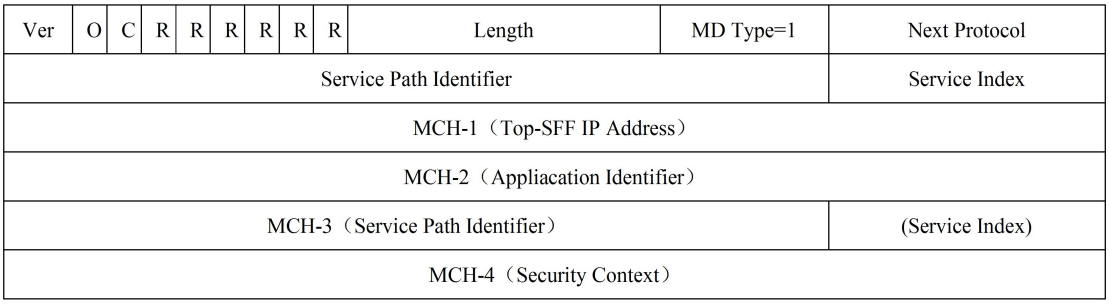


Figure 6: Sub-domain Network Service Header

3.2.3 Metadata-based Policy Enforcement

To realize metadata-based policy enforcement, we modified the python-based NSH script in OpenDaylight SFC Project [18], which acts as a SF and has naive forwarding ability. Besides processing the NSH, we make it store local policy associated with metadata and can react to the packets according to policy and metadata in the context header. For example, the FW can block the packet if its MCH-4 stores the value that presents TCP SYN flood.

What's more, to cooperate with NSH-unaware function such as nDPI, we modify and deploy this script as a wrapper co-located with the NSH-unware function. Before the NSH is standardized and supported by kinds of network functions, it is a temporary and feasible way to use NSH-unware but powerful functions as the packet process engine. The python-based script acts as a proxy encapsulating/decapsulating original packets to/from VxLan-GPE and NSH encapsulated packets. It also acts as a policy agent to update metadata according to inspection results of NSH-unaware functions. In our use cases, the script co-locates with nDPI and writes application identifiers into MCH-2 after nDPI reports inspection results.

3.3 Proactive-based Dynamic Branching

Metadata-based reclassification and dynamic service path branching enables fine-grained on-demand service and keeps a function from processing all traffic all the time. For example, a DPI function can reclass user traffic after inspecting the application protocol by updating the context header, and steer the flow to a new path to pass through a traffic shaper function, then the traffic shaper can react to this flow based on the context header and return back the flow to original path after process. While, other application traffic can still traverse the original service path. Metadata-based reclassification and policy enforcement brings more flexible.

In general, there are two kinds of way to realize dynamic branching. The passive way uses metadata to trigger the control plane to reconfigure the data plane in time. While, the proactive way relays on prefetched configuration and does not notice the control plane when an event happens. It's obvious that the proactive way is agiler than the passive way, although it increases cost in data plane such as more

flow tables. Most dynamic systems in literatures only support the passive way, as they do not design a data plane protocol like NSH to carry metadata. The PSI [20] uses flow tag to realize proactive tag-based forwarding based on their previous work [3] and proves that the proactive way is much agiler, however, it has the problem of scalability as it uses the ToS field in IPv4 header to store metadata. The NSH is able to realize proactive context-based forwarding naturally and we argue that it's necessary to utilize the dynamic ability in mobile networks for fine-grained and agile on-demand service provision. Thus, we design two types of proactive-based dynamic branching approaches used in Sub-domain and Top-domain, respectively, according to the hSFC architecture's characteristics.

In the hSFC environment, service function composition/decomposition is recursive. The top-domain has the knowledge of Sub-domains' attribute and steers traffic based on proper Sub-domain composition. In each Sub-domain, the situation is similar. SFs are composed properly to provide specific network service.

