ASA: Adaptive VNF Scaling Algorithm for 5G Mobile Networks

Yi Ren†, Tuan Phung-Duc‡, Yi-Kuan Liu‡, Jyh-Cheng Chen‡, Fellow, IEEE, and Yi-Hao Lin‡

†School of Computing Science, University of East Anglia, Norwich, U.K.
‡Faculty of Engineering, Information and Systems, University of Tsukuba, Ibaraki, Japan

Abstract—5G mobile networks introduce Virtualized Network Functions (VNFs) to provide flexible services for incoming huge mobile data traffic. Compared with fixed capacity legacy network equipment, VNFs can be scaled in/out to adjust system capacity. However, hardware-based legacy network equipment is designed dedicatedly for its purpose so that it is more efficient in terms of unit cost. One challenge is to best use VNF resources and to balance the traffic between legacy network equipment and VNFs. To address this challenge, we first formulate the problem as a cost-performance tradeoff, where both VNF resource cost and system performance are quantified. Then, we propose an adaptive VNF scaling algorithm to balance the tradeoff. We derive the suitable VNF instances to handle data traffic with minimizing cost. Through extensive simulations, the adaptive algorithm is proven to provide good performance.

Index Terms—Dynamic Auto Scaling Algorithm, Network Function Virtualization (NFV), Virtual EPC, Cloud Networks, 5G, Modeling and Analysis

I. INTRODUCTION

Network function virtualization (NFV) is one of the key features introduced by recent 5G mobile network standard. NFV provides flexible and fine grained solutions to meet heterogeneous type and huge amounts of user traffic demands, and is changing the way of how mobile operators increase the capacities of their network infrastructures. The NFV technique virtualizes special purpose hardware resources as virtualized network function (VNF) instances so that software-based network functions can run on the general purpose equipment, deployed in the mobile providers’ cloud. Different from legacy network equipment, VNF instances can be scaled-out/in (turned on/off) to adjust network capacity dynamically, which can save operation cost and increase resource utilization. The great flexibility of such VNF autoscaling strategies forms a cost-performance tradeoff: system performance is improved by adding more VNF instances while operation cost is decreased by reducing the number of VNF instances.

A design challenge for such autoscaling strategies in 5G networks is to take legacy network equipment into consideration. Indeed, although NFV provides such a flexible and fine-grained property, legacy equipment is generally more efficient with respect to unit cost, i.e., cost-performance (c/p) ratio. In other words, legacy equipment and virtualized equipment have different c/p ratio (service capacity per price unit). Specifically, legacy network equipment is usually hardware-based and is dedicated designed and optimized for its purpose, e.g., Barracuda X Series, Juniper Networks SRX Series for standalone hardware firewalls, ternary content-addressable memory (TCAM) in software define network (SDN) switches, etc. Whereas, virtualized equipment is virtualized as VNF instances in general purpose equipment, which is generally slower in service rate and has less c/p ratio than the former one [1]. Also, when allocating hardware resources (e.g., CPU, memory, etc.) to a VNF instance, different configuration leads to different service capacities for a VNF instance, which has significant impacts while designing autoscaling strategies. Existing research in 4G/5G networks (e.g., [2]–[4]) or cloud networks (e.g., [5]–[7]) usually ignore this issue.

In this paper we present ASA, an Adaptive Scaling Algorithm, to well balance the cost-performance tradeoff while maintaining an acceptable level of performance for 5G mobile networks. We first propose an analytical model that considers and quantifies the different service capacity issue and the impact of VNF capacities. Based on the model, mobile operators can configure the system parameters to evaluate their autoscaling strategies. Also, in our proposed analytical model, we use a novel recursive algorithm to reduce the computational complexity. The computational cost to solve a Markov chain with $M$ states is reduced from $O(M^3)$ to $O(M)$. The reduction is significant so that mobile operators can quickly evaluate the performance of their auto-scaling strategies, saving on cost and time.

