GREP: Guaranteeing Reliability with Enhanced Protection in NFV

Jingyuan Fan
SUNY Buffalo
Buffalo, NY, USA
jfan5@buffalo.edu

Zilong Ye
SUNY Buffalo
Buffalo, NY, USA
zilongye@buffalo.edu

Chaowen Guan
SUNY Buffalo
Buffalo, NY, USA
chaoweng@buffalo.edu

Xiujiao Gao
SUNY Buffalo
Buffalo, NY, USA
xiujiaog@buffalo.edu

Kui Ren
SUNY Buffalo
Buffalo, NY, USA
kuiren@buffalo.edu

Chunming Qiao
SUNY Buffalo
Buffalo, NY, USA
qiao@computer.org

ABSTRACT

Network Function Virtualization (NFV) is a promising technique to greatly improve the effectiveness and flexibility of network management through a process called Service Function Chain (SFC) mapping, which can efficiently provision network services over a virtualized and shared middlebox platform. However, such an evolution towards software-defined middlebox introduces new challenges to network services which require high reliability. Sufficient redundancy can protect the network services when physical failures occur, but in doing so, the efficiency of physical resources may be greatly decreased. This paper presents GREP, a novel online algorithm that can minimize the physical resources consumption while guaranteeing the required high reliability with a polynomial time complexity. Simulation results show that our proposed algorithm can significantly improve the request acceptance ratio and reduce resource consumption.

CCS Concepts

•Computer systems organization → Reliability; Redundancy; •Networks → Network resources allocation; •Theory of computation → Design and analysis of algorithms;

Keywords

Service Function Chain (SFC); reliability; algorithms

HotMiddlebox '15 August 21, 2015, London, United Kingdom

 ISBN 978-1-4503-3540-9/15/08…$15.00

DOI: http://dx.doi.org/10.1145/2785989.2786000

1. INTRODUCTION

Network Function Virtualization (NFV) is a driving force behind implementing middleboxes in software-based processing functions that run on the standardized commodity storage, servers and switches. NFV enables a virtualized and shared middlebox platform that can significantly reduce the hardware cost and investment, as well as highly improve the efficiency and flexibility of utilizing physical hardware resources. From the service providers' perspective, it is essential to find an optimal mechanism to deploy middleboxes from distinct service providers on the shared platform through *Service Function Chain* (SFC) mapping. A SFC consists of a set of *Virtual Network Functions* (VNFs) interconnected by logical links. Multiple SFCs from distinct clients may share the computing and networking resources in order to improve the resource utilization.

However, software-based middleboxes introduce unique reliability challenges to future networks. Traditional ways of improving reliability by utilizing diversity may no longer work as we use consistent platform architecture in NFV. Faults at one physical equipment can influence multiple service chains, and even its impacts may span over multiple functions in one service chain. In addition, virtual machines used for a service chain may be distributed across different racks in a data center or geographically different locations, and this introduces new potential failures [1]. Since a service is considered available only when all the functions it requires are available, reliably mapping functions onto physical substrate is critical and challenging in NFV.

Ideally, service providers are expected to mask all failures before clients experience any disruptions. Typical 1:1 redundancy architecture has been proven ineffective [2,3]. So, in this paper, we investigate: *what is the minimum number of backup VNFs service provider needs to provision to guarantee a certain degree of reliability? What is the best protection strategy in terms of reliabil-*

ity improvement and resource consumption? Furthermore, how to map both primary and backup VNFs and interconnecting logical links in a resource-efficient way? We introduce redundancy in a virtualization layer because dynamic creations of VNFs with NFV is easy. The goal is to use the least amount of resources to meet each request's reliability requirement such that a higher SFC request acceptance ratio can be made, while reducing the costs for clients and service providers. Hence, developing an effective protection mechanism and an efficient SFC mapping scheme to meet the clients' *Service Level Agreement* (SLA) (e.g., the reliability requirement) are essential while consuming a small amount of physical resources.

