Secure and Intelligent Service Function Chain for Sustainable Services in Healthcare Cyber Physical Systems

Haotong Cao, *Member, IEEE*, Anish Jindal, *Member, IEEE*, Han Hu, Md. Jalil Piran, *Senior Member, IEEE*, and Longxiang Yang

***Abstract*—The gradual integration of Internet of Things (IoT) devices has made the healthcare cyber physical systems (CPSs) ubiquitous and complex. Since IoT devices are heterogeneous, it is crucial to virtualize these devices in a general way so as to orchestrate and allocate their equipped functions and resources in a general manner. Network function virtualization (NFV) is the key networking technology towards the device virtualization. Within NFV, service function chain (SFC) is the crucial issue. While in real networking environment, physical device accidentally fails. How to guarantee the SFC, deployed on top of one or more failed devices, recover to work has not been well addressed yet. Hence, we research this issue, aiming at realizing the secure and sustainable SFC placement and scheduling in an intelligent manner. An intelligent SFC placement and scheduling framework, labeled as *Sec-SFC-Intell*, is proposed in this paper. When receiving a SFC request, the *Sec-SFC-Intell* will deploy the SFC on top of the healthcare CPSs. When the failure of certain one physical device is detected, the affected SFC will be re-placed and re-scheduled in order to recover to run and serve. In order to prove the feasibility and highlight the merits of *Sec-SFC-Intell*, simulation work is conducted. Compared with two compared benchmarks, *Sec-SFC-Intell* has an apparent SFC acceptance advantage of nearly 10%.**

***Index Terms*—Cyber physical systems; IoT; healthcare; service function chain; device failure; sustainable services.**

I. Introduction

A. Backgrounds and Related Work:

CYBER physical systems (CPSs) [1] are multi-dimensional and complex systems by integrating heterogeneous devices and terminals. The healthcare [2] is one dominant aspect in the application of CPSs. By adopting the computation, communication and control technologies, the healthcare CPSs can provide the operators (doctors and hospitals) and users (patients and their relatives) with various traditional (e.g. remote consultation) and upgraded services (e.g. medical experience) [3](Fig. 1(a)). Hence, the healthcare CPSs have great potential in application area. The healthcare CPSs are characterized by dozens of Internet of Things (IoT) [4] devices. Since IoT devices are heterogeneous, it is necessary to adopt the effective and flexible networking technologies (Fig. 1(a)), aiming at integrating and managing these heterogeneous devices in a general manner. Emerging networking technologies, such as network function virtualization (NFV)[5] and software network defining (SDN) [6], contribute to virtualizing and slicing these heterogeneous devices. Hence, it is convenient to manage and allocate their virtualized functions and abstracted resources effectively. Especially, NFV is the key technology towards the entire virtualization. By adopting NFV, underlying IoT elements and devices, along with their equipped functions and resources, can be fully abstracted and operated by service providers (SPs) and operators.

In NFV concept, one crucial technical issue is the service function chain (SFC). The SFC issue consists of four sub-issues [7]: SFC description, SFC composition, SFC placement (embedding, mapping), and SFC scheduling. The first two sub-issues are mainly researched by the industry [7] while the last two sub-issues are mostly studied by the academia. With respect to the last two sub-issues, they concentrate on deploying the requested network service, modeled by SFC, and fulfilling function requests and resource demands of the service. Consequently, multiple algorithms for SFC placement and scheduling have been proposed and developed [5-18], such as the greedy method based algorithm [7], Markov random walk model based algorithm [8], multi-attributes based algorithm [9] and so on. In addition, deep learning (DP) approach, originated from the known machine learning (ML), has been attempted to allocate computing and networking resources of virtual SFCs in recent years [13][14]. Yao *et al.* [13] adopted the reinforcement learning (RL) to do the resource allocation of virtual services while ignoring the security aspect of virtual SFC service. However, in the real application scenarios, SFCs usually are requested and generated dynamically and unexpectedly. SFCs are required to be allocated and deployed within

The work of this paper was partly supported by National Natural Science Foundation of China under Grant 62071246 and 92067201, Jiangsu Provincial Key Research and Development Program under Grant BE2020084-5 and BE2020084-1. (*Corresponding author: Longxiang Yang.*)

Haotong Cao is with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR 999077, China. Email: haotong.cao@polyu.edu.hk.

Anish Jindal is with Department of Computer Science, Durham University DH1 3LE, United Kingdom. Email: anish.jindal@durham.ac.uk.

Han Hu and Longxiang Yang are with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. Email: han_h@njupt.edu.cn, yanglx@njupt.edu.cn.

Md. Jalil Piran is with the Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea. Email: piran@sejong.ac.kr.

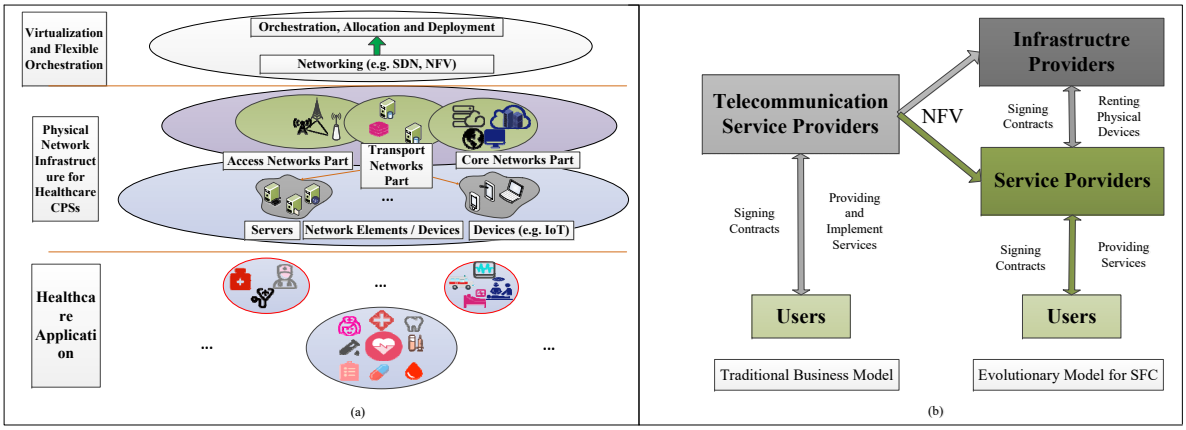

Fig. 1: Overview of Integrating NFV Into Healthcare CPSs and Evolution of Major Business Roles in SFC Research

limited time. The DP-based algorithm consumes too much time to train and test so as to achieve the efficient allocation solution per SFC. The situation where SFC expires while the DP-based algorithm continues to train the optimal mapping and scheduling solution may happen. Thus, the intelligent SFC mapping and scheduling algorithm, running within limited time, is essential to be developed for NFV-enabled healthcare CPSs. Though resource allocation research in CPSs [19-23] has been conducted, the NFV has not been adopted to CPSs yet. Due to the heterogeneity nature of physical elements in healthcare CPSs, security is strongly needed to be considered while doing the resource allocation. Selecting secure physical elements in CPSs to conduct the secure SFC placement and scheduling can guarantee the secure communication, information exchange and protection of valuable information. In addition, previous researches [4][6][7] are based on the assumption that no IoT device fails throughout the whole time. This assumption does not exist in the real networking environment [8]. Physical IoT elements, no matter nodes or connected links, may fail from time to time. How to guarantee the deployed SFC continued and sustainable has not been well addressed yet. Though some relevant publications (e.g. [24][25]) come out in recent years, they have two main flaws: 1) only consider the failure of one physical device; 2) simply consider the node resource and ignore function requests.