II. PROPOSED ASA: ADAPTIVE VNF SCALING ALGORITHM

In this section, we present the system model in Section II-A. The goal of ASA is to reduce mobile providers’ operation cost and the probability of service level agreements (SLAs) violations while providing acceptable levels of performance. Here, the operation cost we focus is VNF instances power consumption while the performance is evaluated by the performance metrics.
A 5G core network system consists of VNFs and legacy network entities, e.g., mobility management entity (MME). Generally, legacy network entities are difficult to be flexibly adjusted but are always running and have better performance than VNF instances. In contrast, a VNF instance can be turned on when the system needs them, but they have to consume extra power and need some setup time to be turned on, leading to extra cost for the mobile provider. With finite budget, a mobile provider should carefully plan and utilize the VNF instances to add more capacity to the system. Here, we consider an algorithm that VNF instances are turned on and off to save the cost if there is no job in the queue waiting for processing.

We assume that user request arrivals follow Poisson distribution with rate λ, as shown in Fig. 1. 5G evolved packet core (EPC) system is modeled as two parts: (1) always-on n₀ legacy network entity with average capability μ₁, and (2) VNF instances each with μᵥ service rate. Due to finite budget, we assume that the mobile provider can turn on at most k VNF instances at the same time. That is, totally at most N servers, where \( N = n₀ + k \). Each VNF instance (server) serves one job at a time and its service rate follows the exponential distribution. Therefore, mobile provider can set different service rate for their legacy network equipment and VNFs by themselves, respectively. Moreover, a VNF instance needs an extra setup time to be available to process user requests, which is assumed to be an exponentially distributed random variable with mean value \( 1/\alpha \). The system queue has limited capacity \( K \), i.e., the maximum of \( K \) jobs can be accommodated in the system. Also, service discipline is First-Come-First-Served (FCFS) for those jobs waiting for processing. A list of notations can be found in Table I.

### Table I: List of Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>The number of servers in server center</td>
</tr>
<tr>
<td>( K )</td>
<td>The number of maximum jobs can be accommodated in the system</td>
</tr>
<tr>
<td>( k )</td>
<td>The number of VNF instances</td>
</tr>
<tr>
<td>( C )</td>
<td>System cost-performance tradeoff</td>
</tr>
<tr>
<td>( W )</td>
<td>Average response time per job</td>
</tr>
<tr>
<td>( W_q )</td>
<td>Average response time in the queue per job</td>
</tr>
<tr>
<td>( P_b )</td>
<td>Average system blocking probability</td>
</tr>
<tr>
<td>( S )</td>
<td>Average VM cost</td>
</tr>
<tr>
<td>( w_1 )</td>
<td>Weight factor for ( W_q )</td>
</tr>
<tr>
<td>( w_2 )</td>
<td>Weight factor for ( S )</td>
</tr>
<tr>
<td>( w_3 )</td>
<td>Weight factor for ( P_b )</td>
</tr>
<tr>
<td>( n_0 )</td>
<td>The number of permanently operative servers</td>
</tr>
<tr>
<td>( U_i )</td>
<td>The up threshold to control the reserve sub-blocks</td>
</tr>
<tr>
<td>( D_i )</td>
<td>The down threshold to control the reserve sub-blocks</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Job arrival rate</td>
</tr>
<tr>
<td>( \mu_1 )</td>
<td>Service rate for legacy server</td>
</tr>
<tr>
<td>( \mu_v )</td>
<td>Service rate for VNF instance</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Setup rate for each virtual server</td>
</tr>
</tbody>
</table>

The system performance is evaluated by two performance factors: the average response time in the queue per request, \( W_q \), and the system blocking probability \( P_b \). The operation cost is evaluated by the average number of VNF instance consuming power, \( S \). We define them as follows.

- **The average response time in queue** \( W_q \) is defined as the average waiting time of a user request in queue. In other words, it means how long time a job request can be served.
- **The average blocking probability in system** \( P_b \) is defined as the probability of a job blocked by the system.
- **The average number of running VNF instances** \( S \) denotes the operation cost of virtual equipment.