In this paper, we address the problem of reliable SFC mapping, and propose an online algorithm called *Guaranteeing Reliability with Enhanced Protection* (GREP), which uses a novel protection strategy, a novel backup selection mechanism and a low-complexity reliability evaluation method to guarantee each client's reliability requirement while minimizing the amount of resources allocated. Note that the reliable SFC mapping problem is more difficult than the problem of survivable virtual infrastructure mapping studied earlier in [4-6] for two reasons: 1) we need to consider network function restrictions; 2) we need an effective algorithm to search for and evaluate an efficient backup plan to meet specific reliability requirements of heterogeneous SFC requests. To the best of our knowledge, there is no existing work on addressing such a problem. In summary, we list the main contributions as follows:

- We propose a novel enhanced *Joint Protection* (JP) approach, and demonstrate its advantages by comparing with traditional *Dedicated Protection* (DP) and *Shared Protection* (SP) in terms of acceptance ratio performance and resource consumption.
- We for the first time prove that there's no polynomial time algorithm for solving the proposed reliable SFC mapping problem.
- We develop an approximate method for computing SFC reliability with a polynomial time complexity, and show the approximation error is negligible.
- We propose a novel backup selection strategy, prove its local optimality, and illustrate that it can save the number of backup VNF by 37%.

The rest of the paper is organized as follows. Section 2 illustrate the problem. Section 3 analyzes the problem complexity, and describes a polynomial running time algorithm with a novel VNF protection strategy to guarantee reliability requirement. Section 4 shows experiment and evaluation results, and Section 5 concludes.

2. PROBLEM DESCRIPTION

We consider our problem with a generic network model. Given a *Physical Substrate* (PS) $P_s(N_s, L_s)$, where N_s is the set of *Physical Nodes* (PNs) and L_s is the set of *Physical Links* (PLs). For each PN $n \in N_s$, it is associated with a set of k types of resources $S_n^k = \{s_n^i | i \in [1,k]\}$, where s_n^i denotes the capacity of resource of type i. In addition, each PN n is associated with a reliability r_n which can be characterized in terms of *Mean Time Between Failures* (MTBF). Given the set of resources available at a PN n, it can provide a set of functions denoted by f_n^P. $F_n = \bigcup_{i=1}^{N_s} f_i^P$ is the set of all functions that the network can provide. For example, PN F, as shown in Fig. 1, can provide functions for *Home Subscriber Server* (HSS), and it's reliability is 0.94. For a given SFC request r denoted by $V_r(N_r, L_r, \theta_r)$, $N_r = \{n_r^1, n_r^2 ... n_r^{|N_r|}\}$ is the set of VNFs, each of which requires a set of resources to perform one single network function $f_{n_r^j} \in F_n$. $F_r = \bigcup_j f_{n_r^j}$ is the set of functions that request r needs. Each logical links $l_r \in L_r$ has a bandwidth demand and θ_r is the reliability requirement of this SFC. To map a SFC, we not only need to map a VNF to a PN by reserving an appropriate type and amount of resources in the chosen PN to perform the function requested by that VNF, but also map a logical link through allocating an appropriate amount of bandwidth along each and every physical link along the chosen path to carry the traffic flow from one VNF to another VNF. For instance, in Fig. 1, the VNF requiring *Media Resource Function* (MRF) on packet flow 1 can only be mapped onto PN D as it's the only PN providing such function. The traffic from MRF to HSS can flow from PN D to F directly or redirect to any other path starting with PN D and ending with PN F.

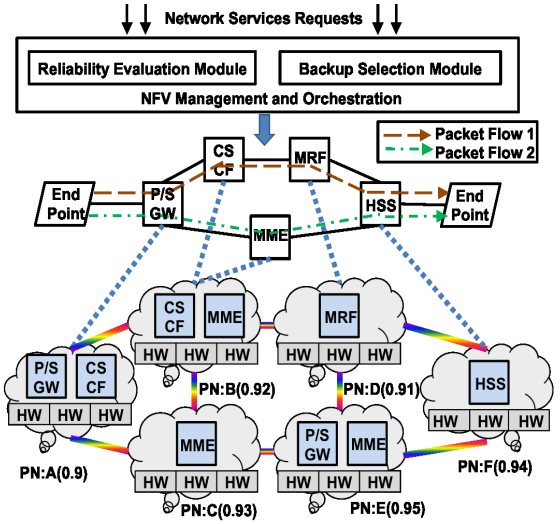

Figure 1: SFC mapping for NFV

However, merely mapping primary VNFs is not enough for achieving high reliability. A SFC is considered being available at a given time if all the functions it requests are able to function normally. In this work, we only consider node failures, so when no protection is provisioned, the reliability of a SFC request r can be obtained as $R_r = \prod_{f \in F_r} r_f$, where r_f is the reliability of the PN providing function f. A SFC request is considered as being blocked if any VNFs or logical links cannot be mapped or the reliability cannot meet the client's requirement. Therefore, we can define the SFC