With the goal of providing intelligent, secure and sustainable placement of SFCs and removing existing limitations in NFV-enabled CPSs [7][8][9], we propose one novel and intelligent SFC placement and scheduling framework in this paper. The proposed SFC framework is abbreviated as *Sec-SFC-Intell*. When serving one requested SFC, the *Sec-SFC-Intell* framework firstly deploys the requested SFC to suitable physical elements in the intelligent manner. The selected physical elements must be equipped with abundant functions, resources and security demands. Meanwhile, the *Sec-SFC-Intell* will monitor and check the states of all underlying physical devices. When certain one or more failures of physical devices are detected, *Sec-SFC-Intell* will isolate the failed elements and their directly connected elements from the whole underlying elements. Meanwhile, the *Sec-SFC-Intell* will re-place and re-schedule the affected nodes and links of SFC so as to recover the SFC to serve. Hence, the sustainability of affected SFC is achieved. In order to validate the merits of *Sec-SFC-Intell*, the simulation work is made in this paper. Two basic and mostly related SFC placement and scheduling frameworks are derived and selected for performance comparison. Simulation results are plotted and discussed in order to prove the merits of *Sec-SFC-Intell*. For example, *Sec-SFC-Intell* has an apparent acceptance advantage of nearly 10%, comparing with two selected frameworks.

B. Main Contributions:

1) The business model of SFC research is introduced in this paper. System model of SFC placement and scheduling in healthcare CPSs, supporting virtualization scheme, is presented, too. An example of secure and sustainable SFC placement and scheduling is discussed, too. More resource types (node and link) and attributes are considered in the proposed system model, compared with traditional SFC placement model [7].

2) A novel and intelligent SFC placement and scheduling framework, abbreviated as *Sec-SFC-Intell*, is proposed in this paper. The *Sec-SFC-Intell* not only provides a secure and intelligent SFC placement solution, but also guarantees the deployed SFC recovered and continued to work even though certain one or more physical elements of CPSs fail accidentally. The *Sec-SFC-Intell* framework is designed to provide the secure and sustainable SFC services in an intelligent manner. Since SFC research in healthcare CPSs is in its early stage [2][4][6], we consider the general slices placement in this paper. Specific slices, such as remote medical services with rigid latency, will be further considered.

3) The simulation work is conducted so as to prove the merits of *Sec-SFC-Intell* framework. Within the simulation work, two mostly-related SFC placement and scheduling frameworks (*BasicOne-SFC*, *BasicTwo-SFC*) are derived and selected for performance comparison. Simulation results and key performance metrics are carefully illustrated.

C. Paper Organization:

The rest of this paper is organized as follow. Section II presents the business model, system model and performance metrics of SFC placement and scheduling in healthcare CPSs. The proposed *Sec-SFC-Intell* framework is detailed in Section III. In Section IV, the simulation work is provided, with the purpose of highlighting the merits of *Sec-SFC-Intell* framework. In the last section, the paper is concluded.

II. System Model of SFC Research

Three major subsections constitute this section. The first subsection introduces the major business roles in SFC research. The second subsection focuses on the SFC system model and the description of one secure and survivable SFC example. The third subsection introduces the main performance metrics that can be used to evaluate SFC.

A. Major Business Roles in SFC Research

From the point of traditional telecommunications market, two major business roles exist: telecommunication service providers (TSPs) and their contracted end users. Please refer to Fig. 1 (b). TSPs are responsible for managing, maintaining and upgrading all dedicated equipment and devices, such as various IoT devices, servers and routers. TSPs are responsible for providing and realizing various network services and applications, too. Users are responsible for signing contracts with TSPs and proposing service requests. Users pay off their service requests when TSPs realize these service requests.

With introducing NFV [3], underlying physical devices will be completely virtualized in a general way. Functions and resources of the dedicated devices can be easily managed in the form of blocks and slices. Hence, TSPs will be decoupled and evolved into another two detailed and isolated business roles: infrastructure providers (InPs) and SPs. Please refer to Fig. 1(b). InPs are responsible for constructing, managing, maintaining and upgrading underlying physical equipment while SPs are responsible for providing and realizing services, including the sustainable services. Within this business model, SPs will rent their requested physical equipment and elements and pay to InPs, according to their signed contracts. SPs earn benefits by deploying and providing services to their signed Users. In order to maximize the net benefits of SPs, SPs are required to realize requested services as many as possible. In addition, the realized services will be affected due to their placed physical elements' accidental failures, leading to SPs' penalty fees. Hence, it is vital to research the SFC with resilient and survivable awareness. If that, SPs' net benefits can be maximized. Thus, it is necessary to research the placement and scheduling of SFCs.

B. System Model for SFC Research and Sustainable Example

The system model of SFC for CPSs research is made up of two major entities: physical network model for CPSs and SFC model.

With respect to the physical network model, we use the undirected weighted graph $PhyNet = (PhyNodes, PhyLinks)$. $PhyNodes$ represents the set storing all physical nodes. These physical nodes all support the NFV scheme. Thus, their equipped network functions and resources can be fully virtualized and managed in the form of slice or block. SPs can rent them from the infrastructure providers and provide tailored services. With respect to certain one physical node M, its equipped functions, such as firewall, network address translation (NAT), can be virtualized. Thus, its virtualized functions can be labeled, such as $Func1(M)$, $Func2(M)$. In this paper, $Func1(\ )$ corresponds to firewall while $Func2(\ )$ corresponds to NAT. In total, four different functions are considered in this paper. With respect to abstracted resources, central processing unit (CPU), memory, and storage resources are considered in this article. They are labeled as $CPU(M)$, $Mem(M)$ and $Sto(M)$. Security probability of M is labeled as $Sec(M)$, representing the security ability of M against attacks. In addition, one dominant time attribute is consider: data processing delay $Delay(M)$. This attribute is vital as it reveals the data processing ability of M. With respect to certain one physical link MN, its accounted resource is bandwidth, labeled as $Band(MN)$.

Network services, represented and modeled by SFCs, are usually requested unexpectedly. With respect to the ith coming SFC, it is modeled by directed weighted graph $SFC(i) = (SFC(i)Nodes, SFC(i)Links)$. $SFC(i)Nodes$ is a set defined for storing all nodes of $SFC(i)$. $SFC(i)Links$ is a set for storing all chained links of $SFC(i)$. $|SFC(i)Nodes|$ represents the number of nodes in $SFC(i)$ while $|SFC(i)Links|$ represents the number of directed links in $SFC(i)$. For instance, the first node $SFC(i)Node(1)$ has the required firewall function, labeled as $Func1(SFC(i)Node(1))$. One node has one required function type. In this paper, four function types are considered. The required resource types of $SFC(i)Node(1)$ are CPU, memory and storage. Thus, these resources are labeled as $CPU(SFC(i)Node(1))$, $Mem(SFC(i)Node(1))$ and $Sto(SFC(i)Node(1))$. The required security probability of $SFC(i)Node(1)$ is labeled as $Sec(SFC(i)Node(1))$. In addition, the required data processing delay of $SFC(i)Node(1)$ is labeled as $Delay(SFC(i)Node(1))$. With respect to first physical link $SFC(i)Node(1)SFC(i)Node(2)$, its required bandwidth resource is labeled as $Band(SFC(i)Node(1)SFC(i)Node(2))$. Take note that $SFC(i)$ represents the general service model. When representing the remote end-to-end inquiry, the $SFC(i)$ will request high security probability, high node resource and bandwidth demands. When representing the remote surgery, $SFC(i)$ must be equipped with rigid processing delay, high security probability, and abundant bandwidth resources. According to goals of different services, concrete requests are different. In the SFC modeling of this paper, we extract the commonality of network services.