### C. The Proposed ASA

According to the number of waiting user requests in the system queue, the VNF instances will be added (or removed) by ASA. To control the number of VNF instances in the 5G core networks, we assume that \( n_i = n_0 + 1 \) and \( n_i = n_{i-1} + 1 \) (\( i = 1, 2, \ldots, k \)), where \( n_k \) denotes \( k \) VNF instances are running. Here, the maximum number of network entity is \( k + n_0 = N \) by the definition. That is, \( n_k = N \).

ASA utilize two thresholds, up and down, or \( U_i \) and \( D_i \) to control the number of running VNF instances, where \( i = 1, 2, \ldots, k \).

- **\( U_i \)** denote power up the \( i\)-th VNF instances: When the \( i\)-th VNF instance is turned on and the number of user requests in the system increases from \( U_{i-1} \) to \( U_i \), the VNF instance is powered up after a setup time to add more capacity to the system. During the setup time, a VNF instance cannot serve any
user requests, but consumes power (or money for renting cloud services). Here, we choose \( U_i = n_i \).
It is equivalent to that when the number of user requests increases from \( n_{i-1} \) to \( n_i \), the \( i \)-th VNF instance is powered up.

- \( D_i \) denote power down the \( i \)-th VNF instances: When the \( i \)-th VNF instance is operative, and the number of user requests in the system drops from \( D_{i+1} \) to \( D_i \), then the VNF instance is powered down immediately to save the power (or money for renting cloud services). Here, we choose \( D_i = n_{i-1} \). Note that it is equivalent to that when the number of user requests drops from \( n_i \) to \( n_{i-1} \), we turn off the \( i \)-th VNF instance.

To address the tradeoff, we quantify the performance metrics \( W_q, S, \) and \( P_b \) in our technique report [8]. Thus, the system cost-performance function \( C \) has the form

\[
C = w_1 W_q + w_2 \mu v \mu l + w_3 P_b,
\]

(1)

where coefficients \( w_1, w_2, \) and \( w_3 \) denote the weight factors for \( W_q, S, \) and \( P_b \), respectively. Increasing \( w_1 \) (or \( w_2, w_3 \)) emphasizes more on \( W_q \) (or \( S, P_b \)). Here, we do not specify either \( w_1 \) or \( w_2 (w_3) \) due to the fact that such a value should be determined by mobile provider and must take management policies into consideration.

With closed-form solutions in [8], formula (1) can be rewritten as

\[
\begin{align*}
\arg \min_{k, \mu v} & \quad C = w_1 W_q + w_2 \mu v \mu l + w_3 P_b, \\
\text{subject to} & \quad 0 < \beta < \beta' .
\end{align*}
\]

(2)

where \( \beta \in \{ W_q, S, P_b \} \). One can easily find the local minimum when \( C'' = 1 \) and \( C'' > 0 \) hold. Algorithm 1 presents how to find optimal service rate of VNF instances \( \mu v \) and optimal maximum number of VNF instances \( k \) for minimizing the cost function (2) based on the constraints set by mobile operators.

### III. SIMULATION AND NUMERICAL RESULTS

The analytical results of model are cross-validated by extensive simulations by using ns-2, version 2.35. Although simulation results are special case for the model, which were used to validate our analysis model and demonstrate the numerical results, one easily replace these parameters with other values. In other words, mobile operators can configure different settings to test their autoscaling strategies, saving on cost and time.

It is important for mobile providers to maintain core network performance and reduce SLAs violations with given budget. In the previous sections, we have proposed the analytical model and cross-validated with extensive ns2 simulations. Here, we show some results of the cost function (2) on selecting optimal \( k \) and \( \mu v \).

Figs. 2, 3 show the results of performance metrics \( S, W_q, \) and \( P_b \) with respect to \( k \) and \( \mu v \) according to (2).

