Federated Service Chaining: Architecture and Challenges

Lin Cui, *Member, IEEE,* Fung Po Tso, *Senior Member, IEEE,* and Weijia Jia, *Fellow, IEEE*

***Abstract*—Emerging edge computing has seen latency sensitive services moving rapidly from cloud to the edge to take advantage of its close vicinity to end users, while cloud is retained for carrying out latency insensitive and computation intensive tasks. When more edge computing service providers come to the market, the network will become increasingly more fragmented because of their proprietary services and policies deployed in the network. This means the Internet can become more cumbersome and riskier as there will be more tiers and potential vulnerabilities that could be exploited.**

To tackle this issue, we envisage a federated service chaining paradigm in which operators can share and put service functions in other participants' networks so as to improve resource utilisation, collaboratively mitigate cyber threats and enable service innovations. In this position paper, building on our past experience in enabling federated cloud infrastructure and heterogeneous service chaining, we present a Federated Service Chaining (FSC) architecture followed by discussions of its key components. Several key research challenges are described for the successful realisation of such architecture. We hope this paper can open a discussion in research community and generate enough research interest to significantly advance this field.

***Index Terms*—Federation, service chain, cloud, edge computing.**

I. Introduction

In recent years, due to the advancement of computing technologies, e.g., cloud computing and edge computing, most applications service providers (ASPs), which provide application functionality and associated services to users, tend to deploy and deliver their services across distributed network domains, e.g., data centers (DCs), internet service provider (ISP) networks and edge clouds, to improve service responsiveness. In fact, end-user experience depends on a collection of those networks working together through interoperation and resource sharing, for example, networking access bandwidth (e.g., ISP peering), storage and computing service (e.g., CDN). This is becoming increasingly more important as we have witnessed a remarkable shift of computation from centralised clouds to distributed network edges. However, the emphasis is in fact on the cloud-edge collaboration in which less resource demanding tasks are performed on the edge whilst computation intensive tasks are carried out in the cloud due to its vast amount of computation power and network bandwidth.

Lin Cui is with the Department of Computer Science, Jinan University, Guangzhou, China

Fung Po Tso is the corresponding author and is with the Department of Computer Science, Loughborough University, UK, LE11 3TU. Email: p.tso@lboro.ac.uk

Weijia Jia is with the State Key Laboratory of Internet of Things for Smart City, FST, University of Macau, Macau SAR, China.

In the meantime, network stakeholders, e.g., ISPs, carriers, edge cloud providers (ECPs) and cloud DCs, often need to implement complex network policies that match their business objectives, such as traffic engineering, quality of service (QoS) and security. Those policies are usually composed by a sequence of various service functions (SFs) called service chain (SC), e.g., firewall, video transcoding, caching and monitoring [1]. Distinct business objectives of different networks mean that they have various policy requirements and configurations, which have crucial impact on the performance and security of each network [2]. To deliver efficient and reliable services across multiple network domains to users, these stakeholders should cooperate to provide consistent and improved policies and services. Figure 1 in Section II-C illustrates an example scenario for such purpose.

However, existing works mainly focus on service chaining within individual networks, e.g., SCs for cloud DCs [3] or edge networks [4]. In such scenarios, all SFs and SCs are only concerned by a single stakeholder and the deployment and management of SCs can be easily achieved. Different stakeholders across multiple network domains make it much more challengeable. Furthermore, current underlying network models are usually static, network-dependent and lack of auto-configuration. This imposes significant limitations and challenges on the handling of the rapid change of user traffic patterns as a result of fast service innovations. Although software defined networking (SDN) and network function virtualization (NFV) enable the flexibility to deploy internal policies efficiently, there is still no suitable mechanism at the moment for providing consistent policies across different networks.

As a result, it is important to study and develop techniques to federated service chains with an integrated interoperation layer to deploy applications securely and efficiently across federated network domains so as to improve both QoS, resource utilisation, and network security. Our previous investigations into service chaining in DCs [2] and mobile edge environments [5] have demonstrated that both resource utilisation and latency can be greatly improved when heterogeneity of servers and network devices is considered. Such heterogeneity will multiply significantly across multiple network domains. From economy's point of view, we have already seen the success of cloud federation whose global cloud service brokerage market size was valued at USD 4.96 billion in 2018 and is predicted to expand at a Compound annual growth rate (CAGR) of 17.3% from 2019 to 2025 [6]. *Thus, our position is that effective federation of networking policy enforcement across multiple network domains is fundamental for the deployment of future*

TABLE I: Comparisons of traditional and federated SC

	Traditional SC	Federated SC
Stakeholders	Single	Multiple
Complexity	Controller has full control	Complex, limited access to other domains
Consistency	Easy to achieve	Difficult, needs interoperation
Flexibility	Limited to each domain	Can offload some SFs to other domains
Services diversity	Limited	More diversity
Troubleshooting	Easy	Difficult

Internet applications.

In light of this position, in this paper, we conceptualize Federated Service Chaining (FSC) which extends traditional service chain to cross multiple network domains. A use case is described and analyzed to demonstrate the importance of being able to federate service chains across networks to meet specific security and performance requirements of applications. Then, we propose a reference architecture for FSC and investigate essential components and issues for federation based on our past experience in building federated micro cloud infrastructure [7][8]. Finally, we identify several research directions in FSC. To the best of our knowledge, this is the first study on federation for service chaining.

II. Federated Service Chaining

A. FSC Definition

Many applications and services need to be delivered across multiple networks. Each network, which is also referred as domain, belongs to one stakeholder. We define deploying policies with SFs across a federation formed by two or more domains as *Federated Service Chaining* (FSC). These SFs can be abstracted functions provided by each domain through technologies like SaaS (Software as a Service) [9], or user-defined functions based on VMs or containers. We also note that network functions and service chains are usually referred as service functions and service function chains (SFC) respectively, in IETF documents [1][10]. In this paper, these terms are used interchangeably.