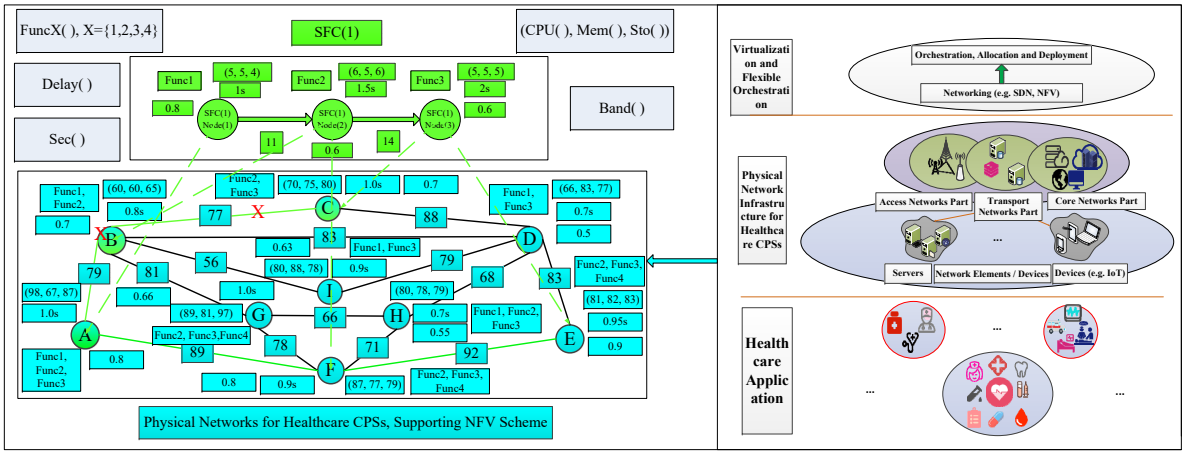

Fig. 2: SFC System Model for Healthcare CPSs and One Secure and Sustainable SFC Example

With the goal of helping to understand the system model, we plot Fig. 2. Within Fig. 2, one physical network and one SFC are included. Their function, resource, security and time attributes are labeled and highlighted. In addition, the secure and sustainable example of $SFC(1)$ is plotted and highlighted in Fig. 2. We aim at helping readers to understand the physical element failures and sustainable SFC re-placement well.

At first, it is the initial SFC placement and scheduling. Derived from Fig. 2, the initial placement and scheduling results of SFC are: $SFC(1)Node(1)$ is placed onto physical node A, $SFC(1)Node(2)$ is placed onto physical node B, $SFC(1)Node(3)$ is placed onto physical node C. Meanwhile, virtual link $SFC(1)Node(1)SFC(1)Node(2)$ and $SFC(1)Node(2)SFC(1)Node(3)$ are placed onto physical path AB and BC, respectively. In addition, all required function, resource, security and time attributes of $SFC(1)$ are matched.

When the physical node B and physical path BC fails, the implemented $SFC(1)$ will be interrupted and stops serving its users. Thus, the re-placement and re-scheduling of the interrupted SFC must be done so as to make the SFC sustainable to serve. After doing the re-placement and re-scheduling, the $SFC(1)$ is recovered. The re-placed and re-scheduling results are: $SFC(1)Node(2)$ is placed onto physical node F, $SFC(1)Node(3)$ is placed onto physical node E. Meanwhile, virtual link $SFC(1)Node(1)SFC(1)Node(2)$ and $SFC(1)Node(2)SFC(1)Node(3)$ are placed onto physical path AF and FC, respectively.

As shown in the example, we aims at recovering the implemented SFC, no matter node or link failures. In addition, more than one element failures accounts while previous research [24][25] has not considered these yet.

C. Main Performance Metrics

In the study of SFC, three main types of performance metrics are involved: success type, resource consumption type and time type. With respect to the success type, SFC acceptance ratio, labeled as $SfcAR$, is usually adopted. $SfcAR$ has the function of revealing the placement and scheduling ability. $SfcAR$ is formulated in Equation (1).

$$\boldsymbol{SfcAR}(\,) = \frac{SuccessSfc(\,)}{TotalSfc(\,)} \tag{1}$$

where the name of the evaluated algorithm or framework is included in the brackets. $SuccessSfc(\,)$ is a variable indicating the number of successfully deployed SFCs while $TotalSfc(\,)$ is a variable. $TotalSfc(\,)$ has the function of recording the value of all requested SFCs.

With respect to the resource consumption type, they are CPU resource utilization, memory resource utilization, storage resource utilization and bandwidth resource utilization. We formulate four equations (Equation (2), Equation (3), Equation (4) and Equation (5)):

$$\boldsymbol{CPU}(\,) = \frac{ConsumedCPU(\,)}{TotalCPU(\,)} \tag{2}$$

where the evaluated node/device is labeled in the brackets. $ConsumedCPU(\,)$ has the function of recording all consumed CPU resource in the evaluated node/device while $TotalCPU(\,)$ has the function of recording all CPU resource in the evaluated node/device.

$$\boldsymbol{Memory}(\,) = \frac{ConsumedMem(\,)}{TotalMem(\,)} \tag{3}$$

where the evaluated node/device is included in the brackets. $ConsumedMem(\,)$ has the function of recording all consumed memory resource in the evaluated physical node while $TotalMem(\,)$ has the function of recording all memory resource in the evaluated node/device.

$$\boldsymbol{Storage}(\,) = \frac{ConsumedSto(\,)}{TotalSto(\,)} \tag{4}$$

where the evaluated node/device is included in the brackets. $ConsumedSto(\,)$ has the function of recording all consumed storage resource in the evaluated physical node while $TotalSto(\,)$ has the function of recording all storage resource in the evaluated node/device.

$$\boldsymbol{Band}(\ ) = \frac{ConsumedBand(\ )}{TotalBand(\ )} \tag{5}$$

where the evaluated physical link is included in the brackets. $ConsumedBand(\ )$ has the function of recording consumed bandwidth resource of the evaluated physical link while $TotalBand(\ )$ has the function of recording all bandwidth resource in the evaluated physical link.

With respect to the time type, the algorithm execution time, labeled as $ExTime(\ )$, is considered in this paper. The function of $ExTime(\ )$ is to record the consumed time of achieving the SFC deployment. The name of the evaluated algorithm is included in the brackets. With respect to other performance metrics, such as the throughput, we do not consider them in this paper. We focus on evaluating the allocating ability. If having the prototype, the throughput will be further researched.

III. Secure and Intelligent SFC Placement and Scheduling Framework *Sec-SFC-Intell* for Sustainable SFCs

This section focuses on describing the *Sec-SFC-Intell* framework, consisting of two modules: *Secure and Intelligent SFC Placement and Scheduling and Simultaneous Physical Elements Checking* module (subsection A) and *Re-Placement and Re-Scheduling of Interrupted SFC* module (subsection B). Take note that our *Sec-SFC-Intell* framework does not prepare one extra SFC placement and scheduling when serving the given SFC request. Two major reasons are for this: one is to leave more resource space for new SFCs, the other is to get rid of the effect of accident nodes/links' failure on the extra SFC solution. If affected, the value of preparing the extra placement and scheduling solution for SFC is equal to zero. Fig. 3 is plotted to assist grasping the mainline of *Sec-SFC-Intell* framework.