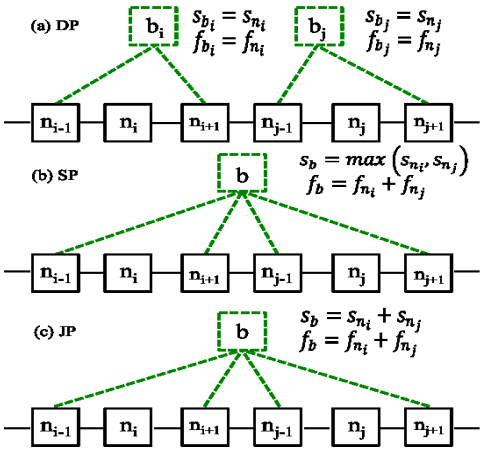

Figure 2: Different protection methods

reliable mapping problem as follows. Given a set of SFC requests, each with a specific reliability requirement, we need to find out the minimum number of backup VNFs needed in order to achieve each reliability requirement and efficiently assign primary and backup VNFs to physical nodes. A decision algorithm can be embedded in a large system that manages the incoming requests.

3. GREP: GUARANTEEING RELIABILITY WITH ENHANCED PROTECTION

3.1 Joint Protection for VNF Failure

As shown in Fig. 2(a), in a traditional DP scheme, backup b_i and b_j will duplicate the functionalities of n_i and n_j respectively, and connect to their neighboring VNFs (e.g., n_{i-1} and n_{i+1}). Although it provides high reliability, it results in high link resource usage, since adjacent VNFs may not share a logical link for connecting to backups. On the other hand, in a traditional SP scheme shown in Fig. 2(b), it saves resources by allocating $\max(s_{n_i}, s_{n_j})$ resources on backup b to protect either n_i or n_j and connecting the backup to the neighboring VNFs of n_i and n_j, but the network can fail when both of the VNFs protected by one backup fail. In this paper, we propose a novel reservation strategy called *Joint Protection* (JP) for VNF failures. It provides protection to both VNFs connected to it, and the amount of backup resources reserved at VNF b will be sufficient for both n_i and n_j (i.e., $s_b = s_{n_i} + s_{n_j}$) as shown in Fig. 2(c). Thus, the SFC can still function normally even if both n_i and n_j fail simultaneously. As a result, one in JP can provide high reliability as in DP while potentially consuming a small number of links as in SP. Our proposition will be validated via simulations.

3.2 Problem Complexity

Recall our reliable SFC mapping problem is to find the minimum number of backup VNFs each request needs to guarantee its reliability requirement. From **Thm. 1** we can see that this problem is intractable.

Theorem 1. *The problem of verifying if the reliability is above a given threshold (denoted by VR) is PP-complete, and finding the global minimum number of backups needed for a SFC request belongs to* $NP^{NP^{PP}}$.

Due to the limited space, we omit the detailed proof here and focus on our solution. To address the challenges, we can decompose this problem into two parts: reliability evaluation, and backup VNFs selection. In this section, we introduce an algorithm called GREP tackling these two challenges.

3.3 Overview of the GREP

The mapping process of SFC requests is similar to [7]. However, for each request, based on the physical network condition, we need to construct a backup plan with the proposed JP strategy using the proposed GREP algorithm to be described next. Through the mapping process, all primary VNFs are mapped to physical nodes with the most remaining resources. PLs connecting primary and backup VNFs are selected based on k-shortest paths.