Since multiple stakeholders are involved, we assume that each domain has a controller, which is responsible to coordinate with other networks in FSC. The whole federated service chain includes two levels of chaining: *Intra-domain service chaining* refers to chaining of SFs that are available within the same domain, whereas *Inter-domain service chaining* is the chaining of SFs that reside in two or more distinct domains.

B. Motivations and Benefits

The main benefits of FSC includes:

- **SLA (Service Level Agreements) assurance:** FSC will let all domains provide consistent and efficient service to users, e.g., consistent priority, latency and bandwidth throughout the whole FSC path.
- **Cost effectiveness:** FSC allows each network to share/rent unused resource to reduce cost. Also, if SCs are deployed in each domain independently, duplicated operations and functions are inevitable. With coordination of FSC, redundant functions can be eliminated within FSC.
- **Security:** FSC enables coordination among domains ensuring deployment of consistent policies. Also, by deploying SFs in edge, unwanted traffic can be blocked at the early stage of the chain in up-stream providers' networks.
- **Flexibility and Innovation:** Through inter-domain SCs across multiple domains, new applications and services can be designed and deployed easily. For example, functions for prepossessing traffic can be deployed on edge clouds dynamically to utilize distributed edge resource to improve performance.

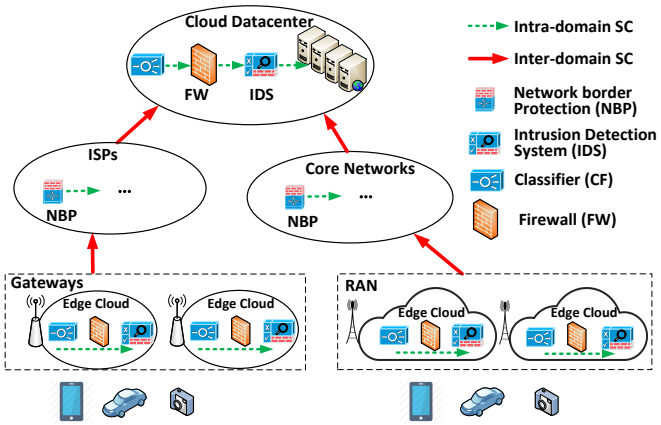

Fig. 1: Use case: Deploying micro services in edge

C. Use case: Deploying micro services in the edge

We demonstrate benefits of FSC through a use case in Figure 1. The introductions of ubiquitous Internet of Things (IoT) and edge computing are fundamentally changing the computational landscape while raising significant cyber security concerns. There have been incidents where IoT devices have been compromised to launch large scale attacks that brought down well-known Internet services [11].

Traditional means for mitigating these cyber threats are usually costly. In blackholing, upstream providers must be explicitly informed (e.g. by phone calls) to create policies that drop malicious packets. Local mitigation often employs SFs for, e.g., shaping of incoming traffic, rebalancing of traffic by manipulation of anycast policies and application of internal filtering but they are very costly. With FSC, ASPs can deploy policies at network edges to detect and push back attacks:

Service innovation on collaborative defense: Ideally cyber threats should be stopped at network edge. This is achievable because edge devices have abundant computation capability. Should malicious activities take place, light-weight edge-based detection and protection SFs running on these devices can promptly, in collaboration with their adjacent and remote peers, detect Botnet infection, alert attacks, and subsequently terminate malicious traffic at early stage before it can cause harm to backbone services.

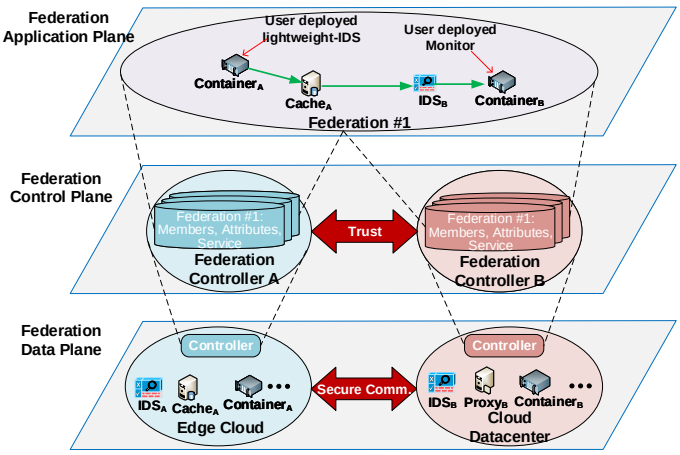

Fig. 2: Three plane illustration of FSC

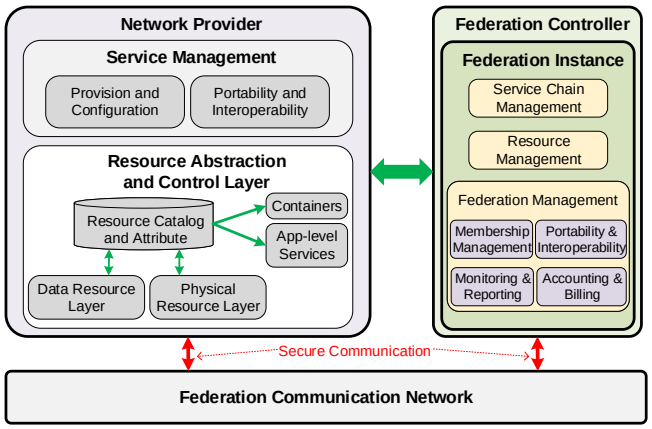

Fig. 3: FSC architecture design

In FSC, functions located at network edge can be dynamically instructed to run intrusion detection systems (IDS) to find attack patterns and signatures. In the events of attack, a dynamic SC can be set up in specific providers' edge networks to force suspected packets to traverse through firewall filtering and IDS. Such collaboration can effectively terminate malicious traffic from the edge of networks. Such idea can be easily extended to accommodate any types of innovations on the cross-domain computation such as edge caching and edge data analytics.