With respect to the SFC processing order, adopted by our *Sec-SFC-Intell*, the known 'first arriving, first being processed' strategy is selected. Two major reasons are responsible for this selection: 1. the *Sec-SFC-Intell* framework is proposed so as to be applied real dynamic networking application. This processing order is suitable to the real environment; 2. the *Sec-SFC-Intell* framework is designed to process each SFC placement and scheduling within polynomial time, not exponential time. Since each SFC has limited lifetime, the consumed time of doing SFC placement must be minimized. Take note that $PhyNet$ and $SFC(i)$ are selected as examples throughout this section.

A. Secure and Intelligent SFC Placement and Scheduling and Simultaneous Physical Elements Checking

When receiving the ith arriving SFC ($SFC(i)$), *Sec-SFC-Intell* framework will run its first module so as to do the secure placement and scheduling of $SFC(i)$ onto $PhyNet$ in an intelligent way. This module has two parallel parts: *Secure and Intelligent SFC Placement and Scheduling* part and *Simultaneous Physical Elements Checking* part (Fig. 3). Both parts run simultaneously.

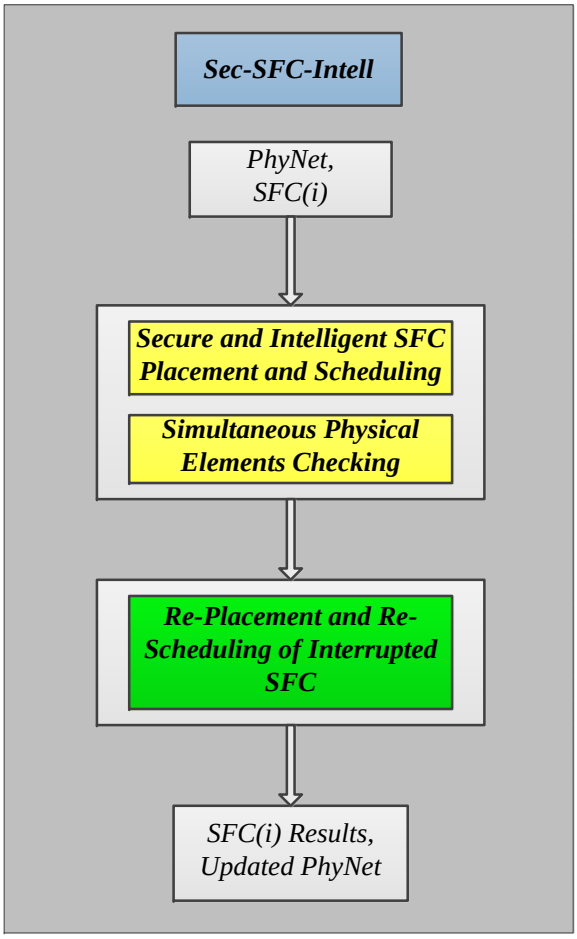

Fig. 3: Workflow of *Sec-SFC-Intell* Framework

1) *Secure and Intelligent SFC Placement and Scheduling*: within this part, the processing methods of $SFC(i)$ and $PhyNet$ must be determined and described in priority.

With respect to the $SFC(i)$, all virtual nodes and virtual links in $SFC(i)$ will be processed in order: first processing all nodes, second processing all links. Since $SFC(i)$ is represented by a weighted directed graph, the first virtual node $SFC(i)Node(1)$ is to be processed in the first place. Remaining virtual nodes are processed in order. After completing processing all virtual nodes, virtual links will be processed. Among all virtual links, the first virtual link $SFC(i)Node(1)SFC(i)Node(2)$ is processed in priority. Remaining virtual links are processed in order.

With respect to the $PhyNet$, it is crucial for *Sec-SFC-Intell* to be equipped with one quantifying method so as to quantify all physical nodes in $PhyNet$. Take note that the quantifying method must be intelligent and accurate enough to reveal placement abilities of all nodes in $PhyNet$. As usual, the local resource product (Equation (6)) can be adopted directly. However, this direct product method cannot reveal the abilities accurately. Hence, we adopt the Markov random model based method [26][27] to quantify and calculate the stable placement abilities of all physical nodes in $PhyNet$. The reason for selecting this quantifying method

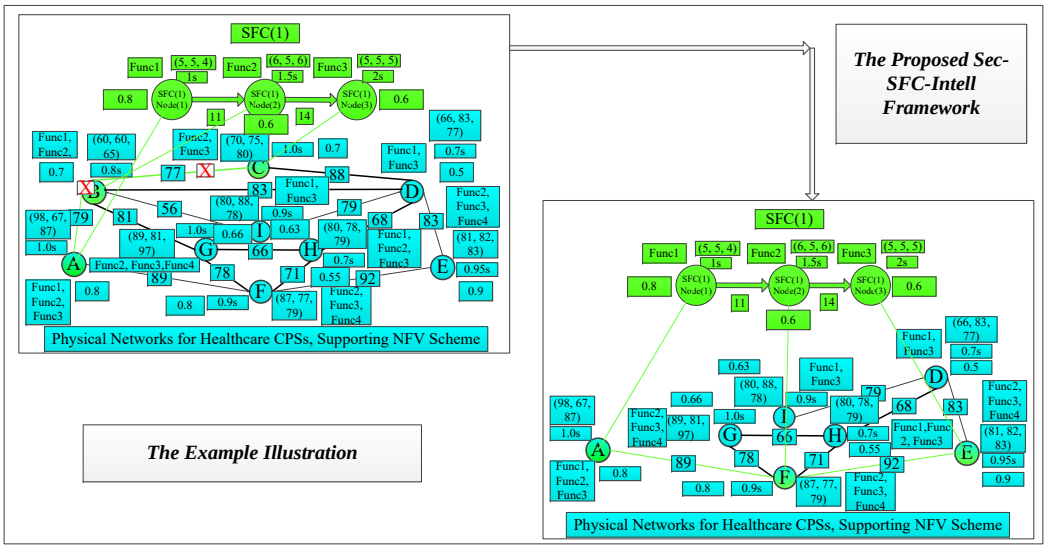

Fig. 4: Example Illustration of *Sec-SFC-Intell* Framework

was well discussed in ref. [26]. Equations are presented, ranging from Equation (7) to Equation (13). Within the Markov method, we adopt the iterative method instead of indirect calculation. This aims at getting rid of continuous calculation. The number of iterations can be adjusted and intelligent enough to reveal placement abilities.

$$LocPro(M) = CPU(M) \cdot Mem(M)$$
$$\cdot Sto(M) \cdot Sec(M) \cdot \sum_{MN \in LinkSet(M)} Band(MN) \quad (6)$$

where $LinkSet(M)$ indicates the direct link set of M. MN represents one link having both end nodes M and N. M and N are both nodes.