We consider the whole network as a composition of several independent sub-networks, and thus, the reliability of the request is the multiplication of the reliability of each sub-network. At first, all VNFs are considered as separate sub-networks, so ρ in **Algo. 1** at first is the multiplication of the reliability of the physical nodes that primary VNFs are mapped to. Until the reliability requirement is met, two VNFs are selected for each iteration (details are discussed in Sec. 3.5) and provided with a backup VNF. Then, a new sub-network is formed, composed of the backup VNF and two sub-networks which consist of those two selected VNFs. The reliability of these two selected VNFs and this new sub-network need to be updated (details are discussed in Sec. 3.4). In general, the pseudo code of the procedure is as follows:

Inputs : $P_s(N_s, L_s)$, $V_r(N_r, L_r, \theta_r)$
Outputs: Backup node plan $\mathfrak{B}_N$ and link plan $\mathfrak{B}_L$

```
Calculate the request's reliability ρ
if ρ ≥ Θ_r
    No need to provide backup
while ρ < Θ_r do
    ℚ ←Select two VNFs n_r^i and n_r^j (based on 3.5) and find the mapping set for the backup
    if ℚ ≠{∅}
        𝔅_N ←Select the one from ℚ with the most remaining bandwidth resource
        𝔅_L ←Calculate backup logical links
        Update r_i,r_j and ρ (based on 3.4)
end
return 𝔅_N and 𝔅_L
```

Algorithm 1: GREP

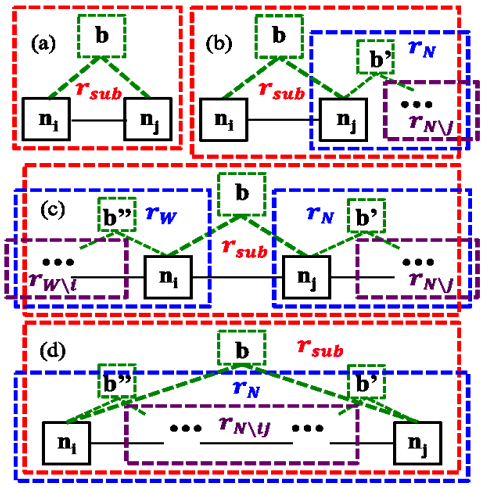

Figure 3: SFC reliability evaluation

3.4 Reliability Evaluation Model

As seen from **Thm. 1**, the problem VR is PP-complete, so there's no polynomial time solution to solve the problem. Previous research [8-9] proposed solutions using *Monte Carlo* and its related methods, but it's hard to determine how many steps are needed to converge to the stationary distribution within an acceptable error, and the procedure is time consuming. In this subsection, we will show a computational easier way to evaluate the reliability in Line 9 in **Algo. 1**.

As the whole network has been separated into several sub-networks, and every time we select two VNFs n_r^i and n_r^j to be provisioned with a backup b, there are totally four possible cases:

(a) **Neither of the selected VNFs has backups** If neither of the two selected VNFs has backups, the reliability of the sub-network they form equals to the probability that the backup is reliable or both of the two selected VNFs are reliable while the backup cannot work.

$$r_{sub} = 1 - (1 - r_b)(1 - r_i r_j) \tag{1}$$

(b) **Only one of the selected VNFs has backups** Without loss of generality, we assume VNF n_r^j is the one already has backups, and we denote the sub-network contains n_r^j as N. We analyze this situation by considering which sub-network provides the function that n_r^j requires. We can observe that this function is provisioned either by the sub-network N or the new backup b, and these two situations are mutually excluded. So the new sub-network is considered to be reliable if the sub-network N functions properly and at least one of the backup b and n_r^i works properly, or backup b is reliable and all the functions in sub-network N except for the function that n_r^j needs are reliable. Therefore,

$$r_{sub} = r_N \times (1 - (1 - r_b)(1 - r_i)) + r_b \times r_{N\setminus j} \tag{2}$$

where $r_{N\setminus j}$ is the probability that sub-network N can provide all the functions except the one that n_r^j needs. To compute $r_{N\setminus j}$, we decompose $r_N = r'_N - r_{N\setminus j} = r'_N - (1 - r_j)\prod_{k=1}^{M}(1 - r_{j_k})r''_N$, where r'_N is the probability that the sub-network N may or may not provide function n_r^j needs, while all the others function normally, r''_N is the probability that all functions except the one n_r^j needs in sub-network N are reliable, r_{j_k} is the kth backup connected with n_r^j excluding backup b, and M is the total number of backups that n_r^j currently has. We then define $\tau = \frac{r'_N}{r''_N} \approx 1 + \epsilon$, where ϵ is a small constant. Noted that as the sub-network contains more nodes, τ is closer to 1. So