SLA assurance: More specifically in this use case the target end systems and upstream network providers' links would otherwise be too saturated to serve legitimate users if the attacks are not stopped in the early stage. Thus, it is clear that the FSC can provide better SLA assurance through resource sharing across providers.

D. Discussions

FSC extends traditional SC but they are significantly different as shown in Table I. Traditional SC normally operates in silos, deployed within a single network. FSC is deployed across multiple domains from cloud DC to edge networks. ASPs can utilize the heterogeneity of different networks to carry out varied services to safeguard and improve application performance. Since multiple stakeholders are involved, the management of FSC is much more challenging.

Similar to FSC, federated cloud, which has strong industry demand recently, enables efficient and secure deployment of resources and services across distributed cloud infrastructures [12]. For example, BEACON [13] enables provision of federated cloud infrastructures over different cloud management platforms. Those works offers great insights for designing a federation architecture of FSC. Compared with BEACON, FSC further considers more heterogeneous networks other than cloud DCs, e.g., edge networks, with special emphasis on deployment of SFs and supporting mobility. Moreover, FSC allows federated network to share virtualized services in addition to sharing of computation, storage and networking resources as commonly seen in federated cloud.

III. FSC Architecture

A. FSC Three-plane View

Figure 2 provides an overview of the three-plane illustration of FSC, demonstrating a federated service chain deployed across an ECP and a cloud DC, which includes federation data plane, control plane and application plane.

The *federation data plane* consists of all resources that are eligible to be shared through federation. Those resources can be at any abstraction level in the system stack, ranging from instances of VMs/containers (for user deployed functions) to arbitrary application-level functions (e.g,. IDS). Each domain has at least one *Federation Controller* (FC), which forms the *federation control plane*. The federation controller is responsible to provide necessary federation functions and interfaces. It can be organized in varied manner, e.g., peer-to-peer or hierarchical deployment (see Figure 4). Different implementation approaches may have different issues concerning consistency, communication latency, etc. Trust relationship should be established among federation controllers based on the secure communication among different domains. Then federation controllers of two or more domains can create a common federation, e.g., "Federation #1" in Figure 2. All federation controllers involved should maintain a consistent state for this federation over its lifetime. Once a federation has been created, federated service chain can be deployed across all domains in the *federation application plane*, which provides interfaces to users.

B. FSC Architecture

Figure 3 depicts a high-level view of the components necessary to carry out FSC. A federation is composed of network entities that can be widely geographically dispersed. They can be connected through the federation communication network with both authentication and encryption.

The *Network Provider* maintains all resources and includes two main components: *Service Management* and *Resource Abstraction and Control Layer*.

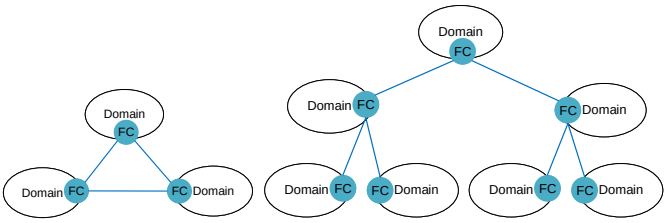

Fig. 4: Peer-to-peer and hierarchical deployment

The *Service Management* provides interfaces with the federation controller and is broken down into *Provision and Configuration* and *Portability and Interoperability* functions. *Provision and Configuration* allows automatically deploying SFs based on the requested service and resources, adjusting configurations, discovering and monitoring resources, and SLA enforcement according to defined rules. *Portability and Interoperability* allows customers to move/use their data and SFs across multiple domains with a unified management interface.

Resource Abstraction and Control Layer manages a catalog for a set of resources and their attributes which describes how do they exposed and accessed in a federation. These resources can not only be infrastructure resources, e.g., VMs/containers, but also arbitrary application-level services, e.g., instances of SFs. Furthermore, resources to be accessed can be either physical resources or data resources.

Each *Federation Controller* is comprised of three components, i.e., *Service Chain Management*, *Resource Management* and *Federation Management*, which can support multiple *Federation Instances*, or simply federations. Details of those components are explained in the following.

IV. Federation Instance

A. Service Chain Management

A federation may observe a number of SCs that would be deployed and scheduled across multiple domains.

1) Service composition: A network service is delivered using one or more SFs. The composition of inter-domain SCs are usually high-level and abstract, determined by each ASP according to their requirements. For example, an ASP may decide to deploy firewalls in the edge to filter malicious traffic close the source, and deploy a cache in regional ISPs to reduce latency of users. On the other hand, intra-domain SCs composition is handled by controller of each domain, e.g., edge cloud. To implement the service required by ASP, controller should select appropriate existing instances of functions or create new instances at appropriate locations. Federation Controllers can employ their own policies during composition, e.g., multiple instances for load balancing or adding a rate limiter along with the cache.