The initial ability value of node M can be computed below:

$$IniAbi(M) = \frac{LocPro(M)}{\sum_{M \in PhyNodes} LocPro(M)} \quad (7)$$

where $PhyNodes$ indicates the node set storing all physical nodes. The, we introduce the forwarding probability (Equation (8)) and jumping probability (Equation (9)):

$$ForPro(MN) = \frac{IniAbi(N)}{\sum_{U \in NebNodes(M)} IniAbi(U)} \quad (8)$$

$$JuPro(MN) = \frac{IniAbi(N)}{\sum_{U \in PhyNodes} LocPro(U)} \quad (9)$$

where $NebNodes(M)$ indicates the neighboring nodes of M. U is certain one physical node. Apparently, the sum of forwarding probability of M is i. The sum of jumping probability of N is 1. Both kinds of probabilities can be accepted as ability voting for M. Thus, another equation can be achieved below:

$$IniAbi(M)^{(2)} = \sum_{N \in NebNodes(M)} \alpha(M) \cdot ForPro(MN) \cdot$$
$$IniAbi(M) + \sum_{N \in PhyNodes} \beta(M) \cdot JuPro(MN) \cdot IniAbi(M) \quad (10)$$

where α and β are both wighting factors. Both sum is equal to 1. After t times of calculation, it can be achieved:

$$IniAbi(M)^{(t+1)} = \sum_{N \in NebNodes(M)} \alpha(M) \cdot ForPro(MN) \cdot$$
$$IniAbi(M)^{(t)} + \sum_{N \in PhyNodes} \beta(M) \cdot JuPro(MN) \cdot IniAbi(M)^{(t)} \quad (11)$$

When further extended, the vector form of physical network can be achieved:

$$VecIniAbi^{(t+1)} = T \cdot VecIniAbi^{(t)} \quad (12)$$

where $VecIniAbi^{(t+1)}$=$(IniAbi(1)^{(t+1)} ... IniAbi(M)^{(t+1)} ...)$. T is one-step transition Markov matrix. Details of T are presented below:

$$T = \begin{bmatrix} \alpha(1) & 0 & ... & 0 \\ 0 & \alpha(2) & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & ... & \alpha(|PhyNodes|) \end{bmatrix} \cdot$$
$$\begin{bmatrix} ForPro(11) & ... & ForPro(1|PhtNodes|) \\ ... & ... & ForPro(2|PhtNodes|) \\ ... & ... & ... \\ ForPro(|PhtNodes|1) & ... & ForPro(|PhtNodes||PhtNodes|) \end{bmatrix}$$
$$+ \begin{bmatrix} \beta(1) & 0 & ... & 0 \\ 0 & \beta(2) & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & ... & \beta(|PhyNodes|) \end{bmatrix} \cdot$$
$$\begin{bmatrix} JuPro(11) & ... & JuPro(1|PhtNodes|) \\ ... & ... & JuPro(2|PhtNodes|) \\ ... & ... & ... \\ JuPro(|PhtNodes|1) & ... & JuPro(|PhtNodes||PhtNodes|) \end{bmatrix} \quad (13)$$

Derived from ref. [27], matrix T is stable and stochastic. The maximum eigenvalue of T is 1. Thus $VecIniAbi$ will converge to the steady ability values. After calculation, all quantified physical nodes ($VecIniAbi$) will be stored, following the decreasing order of calculated values.

Then, the secure and intelligent placement and scheduling of $SFC(i)$ starts. Pseudo codes of this part are presented in **Algorithm 1**. The $SFC(i)Node(1)$ is selected to compare with the highest placement-ability-value physical node ahead. If the highest physical node has available CPU, memory, storage, function, processing delay and security to accommodate $SFC(i)Node(1)$, the initial placement and scheduling of $SFC(i)Node(1)$ is done. If the highest physical node fails to match $SFC(i)Node(1)$ demands, the second highest physical nodes is selected instead. Try until the $SFC(i)Node(1)$ is placed successfully. If no physical node in $PhyNet$ can accommodate $SFC(i)Node(1)$, the $SFC(i)$ will be rejected without any doubt. Pseudo codes of this procedure range from **Line 5** to **Line 10** of **Algorithm 1**. With respect to remaining virtual nodes in $SFC(i)$, the above selecting strategy repeats. Until no virtual node of $SFC(i)$ is left, the secure and intelligent node placement of $SFC(i)$ is done. The above procedures for nodes placements were proved to be completed within polynomial time, derived from ref. [9].

With respect to the placement of $SFC(i)$'s links, the selecting strategy is same to the problem of finding the shortest path between two fixed nodes. Within each pair of nodes, only one resource constraint (bandwidth) is considered in this paper. Hence, the universal shortest-path method (e.g. Dijkstra's algorithm) [27] can be adopted directly. In ref. [26], this placement procedure can be done within polynomial time. As discussed above, the secure and intelligent SFC placement and scheduling part of *Sec-SFC-Intell* can be completed within polynomial time. In addition, we need to take note that previous researchers had not adopted the Markov method to address SFC issue in NFV-enabled CPSs [4][6][7][]8. Though Markov method was adopted to deal with these resource allocation issues (e.g. virtual network embedding, VNE [28][29]), it has not been extended to deal with SFC in NFV-enabled CPSs. This is one major difference with prior works. With respect to our adopted shortest path method, it does not have apparent difference. It is owing to the fact that we need to find the shortest path between two fixed physical nodes. Between both fixed nodes, only one restriction is required to be considered (bandwidth). Hence, the universal shortest path method is enough in this paper.

2) *Simultaneous Physical Elements Checking*: Since *Sec-SFC-Intell* framework is designed to provide the sustainable SFC services, the survivability of *Sec-SFC-Intell* must be developed and strengthened. Hence, the checking all physical elements part is included in the *Sec-SFC-Intell* framework. This part aims at detecting and finding the failed physical element(s) immediately the element(s) failure happens. This part runs when *Sec-SFC-Intell* framework starts to run the secure and intelligent SFC placement and scheduling part.

Algorithm 1 Secure and Intelligent Placement and Scheduling of Nodes in $Slice(i)$

Input: Underlying $PhyNet$ for CPSs, $Slice(i)$
Output: Placement and scheduling results of all nodes in $Slice(i)$

1: Define an integer variable J, having initial value of 0;
2: Define an integer variable Num, having initial value of 0;
3: Set the value of Num to be $|SliceNode(i)|$;
4: **while** $Num \geq J$ **do**
5: Select the Jth virtual node to compare with available physical nodes in $PhyNet$;
6: **if** all (function, resource, delay, security) demands of Jth virtual node are fulfilled **then**
7: Place the physical node to Jth virtual node;
8: Selecting method follows the decreasing order of calculated values in $PhyNet$;
9: **if** certain demand cannot be fulfilled **then**
10: Reject to serve $Slice(i)$;
11: **end if**
12: $J++$;
13: **end while**
14: Skip to do secure and intelligent link placement of $Slice(i)$;
15: Output node placement and scheduling results of $Slice(i)$.

Since this paper concentrates on guaranteeing the affected SFC to recover from any physical element failure, the reason of leading to physical element failures is not within the scope of this paper.

Take note that the checking part works when the initial secure and intelligent SFC placement and scheduling starts. Both are parallel procedures and can run within polynomial time. The complexity of this elements checking part is mainly determined by the amount of physical nodes and links in $PhyNet$. In addition, when failures of certain one or more elements are detected, it will not affect the secure and intelligent SFC placement and scheduling. The elements detecting results will affect the following *Re-Placement and Re-Scheduling of Interrupted SFC* module.

B. Re-Placement and Re-Scheduling of Interrupted SFC

When certain one or more physical elements fail, the failed element(s) will be detected and located. Then, the *Re-Placement and Re-Scheduling of Interrupted SFC* module of *Sec-SFC-Intell* start to work. This module firstly needs to ensure whether the failed elements affects the deployed $SFC(i)$.

If not affected, the failed elements are simply required to be processed. It is not necessary to do the re-placement and re-scheduling the $SFC(i)$. If certain one physical node fails, the failed node and its directly connected links will be isolated and removed from $PhyNet$. If certain one link fails, it will be isolated and removed from the $PhyNet$.