Figure 7 and Figure 8 present two proactive branching approaches for Sub-domain and Top-domain. In figure 7, the branching relies on SFs. When SF1 detects a new context such as a suspicious activity, it can modify SPI/SI of packets and SFFs steer packets to SF3 (for heavier inspection) based on the new SPI/SI. In this approach, SFFs are not aware of context and only pre-configured with flow entries to match different SPI/SI. In figure 8, the branching relies on SFFs. The context information produced by Sub-domain1 is carried in the NSH context header. The Top-domain SFF can reclassify the packets based on the well-known metadata value in the context header.

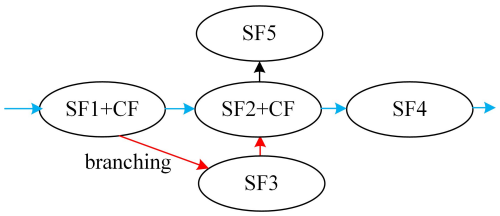


Figure 7: Branching in a Sub-Domain

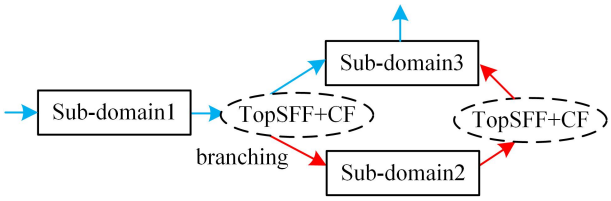


Figure 8: Branching in the Top-Domain

As the management of the Top-domain and Sub-domain is independent, it's not feasible to make Sub-domain change the SPI/SI of the Top-domain. The branching approach in figure 8 adopts to the Top-domain. In Sub-domains, as SFs are various and produce different metadata, the branching approach in figure 7 decreases the complexity of SFF and is applicable to Sub-domains.

4 Experiment Results

Our current implementation has realized application awareness and branching in the Sub-domain, and the experiments in this section focus on the asymmetric and symmetric flow branching in the Sub-domain. We design two use cases: application-aware traffic control and lawful interception and present experiment results.

4.1 Tarffic Control

In some cases, the network operator intends to limit specific traffic and relies on DPI to detect the application layer information such as P2P protocol. In this experiment, we produce two kinds of flows, FTP and BitTorrent, and assume the network operator requires to limit upstream UDP traffic of P2P application. In the traditional approach, all traffic may pass through an application-aware traffic control function, causing unnecessary costs.

In figure 9, we present a dynamic on-demand service provision situation, where only the target flow f2 (upstream BitTorrent flow) is steered to the Traffic Control (TC) function. The Sub-domain1 and

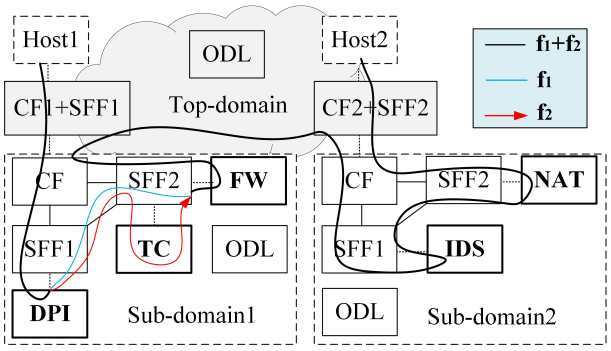


Figure 9: Traffic control for the asymmetric flow

Sub-domain2 are composed to provide a security function chain. The Sub-domain1 are responsible for application-awareness, access control and traffic management, and the Sub-domain2 are responsible for heavier intrusion detection and network address translation. To realize fine-grained and dynamic on-demand traffic control service, only the target flow f2 (upstream BitTorrent flow) should be steered to the Traffic Control (TC) function in the Sub-domain1.