2) Service networking and routing: The service chaining is concerned with stitching SFs in correct logical order to implement desired network services. This involves placing SFs in the most suitable place in the federated environment. Hence, we advocate the idea of inter-domain chaining and intra domain chaining respectively. Inter-domain chaining refers to stitching SFs across different domains. Whereas intra-domain chaining means composing the chain within a specific domain. The general principle for FSC is that nodes with better computational resource should be shared as much as possible, and the functions that are available in the Edge and IoT environments should be as distributed as possible. When SCs are successfully composed, it is important to steer policy-bounded traffic flows through them. The federation communication network should be responsible for allocating network resources to manage the federated virtual network and overlay networks across distributed dispersed domains. Existing mechanism such as Generic Routing Encapsulation (GRE) or Virtual eXtensible Local Area Network (VXLAN), and Segment Routing may be used for steering traffic over service chains in intra domain chaining. However inter domain will need more careful consideration and more complex techniques, depending on types of federated networks, such as GRE, Geneve or other tunneling protocols. The federation communication network enables all domains to be able to communicate with each other when they belong to the same federation.

3) Service chain deployment: Deploying specific functions onto the network is not only subject to the constraint of underlying physical resources, but also other operational objectives such as service latency. For example, if both operator A and B have sufficient resource to accommodate operator X's request for a firewall. Since the latency to operator B's network is smaller, it will favour operator B's network for placing this firewall. Hence, the actual SC deployment is a combinatorial optimization in relation to physical resources. More specifically, two types of chaining are involved:

- *Intra-Domain Service Chaining*: Intra-domain service chains are private/local service chain which are directly deployed by the controller of current domain according to its own policy requirement.
- *Inter-Domain Service Chaining*: Inter-domain service chains maintain the global chaining between two endpoints. Since multiple domains are involved, a common negotiation scheme is required to exchange policy configurations/requirements, routing and reachability information across domains.

B. Resource Management

1) Service availability and discovery: We envision a highly dynamic service demand in the FSC as users and things that demand services flow and ebb quickly. Being able to make available and discover services or resources provided by federation members are vital to the federation. Once a federation is formed, members will decide what service they would like to share, and how would they like to share them. This means that members who own the services will have ultimate control on the shared services according to individual member's access level, including rejecting requests from other members in some extreme circumstances.

Regarding the shared resource, it is important for federation members to specify the types and amount of resources to be

shared because there are some subtle but non-trivial difference among them. For example, if service function boxes (e.g., containers) are exposed, other members can use them host almost any types of SFs as they are virtualized servers, whereas if only service functions are exposed they can only be used to for very specific purpose. This implies a need for being able to describe the nature of shared services and hence agree upon some descriptions that are commonly understood across the federation.

Once these are established, traditional mechanisms for service cataloging and discovery can be applied, e.g., Lightweight Directory Access Protocol (LDAP) and OWL-S.

2) Service mobility and migration: Demand on different services can change over time as a result of the ebb and flow of user traffic on the Internet. For example, large events such as music festivals often create Internet service hotspots in specific areas. One fundamental requirement therefore is to adapt the federated SFs with the dynamism of user traffic. This involves detecting the changes of user demands and migrating SFs or the whole SC with an aim to maintain satisfactory SLA requirements. Once triggered, the SC in question will need to be re-scheduled for the determination of the new best composition. Nevertheless, there is also a need to employ one or more suitable mechanisms to ensure that the predicted benefit will outweigh the cost of migration, hence avoid oscillations.

C. Federation Management

Membership Management keeps track of active participants of the federation with both federated identity credentials and attributes.

Monitoring & Reporting are basic functions that support other components. This includes resource usage, performance, health status and so on.

Accounting & Billing is used to track resource usage by federation members, which may need to be associated with pricing or cost schedule.

Data portability is needed to enable members to access and retrieve data with reasonable cost and format. Similarly, *system portability* allows images (VMs or containers) and SFs to be able to be moved among federations partners. In the meantime, different federation domains should have a unified management interface, or a middleware that presents a more unified API to achieve better service interoperability.

V. FSC CHALLENGES

A. Highly heterogeneous environment

With dramatically increased heterogeneity in terms of computing and networking resources across different domains, especially in edge and IoT environments, achieving effective and efficient SCs over heterogeneous networks in federation environment is very challenging. *One fundamental challenge is how can resource be represented and understood, such that a proper decision can be made*. Even if a member makes resources available within a federation, portability and interoperability would be needed for members to access and deploy services with reasonable cost.

Some challenges concern heterogeneous environments include:

- Effective sharing of resources and services, e.g., representation of different types of resource/service, on-demand sharing and retrieval, real-time monitoring, enforcement of user-defined policy, etc.
- Efficient federated SC deployment and scheduling, including inter-domain SC and intra-domain SC deployment.
- Network construction establishment across domains, e.g., overlay network for inter-domain SC.
- Interaction with other services and applications. For example, how to leverage federated clouds, which mainly focus on integration of different cloud platforms, to enforce FSC across cloud service providers.

B. FSC validation and verification

When SCs are deployed in a federation, it is important to verify whether a particular SC is correctly deployed and enforced as desired to avoid any policy violations. Verifying a SC is a complicated process, especially in federated environment with multiple stakeholders. A federation controller normally has limited visibility on other members networks whilst existing solutions for cloud federation assume complete network visibility. Hence, mechanisms are needed for effectively probing and validating federated SC. Such operations must be able to be easily implemented by all federation members. We have previously studied verification and validation of service chaining in SDN environment [14]. Our experience suggests that, for FSC, a more feasible approach is to create a passive probing (black-box) scheme that is less intrusive.

Challenges about verification and performance measurement of FSC mainly concerns:

- SFs availability check, e.g., how to characterize availability of SFs under federated environment? How to design of fine-grained mechanisms for functionality check of SF instances?
- SFs performance measurement, e.g., passive measurement using live traffic or active measurement with synthetic probe packets.
- Federated service chain availability check and performance measurement, e.g., FSC forwarding path validation on both inter-domain SC and intra-domain SC, policy violation detection, performance measurements over the entire FSC or a specific segment within the FSC.
- Proactively check of alternate paths and service nodes to ensure FSC reliability when ECMP is in use.