TABLE I: Key Parameters for Simulation Work

Physical Network Scale	100
Physical Node Connectivity Probability	0.5
Equipped Functions per Physical Node	Arbitrary Three Functions From [1, 2, 3, 4] Set
Computing, Memory, Storage and Bandwidth	Real Number, Following the Uniform Distribution [70,100]
Data Processing Delay	Real Number, Uniform Distribution [0.5s, 2s]
Physical Node Security Probability	Real Number, Uniform Distribution [0.5, 1]
Each SFC Scale	Integer, Uniform Distribution [3,6]
Virtual Node Connectivity Probability	0.5
Data Processing Delay	Real Number, Uniform Distribution [1S, 4S]
Computing, Memory, Storage and Bandwidth	Real Number, Following the Uniform Distribution [1,10]
Required Function per Slice Node	Arbitrary One Function From [1, 2, 3, 4] Set
Required Node Security Probability	Real Number, Uniform Distribution [0.7, 1]

With respect to more than one failed elements, the above removing strategy can be adopted.

If certain one or more virtual elements of $SFC(i)$ are affected by the failed physical elements, the recovery procedure consists of two parts: one is the failed elements isolation and movement, the other is the re-placement and re-scheduling of the affected virtual elements. The strategy of isolation and movement part is same to what are presented above. With respect to the re-placement and re-scheduling part, it includes the affected virtual nodes and links. The affected virtual nodes are re-placed and re-scheduled in priority. When all affected virtual nodes of $SFC(i)$ are done, the affected virtual links are stared to be re-placed.

With respect to the adopted strategies for affected nodes and links, it is same to the placement and scheduling strategies in *Secure and Intelligent SFC Placement and Scheduling and Simultaneous Physical Elements Checking* module. Only one difference needs to be noted: the failed physical nodes and links are excluded. Derived from ref. [27], the re-placement and re-scheduling of affected $SFC(i)$ can be done within polynomial time. Another example illustration is plotted in Fig. 4 so as to help readers understand the sustainable ability of *Sec-SFC-Intell* well.

C. Updation for Serving Next SFC

As discussed in above subsections, the secure and sustainable placement and scheduling of $SFC(i)$ is done by *Sec-SFC-Intell*. The proposed *Sec-SFC-Intell* framework is guaranteed to provide the sustainable SFC service. Then, the *Sec-SFC-Intell* will serve another arriving SFC so as to continue to provide the sustainable service. Before serving the new arriving SFC, it is necessary to update the $PhyNet$. With respect to $PhyNet$, the resource, function, and other attributes will be updated. In addition, the failed physical elements will be updated, avoiding being selected to serve the new SFC. These updations are promised to be done within limited time. The time complexity is within $O(|PhyNodes| * |PhyLinks|)$ [11], where $|PhyNodes|$ and $|PhyLinks|$ are numbers of nodes and links in $PhyNet$, respectively.

D. Discussion of Sec-SFC-Intell Framework

This subsection generally discusses the complexity of *Sec-SFC-Intell* framework. In the end of above subsections, the complexity of each module is discussed separately. Hence, the secure and intelligent SFC placement and scheduling can be completed within polynomial time (secure and intelligent SFC placement, sustainable re-placement and updation). Thus, *Sec-SFC-Intell* framework can be evaluated in dynamic environment and continuous time event.

IV. Evaluation Work

A. Parameters Settings and Selected Benchmarks

The proposed *Sec-SFC-Intell* framework is evaluated by doing simulation work. It is owing to the fact that no specific prototype for SFC and CPSs has not been developed in both academia and industry yet [30][31]. With respect to the physical network, representing underlying CPSs and consisting of physical elements and devices, its scale and key parameters are presented in Table I. With respect to each SFC, its scale and key parameters are introduced in Table I. In this evaluation work, the simulation will last 10000 time units. SFCs are generated, following the Poisson Distribution. Its arriving rate is 4 SFCs per 100 time units. In every 100 time point, 5 physical elements (nodes and links) on average is set to fail.

With respect to the compared SFC frameworks, two basic benchmarks are derived and selected: *BasicOne-SFC* and *BasicTwo-SFC*. The *BasicOne-SFC* framework simply has the initial SFC placement and scheduling part while ignoring recovering the physical elements failures. The difference between *Sec-SFC-Intell* and *BasicTwo-SFC* is the SFC re-placement and re-scheduling. *Sec-SFC-Intell* concentrates on the affected virtual elements re-placement when physical

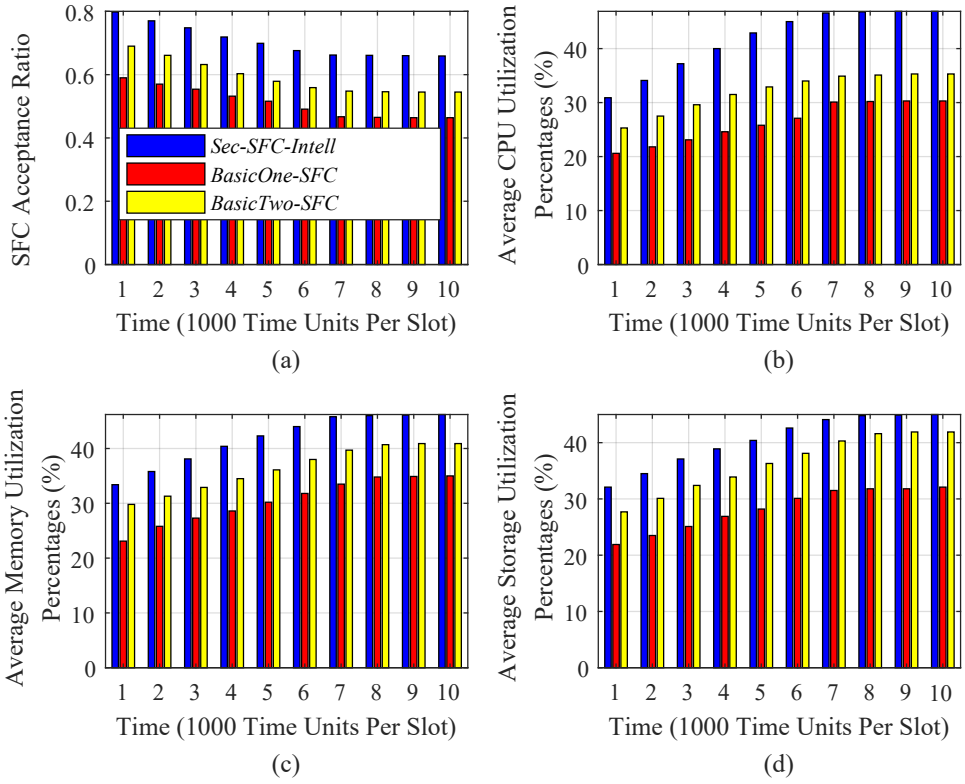

Fig. 5: Major Simulation Results of *Sec-SFC-Intell*, *BasicOne-SFC*, *BasicTwo-SFC*

elements happen while *BasicTwo-SFC* re-places and re-schedules the whole SFC when physical elements happen. As described in Section III, we do not select the counterpart, preparing extra one SFC placement and scheduling solution for each served SFC, to be the compared framework in the evaluation section.