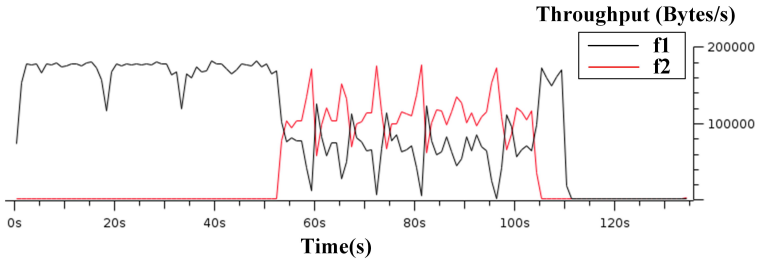
The DPI function in the Sub-domain1 is the combination of python-based script and nDPI. As long as the target flow f2 is identified, the script will modify SPI/SI (from 343/254 to 344/254 in our experiment) of the packets and update the MCH-2 with application protocol identifier (The BitTorrent ID in nDPI is 37). SFF1 and SFF2 in the Sub-domain1 will steer the packets of f2 to the TC function according to new SPI/SI value. In this way, only the target flow is processed by the TC function. What's more, the TC function is not necessary application-aware and can use a local policy entry based on application identifier such as [application-id, max-bandwidth]. After process, the TC return packets from the branching path to the master path by restoring SPI/SI.

We perform the experiment two times with and without traffic control policy, respectively. Figure 10(a) shows the experiment result without traffic control, when the f2 appears, f1 and f2 compete for bandwidth and their throughput are fluctuating. Figure 10(b) shows the result with traffic control based on application awareness and proactive dynamic branching. The f2 appears at time T1, and the policy agent assign the MCH-2 and change the SPI/SI at T2 after it queries the nDPI and finds that the application protocol has been figured out. As the f2 is re-steered to the TC function and limited, there is no bandwidth competition between f1 and f2. The competition avoidance is timely due to proactive context-based forwarding and prefetched configuration from control plane.

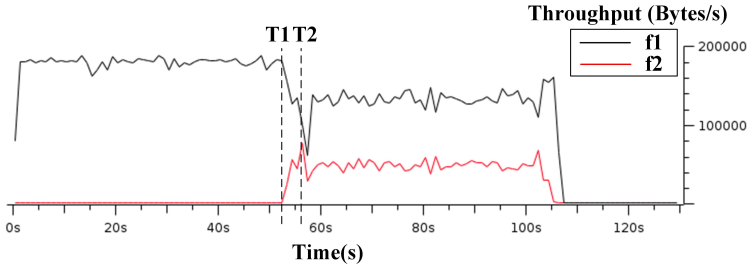
4.2 Lawful Interception

In some cases, upstream and downstream traffic should pass symmetric service paths like TCP flows. When branching, the upstream and downstream traffic should be both steered to the same functions. In this experiment, we deploy dynamic symmetric branching for FTP data flow.

In figure 11, we present the on-demand Lawful Interception (LI) service scenario. The Sub-domain2 has the policy that if the server of f2 (FTP data flow) is recorded in the IDS as a sensitive host, then the data should be intercepted. For the upstream packets, it's straight for the IDS to direct them to the LI function as long as the DPI function in the Sub-domain1 identified the FTP data flow. However, the IDS cannot steer the downstream packets to LI. To solve this problem, an upstream function should participate in the metadata sharing and branching. In the mobile network, a NAT function is usually



(a) Bandwidth competition without traffic control



(b) Traffic control with application awareness

Figure 10: Throughput with and without application-based traffic control

located in the end of the chain and is stateful. It can store an upstream flow's metadata and assign them in the context header for the corresponding downstream packets, then functions in the chain are aware of both upstream and downstream packets' context information.