C. Security and trust

Different from cloud federation, the biggest challenge of FSC is to enable trust. The cloud federation brings together established businesses that can be trusted partly by their business reputation. However, FSC contains a lot of small operators at the edge of the networks that demand and provide services. They may not have the business reputation needed to be considered trustworthy.

While FSC can be used to improve security for the whole Internet, necessary mechanisms must also be established to defend the federation against attack. One of the most prominent is to prevent the federation from insider attack. This includes a member intentionally removing the firewall or IDS that is required for the downstream members, letting through unfiltered traffic onto the downstream members' unprotected network.

Concerns regarding trust and secure FSC includes:

- How to go beyond the minimum common denominator of services across domains to enable richer FSC applications and services, e.g., encouraging more ESPs to join FSC.
- How to establish and coordinate trust relationship, authorization, access control and billing across different network stakeholders, including transitivity and delegation of trust through federation.
- How to establish a secure communication channel for both inter-domain and intra-domain.

D. Traffic flows classification

An incoming traffic flow should be classified according to classification rules to determine which federation it belongs to and which service chain of the federation will handle it. The classification is based on the content of one or more packet header fields so that the classification rule may vary in different granularity. This may bring a problematic case in which an incoming packet matches two or more classification rules of different SCs or federations, which can result in incorrect operations on flows.

Another is the billing mechanism. In cloud federation, it is easy to track the source and bill the upstream providers for the services. However, for FSC, the immediate upstream provider may not be the source of the request, hence making both metering and billing difficult.

Moreover, federation brings additional challenges to the classification in which two similar but isolated SCs may exist. For example, if two service chains that belong to two different federations have installed rules to filter traffic destining to the same destination but with contradicting actions (e.g., Drop vs Allow), the classification system will be unable to correctly classify incoming traffic flows on to the correct service chaining that govern the traffic.

Major challenges in this area includes:

- Fine-grained classification schemes for inter-domain SC and intra-domain SC respectively.
- Effective violation detection mechanisms for both inter-domain SC and intra-domain SC.
- Flexible classification for accounting and billing to track resource usage of federation members.

E. Migration and state consistency

The dynamic nature of the Internet means the services running atop change frequently. On the other hand, the underlying physical resources, Edge and IoT in particular, can join and/or leave federations unexpectedly. This dictates the needs for dynamically migrating individual SFs to better utilize physical resources (for improving the return on investment). For this to take place efficiently, the first challenge is that the controller will need to have very timely and accurately view of the resource availability across the federation, and the service chaining algorithms need to be adaptive. The second challenge is to ensure consistent state after migrating any stateful SFs as incorrect states can easily lead to policy violations.

Major challenges in this area include:

- SFs migration within and across network domains, e.g., algorithms to determine when and where to migrate a SF, migration schemes to ensure non-interruption of services.
- Network update for SF migration, e.g., network reconfiguration for migration within a domain and overlay network coordination for inter-domain SC.
- Efficient state migration to ensure consistency, e.g., consistency levels characterization (i.e., strong or weak), inconsistency detection and recovery.
- Fault tolerance mechanisms for FSC reliability, e.g., passive or active replication of SF instances and federation controllers, topology-independent loop-free alternate path for both inter-domain and intra-domain SC.

VI. Conclusions

In this paper, we propose a new networking paradigm called Federated Service Chaining, which focuses on cross networks collaboration in harnessing computation resources for greater network wide resource utilisation, enforcing functions and promoting service innovation. We present design of the FSC architecture, as well as its major components. Since realization of FSC architecture is not a straightforward task, we also lay out the most prominent research challenges that we can foresee. We hope that this paper will inspire more further research works in the field of federated service chaining.

Acknowledgement

This work is partially supported by Natural Science Foundation of China (NSFC) No. 61772235, 61532013 and 61872239; 0007/2018/A1, 0060/2019/A1, DCT-MoST Joint-project No. 025/2015/AMJ of Science and Technology Development Fund, Macao SAR (FDCT); University of Macau Grant: MYRG2018-00237-FST, CPG2020-00015-IOTSC and SRG2018-00111-FST; the UK Engineering and Physical Sciences Research Council (EPSRC) grants EP/P004407/2 and EP/P004024/1.

References

[1] P. Quinn and T. Nadeau, "Problem Statement for Service Function Chaining," RFC 7498, Apr. 2015, Accessed on: Jan. 19, 2020. [Online]. Available: https://rfc-editor.org/rfc/rfc7498.txt

[2] L. Cui, F. P. Tso, S. Guo, W. Jia, K. Wei, and W. Zhao, "Enabling heterogeneous network function chaining," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 842–854, 2019.

[3] M. T. Surendra, S. Majee, C. Captari, and S. Homma, "Service function chaining use cases in data centers," *Working Draft, IETF Secretariat, Internet-Draft draft-ietf-sfcdc-use-cases-06, January*, 2017.

[4] H. Huang and S. Guo, "Proactive failure recovery for nfv in distributed edge computing," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 131–137, 2019.