B. Evaluation Results and Discussion

In Fig. 5, we illustrate major results of three selected SFC frameworks. The merits of *Sec-SFC-Intell* framework will be described from the following three aspects:

1) ***SFC Acceptance Advantage***: In SFC research, the key and dominant performance metric is the SFC acceptance ratio [7]. The SFC acceptance ratio can directly reveal the SFC framework/algorithm's ability of successfully placing and scheduling SFCs. Generally speaking, the higher value of acceptance ratio, the stronger ability of SFC framework/algorithm. The SFC acceptance results of three selected frameworks are plotted in Fig. 5 (a). Derived from Fig. 5 (a), three apparent discoveries can be found. The first discovery is the *Sec-SFC-Intell* framework has apparent SFC acceptance advantage, compared with two frameworks. Since the simulation starts, the performance gap between *Sec-SFC-Intell* framework and the better performed *BasicTwo-SFC* framework is becoming larger and larger. With time extending, the gap will be stable, approaching 10% in the end. The second discovery is that SFC acceptance results of three frameworks all have the 'high-decrease-stable' behaviors. The last discovery is that both *Sec-SFC-Intell* and *BasicTwo-SFC*, considering the survivable SFC re-placement, perform better than the BasicOne-SFC without considering SFC re-placement and re-scheduling.

This paragraph focuses on analyzing the causes of above three discoveries. With respect to the reason of the first discovery, it is the intelligent and efficient placement and scheduling strategy of *Sec-SFC-Intell*. On the one hand, the *Sec-SFC-Intell* can calculate the accurate placement-ability-value of all physical devices and nodes in an intelligent manner, serving as the basis of the SFC assignment and placement. Resource and security abilities of all physical nodes can be made full use by our *Sec-SFC-Intell* framework. On the other hand, *Sec-SFC-Intell* can re-place and re-cover the affected virtual elements accurately without re-placing all virtual elements of the affected SFC, when certain one physical failure happens. The *BasicOne-SFC* framework ignores the re-placement while *BasicTwo-SFC* re-places all virtual elements in SFC. Hence, the proposed *Sec-SFC-Intell* can efficiently do the SFC placement and achieve the apparent acceptance advantage. With respect to the reason of the second discovery, it is the finite physical resources for accommodating continuous SFCs. As physical resources are abundant in the early stage, the SFC acceptance ratio of three frameworks will remain at a high level. With physical resources being consumed fast and released slowly, no abundant resources will be reserved for maintaining the high SFC acceptance ratio. However, the balance will be achieved. The SFC acceptance ratio will be stable in the end.

With respect to the reason of the third discovery, it is the importance of re-placement and re-scheduling of *Sec-SFC-Intell*. Since physical elements fail from time to time, the accommodated SFCs, having affected virtual elements, will be interrupted and require the sustainable placement. Thus, it is necessary to do the re-placement and re-scheduling of the affected virtual elements. Different strategies have different behaviors. If no re-placement is done, the acceptance will decrease directly. That is why *BasicOne-SFC* has the lowest long-term SFC acceptance throughout the simulation work.

2) ***Resource Utilization Advantage***: In the remaining three figures of Fig. 5, the node resources (CPU, memory, storage) utilization results are plotted. The most apparent discovery is that our proposed *Sec-SFC-Intell* utilizes the most amount of resources than the *BasicOne-SFC* and the *BasicTwo-SFC*. Apparently, the cause of this discovery is in accordance with the SFC acceptance advantage reason. Since *Sec-SFC-Intell* achieves the highest SFC acceptance ratio, it will consume more physical resources (CPU, memory, storage). With respect to the causes of this discovery, it has been well discussed in the above paragraphs. For saving pages, we do not repeat the cause discussion in this paragraph. In Fig. 6, we plot the average bandwidth consumption percentages of three evaluated frameworks. As expected, our *Sec-SFC-Intell* framework consumes the most amount of bandwidth resources of CPSs and has apparent advantage than the remaining frameworks. With respect to the reason of high consumption, our *Sec-SFC-Intell* accommodates the most amount of slices. Thus achieving the highest bandwidth consumption.

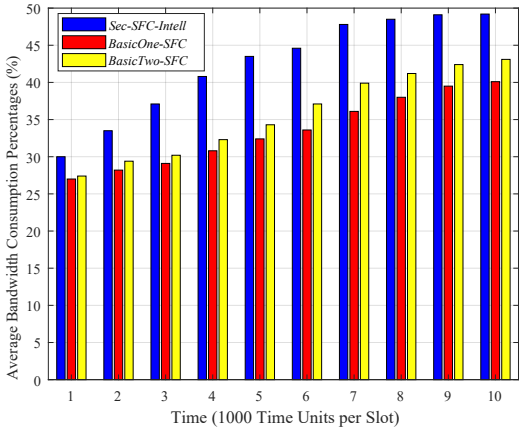

Fig. 6: Bandwidth Results of *Sec-SFC-Intell*, *BasicOne-SFC*, and *BasicTwo-SFC*

3) ***Better Secure, Intelligent and Sustainable Performance***: Derived from all simulation results in Fig. 5 and Fig. 6, it is important to consider the re-placement and re-scheduling when physical elements fail. Not only the high SFC acceptance ratio can be fulfilled, but also the sustainable SFC can be achieved. That is to say, it is necessary to improve the sustainable ability of SFC algorithms. With considering and enhancing the SFC sustainable ability, the SFC acceptance will remain to be high. In addition, the physical resources will be made the full use. Our proposed *Sec-SFC-Intell* framework can fully exploit the node potential so as to realize the secure and intelligent SFC placement. In addition, our *Sec-SFC-Intell* considers the SFC re-placement and re-scheduling in case of the failure of any physical device. On these basis, the acceptance ratio advantage of our *Sec-SFC-Intell* can be guaranteed. Thus achieving the best SFC placement performance among three selected frameworks.

In addition, the SFC in NFV-enabled CPSs for healthcare purpose is a multi-dimensional resource allocation [4][8] problem, it is hard to adopt the exact approach to get the optimal and exact solution per SFC. This is the common sense in the literature [6][7][28][29]. Since our proposed *Sec-SFC-Intell* framework and its included method are not based on the exact solution (e.g. integer linear programming [27]), it is hard to achieve the theoretical upper bound of slice acceptance ratio and corresponding resource utilization. Not to mention proving its efficiency by using pure and accurate equations. In addition, our proposed method strictly belong to the heuristic and can do the re-placement of affected SFC flexibly. That is its intelligence, comparing with previous one-round of heuristic algorithms. In order to prove its efficiency, we derive two benchmarks for comparison. Each benchmark lacks in part of our proposed framework. By running the simulation, our proposed framework performs better. Thus proving its efficiency.

V. Conclusion Marks

Healthcare CPSs, consisting of dozens of IoT devices and terminals, promise to provide operators (doctors and hospitals) and users (patients and their relatives) with more tailored and sustainable services when integrating the NFV technology. These services are usually modeled as SFCs in NFV research. In this paper, we investigate and propose one novel and intelligent SFC placement and scheduling framework, abbreviated as *Sec-SFC-Intell*. The goal of *Sec-SFC-Intell* is to provide the secure and sustainable SFC services for healthcare CPSs in an intelligent and efficient manner. When receiving one network service, modeled by SFC, our *Sec-SFC-Intell* firstly realizes the SFC placement and allocates the required resources and security demands. Meanwhile, the *Sec-SFC-Intell* will monitor and check states of all underlying physical elements. When detecting certain one or more elements failures, *Sec-SFC-Intell* will isolate them from the whole physical CPS. In addition, the affected virtual elements of the SFC will be re-placed and re-scheduled so as to make the affected SFC sustainable. To validate the merits of *Sec-SFC-Intell*, two derived frameworks (*BasicOne-SFC*, *BasicTwo-SFC*) are selected for simulation comparison. Especially, the SFC acceptance and resource utilization advantages of *Sec-SFC-Intell* are highlighted.