$$r_{N\setminus j} = \frac{(1 - r_j)\prod_{k=1}^{M}(1 - r_{j_k})r_N}{\tau - (1 - r_j)\prod_{k=1}^{M}(1 - r_{j_k})} \tag{3}$$

(c) **Both of the selected VNFs have backups and they belong to different sub-networks** Using similar strategy, either both sub-networks W and N that contain n_r^i and n_r^j respectively work properly, or backup b is reliable and at least one of the sub-networks W and N fails to provide functions that n_r^i and n_r^j need. Hence,

$$r_{sub} = r_N r_W + r_b(r_{N\setminus j} r_{W\setminus i} + r_{N\setminus j} r_W + r_{W\setminus i} r_N) \tag{4}$$

(d) **Both of the selected VNFs have backups and they belong to the same sub-network** Similarly, when the sub-network N works properly, whether or not backup b can work has no effect on the whole network's availability. And when backup b is reliable, at most one of the functions that needed by n_r^i and n_r^j should be reliable. Thus,

$$r_{sub} = r_N + r_b \times r_{N\setminus ij} \tag{5}$$

where $r_{N\setminus ij}$ denotes the probability that sub-network N can provide all functions except ones that n_r^i and/or n_r^j needs. Here, verifying if n_r^i and n_r^j have common backup VNFs is necessary to avoid double calculating.

Fig. 3 illustrates all the four cases for evaluating SFC reliability. After updating the reliability of the changed sub-networks, the reliability of the two selected VNFs needs to be updated accordingly as well. As seen from the methods described in this subsection, for each iteration the computation complexity of computing reliability is polynomial time with respect to the number of backup a VNF has. We will later show in the simulation that the approximation error is small enough to be neglected.

3.5 Backup Selection Model

For each iteration in **Algo. 1**, we need to select two VNFs and provide them a backup (Line 5). The question is how we should choose these two VNFs so that we can minimize the number of backups for a request.

Definition 1. *Define an* **improvement ratio** *as the ratio of the improvement of the network overall reliability to the network overall reliability before adding a backup.*

Theorem 2. *Provisioning a backup VNF to two primary VNFs whose reliabilities are among the lowest maximizes the improvement ratio for each case described in Section 3.4.*

We have four cases as described in Sec. 3.4, here we only prove **Thm. 2** for the second case. Note that we can similarly prove **Thm. 2** for other cases as well (in fact, case 1 is simpler, and cases 3 and 4 are based on case 2).

PROOF. Given that two VNFs n_r^i and n_r^j are selected, b as a backup VNF for n_r^i and n_r^j, and n_r^j already has backups. We also assume that the reliability of backup b is a constant. As n_r^j already has backup, we must have computed the reliability of the sub-network N which contains n_r^j already. Then the reliability of the whole network before connecting n_r^i and n_r^j with backup b is $r_{before} = r_1 r_2 ... r_k r_i r_N r_p ... r_q$, and the reliability after adding backup is $r_{after} = r_1 r_2 ... r_k (r_N(1-(1-r_b)(1-r_i)) + r_b r_{N\setminus j}) r_p ... r_q$ where $r_1 r_2 ... r_k$ and $r_p ... r_q$ are the reliability of the sub-networks not selected, and r_N is dependent on r_j. The improvement ratio u is,

$$u = \frac{r_{after} - r_{before}}{r_{before}} = -r_b + \frac{r_b}{r_i r_N}(r_N + r_{N\setminus j}) \quad (6)$$