[5] T. A. Genez, F. P. Tso, and L. Cui, "Latency-aware joint virtual machine and policy consolidation for mobile edge computing," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–6.
[6] G. V. Research. Cloud service brokerage (CSB) market size, share & trends analysis report by service, by deployment model, by organization size, by platform, by end-use industry, by region, and segment forecasts, 2019 - 2025. Accessed on: Jan. 19, 2020. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/cloud-service-brokerage-csb-market
[7] S. J. Johnston, P. J. Basford, C. S. Perkins, H. Herry, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, S. J. Cox, and J. Singer, "Commodity single board computer clusters and their applications," *Future Generation Computer Systems*, vol. 89, pp. 201–212, 2018.
[8] J. Singer, H. Herry, P. J. Basford, W. Hajji, C. S. Perkins, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, S. J. Cox *et al.*, "Next generation single board clusters," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*. IEEE, 2018, pp. 1–3.
[9] E. Loukis, M. Janssen, and I. Mintchev, "Determinants of software-as-a-service benefits and impact on firm performance," *Decision Support Systems*, vol. 117, pp. 38–47, 2019.
[10] J. M. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," RFC 7665, Oct. 2015, Accessed on: Jan. 19, 2020. [Online]. Available: https://rfc-editor.org/rfc/rfc7665.txt
[11] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
[12] R. Buyya, S. N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L. M. Vaquero, M. A. Netto *et al.*, "A manifesto for future generation cloud computing: research directions for the next decade," *ACM Computing Surveys*, vol. 51, no. 5, p. 105, 2019.
[13] E. Huedo, R. S. Montero, R. Moreno, I. M. Llorente, A. Levin, and P. Massonet, "Interoperable federated cloud networking," *IEEE Internet Computing*, vol. 21, no. 5, pp. 54–59, 2017.
[14] Y. Zhang, L. Cui, F. P. Tso, and Y. Zhang, "Track: Tracerouting in SDN networks with arbitrary network functions," in *6th International Conference on Cloud Networking (CloudNet)*. IEEE, 2017, pp. 1–6.

Lin Cui is currently with the Department of Computer Science at Jinan University, Guangzhou, China. He received the Ph.D. degree from City University of Hong Kong. He has broad interests in networking systems, with focuses on the following topics: cloud data center resource management, data center networking, software defined networking, network function virtualization, congestion control and so on.

Fung Po Tso He is currently a senior lecturer in the Department of Computer Science at the Loughborough University. He received BEng, MPhil and PhD degrees from City University of Hong Kong in 2006, 2007 and 2011 respectively. His research interests include: network policy management, network measurement and optimisation, cloud data centre resource management, data centre networking, SDN, distributed systems as well as mobile computing and system.

Weijia Jia is currently a Chair Professor, Deputy Director of State Kay Laboratory of Internet of Things for Smart City at the University of Macau. His research interests include smart city, IoT, knowledge graph constructions; multicast and anycast QoS routing protocols, wireless sensor networks and distributed systems. He has over 500 publications in the prestige international journals/conferences and research books and book chapters.

TABLE I: Comparisons of traditional and federated SC

	Traditional SC	**Federated SC**
Stakeholders	Single	Multiple
Complexity	Controller has full control	Complex, limited access to other domains
Consistency	Easy to achieve	Difficult, needs interoperation
Flexibility	Limited to each domain	Can offload some SFs to other domains
Services diversity	Limited	More diversity
Troubleshooting	Easy	Difficult