#### Algorithm 1 Cost-minimization algorithm

**Input:** system capacity \( K 

**Output:** optimal \( \mu v \) and optimal \( k 

1: Initialize \( \mu v \) as 0.01, \( k \) as 0, \( \Delta C \) as maximum integer
2: Set learning rate \( \beta, \gamma \)
3: Update \( S, W_q, P_b \)
4: Compute \( C = w_1 W_q + w_2 \mu v \mu l + w_3 P_b \)
5: Set \( \mu v(new) = \mu v(old), k_{new} = k_{old} = 1 \)
6: Set \( \mu v(new) = \mu v(old) + 0.01, k_{new} = k_{old} + 1 \)
7: \( \Delta C \) not converge do
8: \( k > K - n_0 \) then
9: break
10: end if
11: Update \( S, W_q, P_b \)
12: Compute \( C = w_1 W_q + w_2 \mu v \mu l + w_3 P_b \)
13: \( temp1 = \mu v(new) - \beta \frac{\partial C}{\partial \mu v} \)
14: \( temp2 = k_{new} - \gamma \frac{\partial C}{\partial k} \)
15: Set \( \mu v \mu l + w_3 P_b \)
16: Set \( \mu v \mu l + w_3 P_b \)
17: \( \Delta C = | C - C | \)
18: \( C = C \)
19: end while
20: return optimal \( \mu v \) and optimal \( k \)

The right y-axis in red color denotes \( \beta \), i.e., \( S, W_q \) or \( P_b \) configured by a mobile provider. The left y-axis in blue color is the value of cost function \( C \). The mobile provider can given the constraint value \( \beta \) and obtain their optimal \( k \) and \( \mu v \) through these figures.

Take Fig. 2(a) as an example. Note that \( \beta \) is \( S \) and the constraint is \( S < S' \). We can see that VNF cost \( S \) increases while the number of VNF instances \( k \) grows. Increasing \( k \) leads to higher \( S \) yet provides a smaller average waiting time \( W_q \) and better QoS for users. This figure shows that \( S \) is 20, which is corresponding to minimal value of cost function \( C \).

If the mobile provider configures the constraint value \( \beta \) higher than 20, ASA sets the optimal \( k \) as 40 since it makes the cost function \( C \) have minimal value. That means that ASA balances the cost-performance tradeoff with the given constraint value \( \beta \). Otherwise, ASA can find \( k \) which allows \( S \) satisfy the constraint \( \beta \) and let the value of cost function \( C \) as small as possible. In this case, ASA tries to provide acceptable performance and meets the constraint on metric \( S \) at the same time. Similarly, the mobile provider can apply ASA to metrics \( W_q \) and \( P_b \).

To find optimal VNF service rate \( \mu v \), we take another example as shown in Fig. 3(a). Here, \( \beta \) is \( S \) and the constraint is \( S < S' \). Also, the \( S \) increases as \( \mu v \) grows. The curve of cost function \( C \) goes down before \( \mu v \) is close to 3.3. Then it grows sharply after \( \mu v \) is larger than 3.3. The optimal VNF service rate is corresponding to
The minimal value of cost function $C$. In this case, minimal value of cost function $C$ needs the constraint $\beta \geq 13.5$. Again, if the mobile provider configures the constraint value $S < 13.5$, ASA can obtain optimal $\mu_v$ which minimizes $C$ while controlling the VNF cost $S$ under a given budget. Otherwise, ASA will find a near-optimal $\mu_v$ by following the blue curved line by decreasing the value of $\mu_v$ until the VNF cost $S$ is under the constraint $\beta$.

Here we only demonstrate ASA on the performance metric $S$ due to page limit. Similar results can be observed from Figs. 2(b,c) and Figs. 3(b,c). Also, ASA can be applied on performance metrics such as $W_q$ and $P_b$ in the same way.

IV. CONCLUSIONS

In this paper, we proposed ASA for addressing the autoscaling cost-performance tradeoff in 5G mobile networks. It is the first work on discussing the impacts of different service rate of legacy network and that of VNF instances in this perspective. We quantified the impacts and a set of performance metrics using a lightweight analytical model. The model improves traditional Markov chain method by using a novel recursive algorithm. The computational complexity is reduced from $O(K^2)$ to only $O(K \times k)$. The reduction is significant. The model gives theoretical insights to mobile operators while designing autoscaling strategies in 5G mobile networks, saving on cost and time. Moreover, we presented a cost function as an example on using the model to develop optimal autoscaling strategies. It provides a guideline for mobile provider to design optimization strategies and analyze their core networks in a systematic way.

REFERENCES