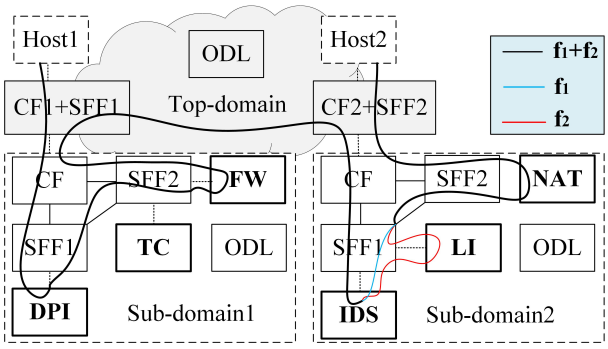
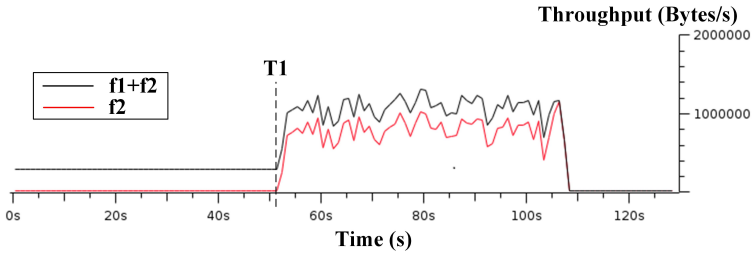
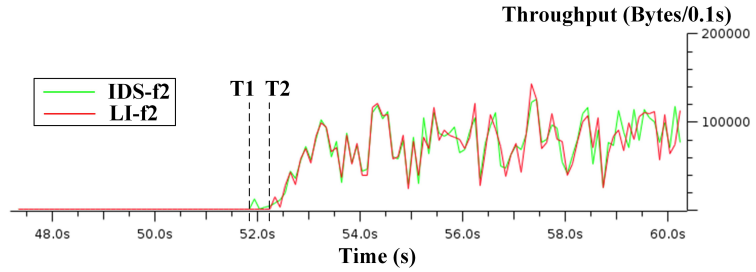


Figure 11: Lawful interception for the symmetric flow

In the experiment, FTP data traffic appears at time T1. After several packets, at time T2, the DPI function recognizes the application protocol and updates the MCH-2 with the well-known identifier (The FTP data protocol ID in nDPI is 175) for all subsequent packets. Then, according to the updated MCH-2, the IDS function in the Sub-doman2 modifies SPI/SI (from 50/254 to 51/254) of subsequent packets and updates MCH-4 with a value (we arbitrarily choose 1) to present the context that the flow should be processed by the LI function. The NAT function check context headers for upstream packets and resign the same value in context headers for the corresponding downstream packets. When the MCH-4 of upstream packets is updated, the SPI/SI of downstream packets will be modified (from 8388659/254 to 8388660/254) by the NAT function, too. In this way, the upstream and downstream packets of the targeted flow will be steered by pre-configured SFF1 to the LI function, as long as the application protocol



(a) Throughput of f1 and f2 in the network



(b) Throughput of f2 in IDS and LI

Figure 12: Throughput of f1 and f2

is detected by the function in the Sub-domain1. After process, LI return packets from the branching path to the master path.

Figure 12 presents throughput of f1 and f2. Figure 12(a) shows that f2 appears at time T1 and throughput in network increases. Figure 12(b) shows throughput (every 0.1 second) of the LI and IDS, the IDS function starts to update context at time T2 and packets are steered to the LI. Based on prefetched branching configuration and policy, the context-based forwarding provides high agility. It is worth mentioning that the branching approach still works if there are more functions in the branching path. The last function in the branching path is responsible for packets restoration to the master path.

5 Conclusion and Future Work

Based on the NSH protocol and the hSFC architecture, we present a design and implementation of application-aware and dynamic security function chaining for mobile networks, to satisfy operators' urgent demands for fine-grained and flexible service provision. Particularly, context header allocation is proposed for packet context information sharing and policy enforcement. Then, a proactive-based branching approach is proposed for on-demand service provision for the dynamic security function chaining, where application features are analyzed first, and carried in context headers for subsequent processes by the relevant security functions. As packet information is shared in the data plane, SFs and SFFs can steer traffic and create branching based on metadata and local policy proactively without the participation of control plane. Finally, we verify the proactive-based branching approach via two use cases, the application-aware traffic control and lawful interception, and related experiment results confirm its feasibility and elasticity.