Figure 1: Use case: Deploying micro services in edge

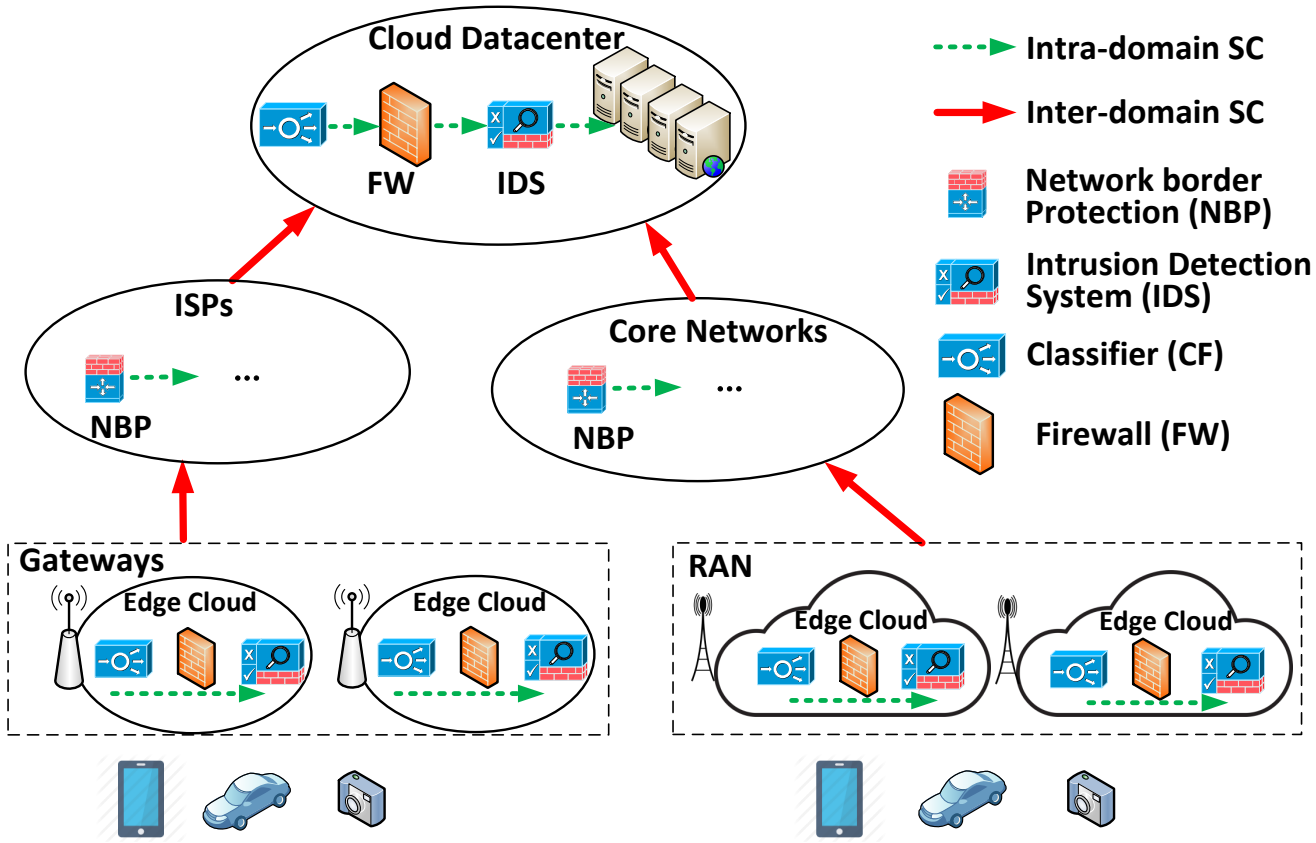

Figure 2: Three plane illustration of FSC

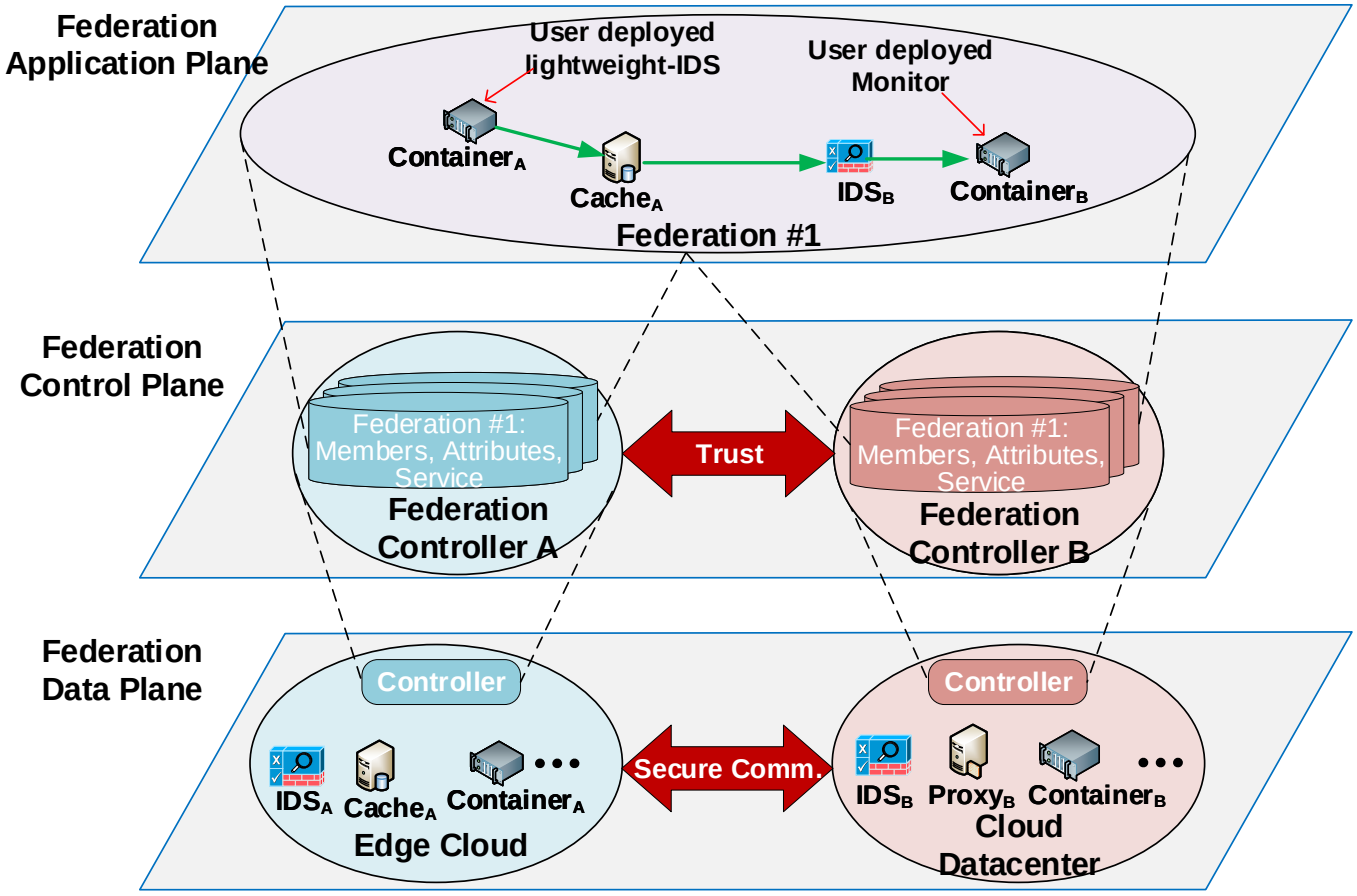

Figure 3: FSC architecture design