In the future works, we will concentrate on following major aspects. The first aspect is that we will try to develop a prototype [32][33] so as to evaluation the performance of our intelligent *Sec-SFC-Intell* framework from the application

aspect. The second aspect is that we will try to research and optimize the concrete performance metric of these SFCs, such as the end-to-end latency. The third aspect is that we will try to research the effect of the combination of backup and re-deployment SFCs assignments. The last aspect is that we will model the security attacks (e.g. DDoS, flooding) in the developed prototype [34]. If that, our proposed framework and its included algorithm can explore the security from the application aspect. It is necessary to consider and develop the security attacks so as to research the security and piracy performance deeply.

REFERENCES

[1] Z. You and L. Feng, "Integration of industry 4.0 related technologies in construction industry: A framework of cyber-physical system," IEEE Access, vol. 8, pp. 122908-122922, 2020.

[2] R. Chaudhary, A. Jindal, G. Aujla, N. Kumar, A. Das, and N. Saxena, "LSCSH: Lattice-based secure cryptosystem for smart healthcare in smart cities environment," IEEE Commun. Maga., vol. 56, no. 4, pp. 24-32, Apr. 2018.

[3] Y. Zhang, M. Qiu, C. Tsai, M. M. Hassan and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," IEEE Systems Journal, vol. 11, no. 1, pp. 88-95, Mar. 2017.

[4] Z. You and L. Feng, "Integration of industry 4.0 related technologies in construction industry: A framework of cyber-physical system," IEEE Access, vol. 8, pp. 122908-122922, 2020.

[5] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. Turck and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 236-262, 2016.

[6] A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," Computer Networks, vol. 167, pp. 1-40, 2020.

[7] G. Mirjalily and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," Chin. J. Electron., vol. 27, no. 4, pp. 704-717, 2018.

[8] X. Li, C. Guo, L. Gupta and R. Jain, "Efficient and secure 5G core network slice provisioning based on VIKOR approach," IEEE Access, vol. 7, pp. 150517-150529, 2019.

[9] H. Cao, L. Yang and H. Zhu, "Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding," IEEE Internet of Things Journal,vol. 5, no. 1, pp.108-120, Feb. 2018.

[10] I. Kovacevic, A. S. Shafigh, S. Glisic, B. Lorenzo, and E. Hossain, "Multi-domain network slicing with latency equalization," IEEE Trans. Netw. Ser. Manag., vol. 17, no. 4, pp. 2182-2196, Dec. 2020.

[11] T. Huu, P. Mohan, and M. Gurusamy, "Service chain embedding for diversified 5G slices with virtual network function slicing," IEEE Commun. Lett., vo. 23, no. 5, pp. 826-829, May 2019.

[12] F. Bahlke, O. D. Ramos-Cantor, S. Henneberger, and M. Pesavento, "Optimized cell planning for network slicing in heterogeneous wireless communication networks," IEEE Commun. Lett., vol. 22, no. 8, pp. 1676-1679, Aug. 2018.

[13] H. Yao, X. Chen, M. Li, P. Zhang and L. Wang, "A novel reinforcement learning algorithm for virtual network embedding," Neurocomputing, vol. 284, pp. 1-9, 2018.

[14] T. Li, X. Zhu and X. Liu, "An end-to-end network slicing algorithm based on deep Q-learning for 5G network," IEEE Access, vol. 8, pp. 122229-122240, 2020.

[15] H. Cao et al., "Resource-ability assisted service function chain embedding and scheduling for 6G networks with virtualization," IEEE Trans. Veh. Tech., vol. 70, no. 4, pp. 3846-3859, April 2021.

[16] X. Fu, F. Yu, J. Wang, Q. Qi and J. Liao, "Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning," IEEE Commun. Magaz., vol. 57, no. 9, pp. 102-108, Sep. 2019.

[17] S. Kulkarni et al., "NFVnice: Dynamic backpressure and scheduling for NFV service chains," IEEE/ACM Trans. Netw., vol. 28, no. 2, pp. 639-652, Apr. 2020.

[18] J. Du, L. Zhao, X. Chu, F. R. Yu, J. Feng and I. C.L., "Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems," IEEE Trans. Veh. Tech., vol. 68, no. 2, pp. 1757-1771, Feb. 2019.

[19] Y. Hao, M. Chen, H. Gharavi, Y. Zhang, and K. Hwang, "Deep reinforcement learning for edge service placement in softwarized industrial cyber-physical system," IEEE Trans. Indus. Infor., vol. 17, no. 8, pp. 5552-5561, Aug. 2021.

[20] D. Wang, N. Zhao, B. Song, P. Lin, and F. R. Yu, "Resource management for secure computation offloading in softwarized cyberphysical systems," IEEE Internet of Things Journal, vol. 8, no. 11, pp. 9294-9304, Jun. 2021.

[21] K. Chang, K. Chu, H. Wang, Y. Lin, and J. Pan, "Agent-based middleware framework using distributed CPS for improving resource utilization in smart city," (Elsevier) Future Generation Computer Systems, vol. 108, pp. 445-453, Jul. 2020.

[22] A. Termehchi and M. Rasti, "Joint sampling time and resource allocation for power efficiency in industrial cyber-physical systems," IEEE Trans. Indus. Infor., vol. 17, no. 4, pp. 2600-2610, Apr. 2021.

[23] J. Zhao and Q. Dong, "Allocation algorithm of CPS communication resources based on cooperative game," (Elsevier) Computer Communications, vol. 160, pp. 63-70, Jul. 2020.

[24] Y. Hu, Y. Guo, "Survivable service function chain mapping in NFV-enabled 5G networks," in Proc. of IEEE NetSoft 2021, pp. 375-380, Jul. 2021.

[25] H. Cao, H. Zhao, D. X. Luo, N. Kumar, and L. Yang, "Dynamic virtual resource allocation mechanism for survivable services in emerging NFV-enabled vehicular networks," IEEE Trans. Intell. Trans. Syst., vol. pp, no. 99, pp. 1-13, Oct. 2021.

[26] H. Cao, A. Xiao, Y. Hu, P. Zhang, S. Wu and L. Yang, "On virtual resource allocation of heterogeneous networks in virtualization environment: A service oriented perspective," IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 4268-4280, Dec. 2020.

[27] T. H. Cormen, C. Stein, R. Rivest and C. Leiserson, Introduction to Algorithms, 2nd ed. McGraw-Hill Higher Education, 2001.

[28] H. Cao, H. Hu, Z. Qu, and L Yang, "Heuristic solutions of virtual network embedding: A survey," China Commun., vol. 15, no. 3, pp. 186-219, Mar. 2018.

[29] H. Cao, S. Wu, Y. Hu, Y. Li, and L. Yang, "A survey of embedding algorithms for virtual network embedding," China Commun., vol. 16, no. 12, pp. 1-33, Dec. 2019.

[30] X. You, C. Wang, et al., "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," Sci. China Inf. Sci., vol. 64, no. 1, 110301, Jan. 2021.

[31] P. Zhang, C. Wang, C. Jiang and A. Benslimane, "Security-aware virtual network embedding algorithm based on reinforcement learning," IEEE Trans. Net. Sci. Engi., vol. 8, no. 2, pp. 1095-1105, Apr. 2021.

[32] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey", IEEE Trans. Netw. Service Manag., vol. 13, no. 3, pp. 518-532, Sep. 2016.

[33] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, "Recent advances of resource allocation in network function virtualization," IEEE Trans. Para. Distr. Sys., vol. 32, no. 2, pp. 295-314, Feb. 2021.

[34] A. Jindal, A. Dua, N. Kumar, A. Vasilakos, and J. Rodrigues, "An efficient fuzzy rule-based big data analytics scheme for providing healthcare-as-a-service," in 2017 IEEE International Conference on Communications (ICC), pp. 1-6, May 2017.