In the future work, we will focus on the dynamic branching in the Top-domain, as it's important to enable inter-domain service composition and on-demand service provision. In addition, besides using explicit prefetched countermeasures from the control plane for agility, we will enhance the control plane with more resolution capability, as it's necessary for those situations where countermeasures are inex-

plicit and need real time computation, such as resource optimization for cost efficient purpose or price and benefit assessment of multiple candidate countermeasures.

Acknowledgments

This paper is supported by National High Technology of China ("863 program") under Grant No. 2015AA015702, NSAF under Grant No. U1530118, NSFC under Grant No. 61602030 and National Basic Research Program of China ("973 program") under Grant No. 2013CB329101.

References

[1] A. Bremler-Barr, Y. Harchol, D. Hay, and Y. Koral. Deep packet inspection as a service. In *Pro. of the 10th ACM International on Conference on emerging Networking Experiments and Technologies (CoNEXT'14), Sydney, Australia*, pages 271–282. ACM, December 2014.

[2] D. Dolson, S. Homma, D. Lopez, M. Boucadair, D. Liu, T. Ao, and V. Vu. Hierarchical service function chaining. IETF Internet-draft draft-ietf-sfc-hierarchical-02, 2017. `https://tools.ietf.org/html/draft-ietf-sfc-hierarchical-02`.

[3] S. K. Fayazbakhsh, L. Chiang, V. Sekar, M. Yu, and J. C. Mogul. Enforcing network-wide policies in the presence of dynamic middlebox actions using flowtags. In *Pro. of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14), Seattle, Washington, USA*, pages 533–546. USENIX, April 2014.

[4] W. Haeffner, J. Napper, M. Stiemerling, D. Lopez, and J. Uttaro. Service function chaining use cases in mobile networks. IETF Internet-draft draft-ietf-sfc-use-case-mobility-07, 2016. `https://tools.ietf.org/html/draft-ietf-sfc-use-case-mobility-07`.

[5] S. Jeong, D. Lee, J. Choi, J. Li, and J. W.-K. Hong. Application-aware traffic management for openflow networks. In *Proc. of the 18th Asia-Pacific Network Operations and Management Symposium (APNOMS'16), Kanazawa, Japan*, pages 1–5. IEEE, October 2016.

[6] S. Kumar, M. Tufail, S. Majee, C. Captari, and S. Homma. Service function chaining use cases in data centers. IETF Internet-draft draft-ietf-sfc-dc-use-cases-06, 2017. `https://tools.ietf.org/html/draft-ietf-sfc-dc-use-cases-06`.

[7] A. Lara and B. Ramamurthy. Opensec: Policy-based security using software-defined networking. *IEEE Transactions on Network and Service Management*, 13(1):30–42, January 2016.

[8] G. Li, M. Dong, K. Ota, J. Wu, J. Li, and T. Ye. Deep packet inspection based application-aware traffic control for software defined networks. In *Proc. of the IEEE Global Communications Conference (GLOBECOM'16), Washington, D.C., USA*, pages 1–6. IEEE, February 2016.

[9] H. Li, H. Zhou, H. Zhang, B. Feng, and W. Shi. Emustack: An openstack-based dtn network emulation platform (extended version). *Mobile Information Systems*, 2016(3):1–15, 2016.

[10] B. Martini, F. Paganelli, A. Mohammed, M. Gharbaoui, A. Sgambelluri, and P. Castoldi. Sdn controller for context-aware data delivery in dynamic service chaining. In *Proc. of the 1st IEEE Conference on Network Softwarization (NetSoft'15), London, UK*, pages 1–5. IEEE, April 2015.