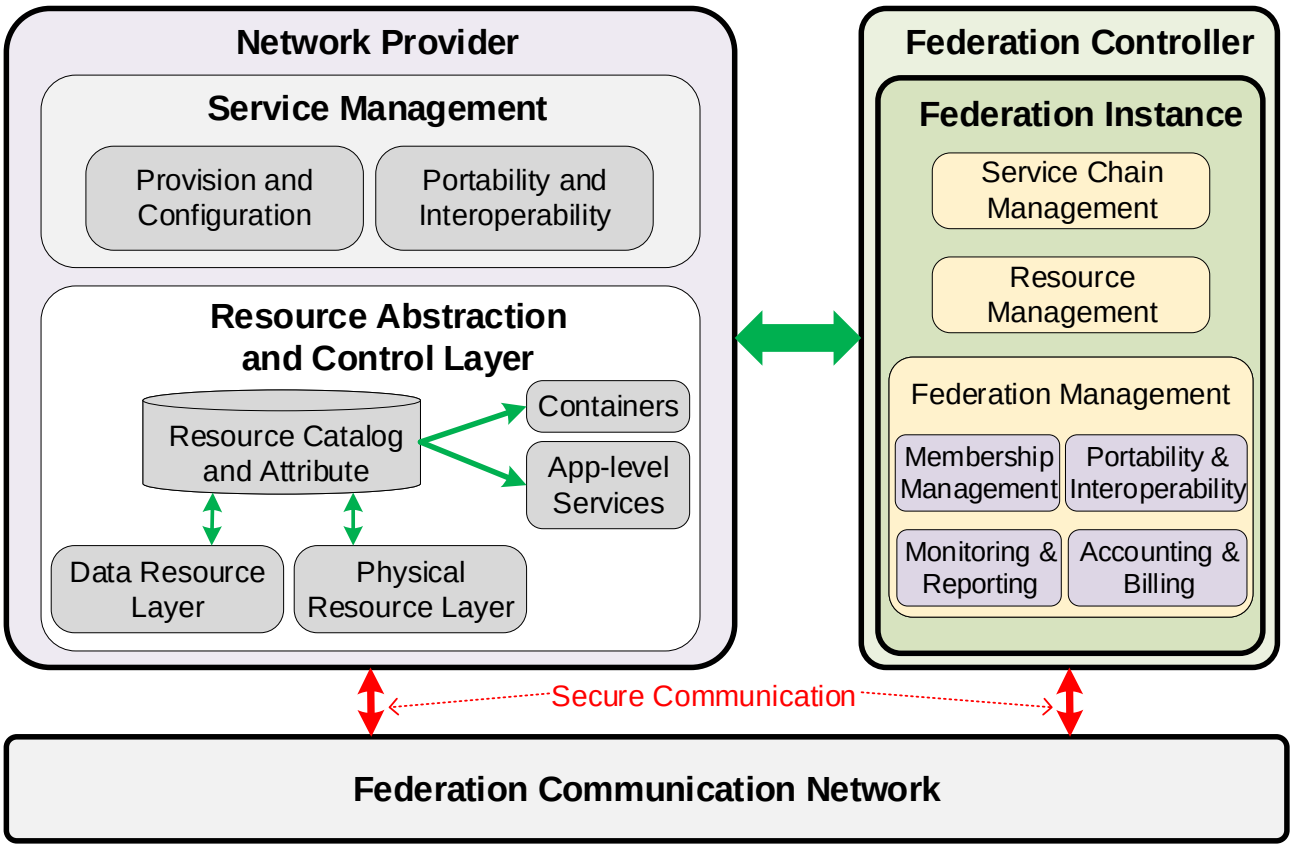

Figure 4: Peer-to-peer and hierarchical deployment
(a) Peer-to-peer

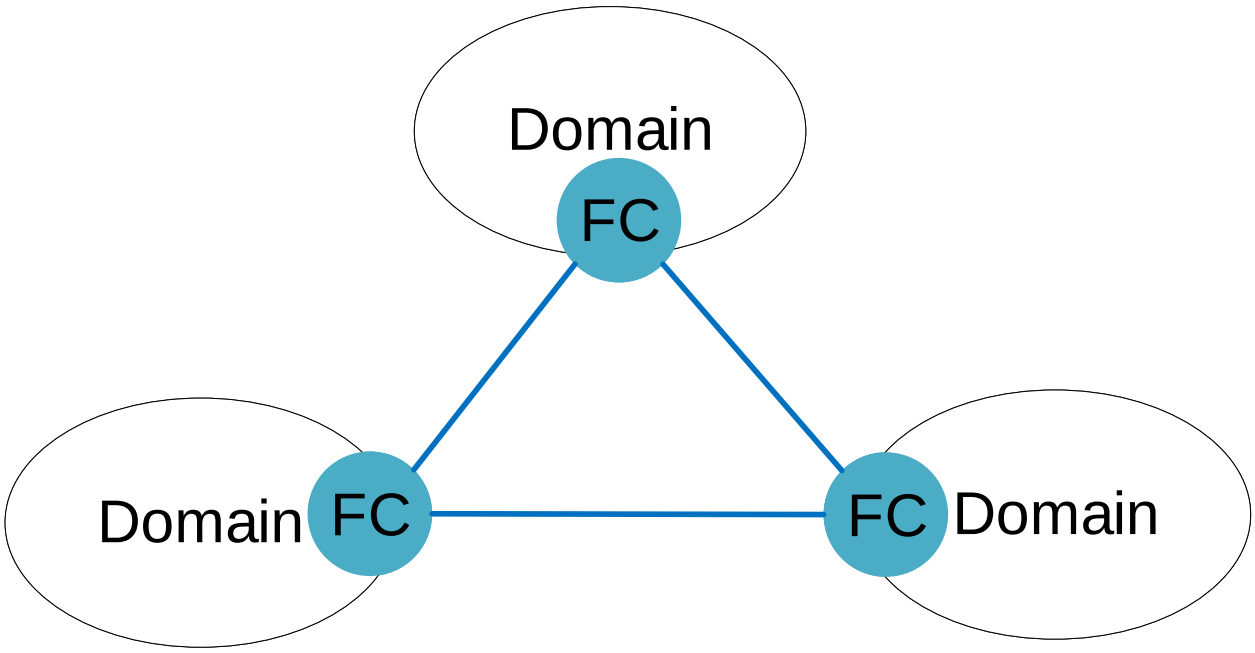

Figure 4: Peer-to-peer and hierarchical deployment
(b) Hierarchical