Let's substitute $r_{N\setminus j}$ with Eq. (3), and let $A = (1-r_j)\prod_{k=1}^{M}(1-r_{j_k})$ then calculate the partial derivative with respect to r_i and r_j respectively,

$$u_{r_i} = \frac{\partial u}{\partial r_i} = -\frac{r_b}{r_i^2 r_N}(r_N + r_{N\setminus j}) \quad (7)$$

$$u_{r_j} = \frac{r_b}{r_i} \frac{(A' r_N^2 + 2\frac{\partial r_N}{\partial r_j} A r_N)(\tau - A) - (\tau - A)' A r_N^2}{(\tau - A)^2} \quad (8)$$

As the reliability is always greater than or equal to 0, and $r_i \in [0,1]$, from Eq. (7) we can easily tell that u decreases monotonically as r_i increases. While Eq. (8) is not that obvious, so we let $u_{r_j} = 0$ to compute the critical point, and get

$$2\frac{\partial r_N}{\partial r_j} = (\frac{A}{\tau - A} + 1)\frac{r_N}{1 - r_j} \quad (9)$$

Solve this partial differential equation, and get

$$r_N = \sqrt{\frac{\frac{\tau}{\prod_{k=1}^{M}(1-r_{j_k})} - (1 - r_j)}{1 - r_j}} \quad (10)$$

which means that when the equation holds true, we get the critical point. However, $\frac{\tau}{\prod_{k=1}^{M}(1-r_{j_k})} \gg 1$ since $\tau > 1$, $M \geq 1$ and both $r_N \in [0,1]$ and $r_j \in [0,1]$, so this equation can never hold, which means u is a monotone function respect to r_j in its domain. Also we can easily check $u_{r_j=1} < u_{r_j=0}$, so we can come to the same conclusion as for r_i that u decreases monotonically as r_j increases. Therefore, selecting two VNFs with lowest reliability leads to the largest improvement ratio.

4. EVALUATION

Our simulations are conducted over the 14-node NSF network. Each node of the network can provide three types of resources, namely CPU, memory and storage, with the capacity of 3500 units. We assume there are 8 types of functions in the network, and each of the physical node can provide four to six functions. The reliability of each node is randomly distributed within [0.9, 0.99]. The network traffic along each link is carried using *Optical Orthogonal Frequency Division Multiplexing* (OOFDM), because it is a cost-effective technique to achieve Terabit-per-second transmission [10], which is needed to support the huge amount of traffic flow generated by reliable SFC mapping. Each of the links has a spectrum capacity of 12THz with a spacing of 12.5GHz per spectrum slot using OOFDM.

Each SFC request consists of interconnected two to six VNFs. Each VNF demands three types of node resources, and the demand for each kind of resource is uniformly distributed between 0 and 30. Each logical link has a bandwidth demand among {10, 40, 100, 200} Gb/s with equal probability. K is set to 3 for searching shortest paths between two VNFs. For each SFC request, we select the reliability requirement among {95%, 99%, 99.9%}, similar to the ones used by Google Apps [11]. ϵ is set to 0.07 for reliability evaluation. The statistics are the average results.

We first perform simulations to compare the number of SFC requests that can be accepted with different VNF protection strategies and backup selection methods. From Fig. 4, we can see that GREP achieves the best acceptance ratio performance, and in particular, it outperforms SP and DP, both of which adopt the backup selection strategy we propose in Section 3.5, by 13.3% and 21.4%, respectively. To show the effectiveness of our proposed backup selection strategy, we also show the case where we randomly select two VNFs for each iteration to protect. Using JP with random backup selection achieves the same performance as SP with the proposed selection model. We show the efficiency of GREP in terms of the number of backup link used per request (only accepted request are considered) in Fig. 5. Because of the adoption of JP, GREP uses 31% and 14% fewer links compared with the other two methods respectively when the reliability requirement is "three nines" (i.e., 99.9%). The reason is, in JP, two VNFs are protected by one node and if these two VNFs are adjacent or share neighbors, then they can also share logical links that connect the backup VNF. Similar observations can be made when comparing the number of backup VNFs needed as shown in Fig. 6. We can find that GREP requires fewer number of backup VNFs,