[11] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz. Service function chaining in next generation networks: State of the art and research challenges. *IEEE Communications Magazine*, 55(2):216–223, February 2016.

[12] nDPI. ndpi, 2017. `http://www.ntop.org/products/deep-packet-inspection/ndpi/` [Online; Accessed on October 3, 2017].

[13] V. N. I. Planning. Verizon sdn-nfv reference architecture, 2016. `http://innovation.verizon.com/content/dam/vic/PDF/Verizon_SDN-NFV_Reference_Architecture.pdf` [Online; Accessed on October 3, 2017].

[14] O. S. Project. Opendaylight sfc project, 2017. `https://wiki.opendaylight.org/view/Service_Function_Chaining:Main` [Online; Accessed on October 3, 2017].

[15] P. Quinn and U. Elzur. Network service header. IETF Internet-draft draft-ietf-sfc-nsh-12, 2017. https://tools.ietf.org/html/draft-ietf-sfc-nsh-12.

[16] I. Trajkovska, M.-A. Kourtis, C. Sakkas, D. Baudinot, J. Silva, P. Harsh, G. Xylouris, T. M. Bohnert, and H. Koumaras. Sdn-based service function chaining mechanism and service prototype implementation in nfv scenario. *Computer Standards & Interfaces*, 54(5):247–265, November 2017.

[17] A.-V. Vu and Y. Kim. An implementation of hierarchical service function chaining using opendaylight platfor. In *Proc. of the 2nd IEEE Network Softwarization Conference and Workshops (NetSoft'16), Seoul, South Korea*, pages 411–416. IEEE, June 2016.

[18] Y. Yi. Nsh tools, 2017. https://github.com/opendaylight/sfc/tree/master/sfc-test/nsh-tools [Online; Accessed on October 3, 2017].

[19] Y. Yi. Ovs nsh pathes, 2017. https://github.com/yyang13/ovs_nsh_patches [Online; Accessed on October 3, 2017].

[20] T. Yu, S. K. Fayaz, M. Collins, V. Sekar, and S. Seshan. PSI: Precise security instrumentation for enterprise networks. In *Proc. of the 2017 Network and Distributed System Security Symposium (NDSS'17), San Diego, California, USA*, February-March 2017.

Author Biography

Guanglei Li received his B.S. degree from Beijing Jiaotong University in June 2015. He is currently working towards the Ph.D. degree in telecommunications and information system at the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing, China. His main research interests are Software Defined Networking, Network Function Virtualization, mobile netwoks, datacenter networks and space networks.

Huachun Zhou received the B.S. degree from the People's Police Officer University of China in 1986. He received the M.S. in telecommunication automation and Ph.D. degrees in telecommunications and information system from Beijing Jiaotong University in 1989 and 2008, respectively. In October 1994, he joined Institute of Automation Systems, BJTU, where he is a lecturer. From Apr. 1999 - Sep. 2009, he was a senior engineer at School of Electronic and Information Engineering, BJTU, and at Network Management Research Center, BJTU. From Oct. 2009 to now, he is a professor in National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU. He has authored more than 40 peer-reviewed papers and he is the holder of 17 patents.His main research interests are in the area of mobility management, mobile and secure computing, routing protocols, network management and satellite network.

Guanwen Li received his B.S. degree in telecommunications engineering from Beijing Jiaotong University in 2014, and then entered National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU. He is currently working towards the Ph.D. degree in telecommunications and information system at BJTU, China. His main research is directed to the architecture of next generation internet, network service management and network security. His other research interests include satellite network and mobile internet.

Bohao Feng received his B.S. degree in telecommunications engineering from Beijing Jiaotong University in 2011, and then entered National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU. He is currently working toward his Ph.D degree in telecommunications and information system at BJTU, China. His research interests are ID/Loc Split network architecture, Software Defined Networking, Information-Centric Networking, network-based caching mechanism, Delay Tolerant Networking, mobile Internet and multicast.