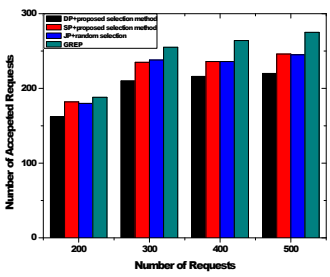

Figure 4: Number of accepted requests under different protection mechanism

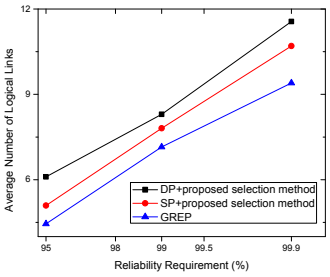

Figure 5: Average backup link number with different reliability requirement

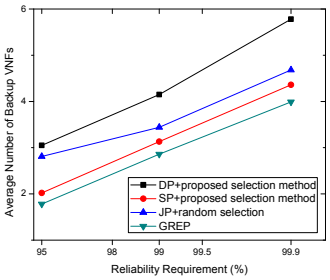

Figure 6: Average backup VNF number with different reliability requirement

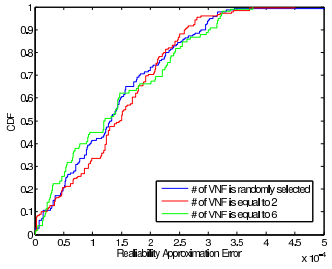

Figure 7: CDF of the error of the proposed approximate reliability calculation method

which implies the potential saving of power consumption using JP (since a fewer backup VNFs means lower static power consumption in PN such as a datacenter). Also, as illustrated in Fig. 6, using our proposed backup selection method merely can save at most 37% of physical nodes. To show the accuracy of our proposed reliability calculation method, we evaluate the *Cumulative Distribution Function* (CDF) of the approximation error of GREP when the number of request is 300 and the reliability threshold is set to 99.9%, as depicted in Fig. 7. We can observe that 98% of the error is smaller than 3.5×10^{-4}, which demonstrates the effectiveness of using GREP to predicting the service reliability.

5. CONCLUSION

In NFV, it is critical to provide effective reliability guarantee with efficient and robust resource allocation in order to support the shared middlebox platform. In this paper, we have proposed GREP for reliable SFC mapping in NFV networks, which can minimize the resources allocated to SFC requests while meeting clients' SLA requirement. We have validated our design through extensive simulations and demonstrated that it can achieve significant performance improvement compared to the traditional protection mechanisms. Meantime, we have shown that GREP is able to evaluate service reliability with a negligible approximation error in polynomial time. As for our future work, we plan to extend the proposed algorithm to (1) jointly optimize the selection of primary and backup mapping nodes, knowing that backups may eventually be needed; (2) share redundancy across multiple SFC requests to further increase resource utilization; (3) take dynamic traffic demands into consideration when devising a backup plan.

6. REFERENCES

[1] Alcatel Lucent. Network Functions Virtualization-Challenges and Opportunities. 2013.

[2] P. Gill, N. Jain, and N. Nagappan. Understanding network failures in data centers: measurement, analysis, and implications. In *ACM SIGCOMM*, 2011.

[3] R. Potharaju, and N. Jain Demystifying the Dark Side of the Middle: A Field Study of Middlebox Failures in Datacenters. In *ACM IMC*, 2013.

[4] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun. Survivable virtual infrastructure mapping in a federated computing and networked system under single region failures. In *IEEE GLOBECOM*, 2010

[5] W. Yeow, C. Westphal, and U. Kozat. Designing and embedding reliable virtual infrastructure. *SIGCOMM CCR*, 2011

[6] M. Rahman, and R. Boutaba. SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization. *IEEE TNSM*, 10(2): 105-118, 2013

[7] M. Yu, Y. Yi, J. Rexford, M. Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *SIGCOMM CCR*, 2008

[8] M. Jerrum, and A. Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. *Approximation algorithms for NP-hard problems*: 482-520, 1996

[9] Q. Zhang, M. F. Zhani, M. Jabri, and R. Boutaba. Venice: Reliable virtual data center embedding in clouds. In *IEEE INFOCOM*, 2014

[10] N. Cvijetic, M. Cvijetic, M. F. Huang, and E. Ip. Terabit optical access networks based on WDM-OFDMA-PON. *IEEE/OSA JLT*, 30(4): 493-503, 2012

[11] Google Apps Service Level Agreement. http://www.google.com/apps/intl/en/terms/sla.html