IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 24, 2021, accepted May 28, 2021, date of publication June 3, 2021, date of current version June 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3085955

A Queueing Model for Threshold-Based Scaling of UPF Instances in 5G Core

CSABA ROTTER[1] AND TIEN VAN DO[2]
[1]Nokia Bell Labs, 1083 Budapest, Hungary
[2]Department of Networked Systems and Services, Budapest University of Technology and Economics, 1111 Budapest, Hungary

Corresponding author: Tien Van Do (do@hit.bme.hu)

The work of Tien Van Do was supported by the NRDI Fund based on the charter of bolster issued by the National Research, Development and Innovation Office through the Ministry for Innovation and Technology.

ABSTRACT The fifth-generation networks User Plane Function (UPF) provides necessary procedures to connect end-devices and data networks. A Protocol Data Unit (PDU) session between specific user equipment (UE) and the UPF is established before data transmission in 5G networks. The UPF software can be packed into either a virtual machine (VM) or a container image and instantiated in service providers' cloud infrastructure. Each UPF instance may handle several concurrent PDU sessions from various users. Operators should control the number of UPF instances to save resource consumption in the provision of Quality of Service (QoS). This paper presents a queueing model for a scenario where a threshold-based algorithm controls the number of UPF instances depending on users' traffic. We derive a method to compute the steady-state probabilities and performance measures efficiently. We investigate the performance of the UPF scaling algorithm in various scenarios with the assumption of specific hardware. Numerical results show that scaling UPF instances can lead to the good utilization of the system resource.

INDEX TERMS 5G, core, PDU session, UPF, scaling, queueing analysis.

I. INTRODUCTION

The fifth-generation networks are expected to provide the service for customers with different requirements from vertical industries [1]–[5]. 5G core consists of multiple Service Based Architecture (SBA) elements [1], [3]. The control plane of 5G systems includes the Access and Mobility Management Function (AMF) and and the Session Management Function (SMF) that performs authentications, mobility-related procedures and UEs' requests for the establishment of communication flows between UEs and data networks. The tasks of the 5G data plane are carried out by 5G base stations and the UPF, which provides necessary procedures to convey data flows between end-devices and various data networks. After authentication steps, a user equipment requests a PDU session with the use of the signaling protocol messages for the data communication to a specific data network and the SMF decides the acceptance based on the actual conditions of the system and the QoS requirements of the PDU session. Upon acceptance, the PDU session is tunnelled through the transport network to the UPF, and the communication between the UE and the data network can start.

The associate editor coordinating the review of this manuscript and approving it for publication was Hosam El-Ocla.

The UPF is implemented in software and can be packed into either a VM or a container image. Service providers launch UPF instances in their cloud infrastructure to serve customers. In 5G networks, highly changing dynamics of requests for PDU sessions are generated by subscriber equipment. To guarantee QoS, each UPF instance should handle a limited number of concurrent PDU sessions, therefore contention for resources is expected. Operators may launch new UPF instances when there are more requests for PDU sessions and terminating idle ones when few customers need PDU sessions. That is, operators may apply threshold-based scaling algorithms for managing UPF instances. Motivated by the need for an efficient method for evaluating threshold-based scaling solutions, we deal with a queueing model for a threshold-based algorithm that can control the number of UPF instances according to the need for PDU sessions. To our best knowledge, no existing works on the queueing analysis for the scaling of the 5G UPF have been presented so far. The main contributions of this paper are the proposal of a queueing model for a threshold-based algorithm that manages the number of UPF instances depending on users' traffic and the efficient computational method of low complexity for the steady-state probabilities and performance measures. The complexity of our solution is linear to the size of the state

space. Numerical results obtained by the queueing analysis show that the threshold-based scaling algorithm can lead to the efficient usage of the resource.

The rest of the paper is organized as follows. Related works are reviewed in Section II. A scaling issue and a threshold-based scheduling algorithm for the 5G user data plane function are presented in Section III. A queueing model is proposed in Section IV where we derive the steady-state probabilities of the system. Numerical results are presented in Section V. Finally, Section VI concludes our paper.

II. RELATED WORKS

Virtualization and container technology advancement in the past decade has initiated a paradigm shift on the development, deployment and operation (DevOps [6], [7]) of telecommunication functions and service. The provision of services based on the DevOps concept comes with a series of decisions related to the choice of an appropriate computing cluster size and a scaling task that controls the number of software instances in response to customers' demands. The management of computing and network resource (including scaling measures to either add or remove resource to the existing deployment) has been investigated in the context of cloud computing and telecommunications as well.

A systematic review of autoscaling methods for applications in a cloud environment was summarized by Lorido-Botran *et al.* [8]. Jaro *et al.* [9] investigated the resource dimensioning aspects of a specific Telecommunication Application Server that was implemented as the particular set of Virtual Network Functions (VNF) to handle several million busy hour call attempts. Tang *et al.* [10] studied a traffic forecasting method for scaling VNF instances running on VMs, and applied a mixed-integer linear programming for the placement of VNF instances. Kumar *et al.* [11] considered network packet processing aspects related to the virtualization of UPF in public clouds, and applied single root input/output virtualization (SR-IOV), a technique for the isolation of PCI Express cards, to speed up the packet processing for the scaling of UPF instances. Herrera and Moltó [12] proposed a bio-inspired algorithm based on the analogue of cell adaptation, cell death and cell reproduction for the container autoscaling. Their simulation results show that their approach could limit the peak time and reduce the response time the connection between cell-reproduction and container orchestration platforms due to the over-provisioned behavior.

Guo *et al.* [13] solved a virtual machine-to-physical (VM-to-PM) machine assignment problem within the framework of stochastic bin packing. Their proposed solution guides the packing of VM-to-PM and the VM autoscaling meanwhile automatically adapts to application resource needs. Gandhi *et al.* [14] performed experiments about the impact of horizontal and vertical VM scaling on cost, performance, and provisioning times in an OpenStack environment. The authors applied Kalman filtering to estimate the service times and analyzed several scaling options with different VM sizes. Taherizadeh and Stankovski in [15] carried out experiments with an autoscaling method that considers the container-level monitoring and the application-level monitoring. For the monitoring purpose, they applied the SWITCH monitoring system from the SWITCH project (http://www.switchproject.eu/). Wu *et al.* [16] presented a design of a distributed key-value store called Anna, that has the autoscaling capability to add and terminate service nodes in response to load dynamics. Baresi *et al.* [17] applied the control theory approach based the formulation of a nonlinear, time-invariant dynamic system that describe the response time as the function of the assigned cores and the request rate. They also carried out some experiments to show the viability of the scaling approach based on the control theory. Zhang *et al.* [18] implemented their orchestration platform for containers that host services for smart devices. They also performed experiments to demonstrate the autoscaling feature of their platform. Casalicchio [19] carried out measurements about the scaling of Kubernetes Pods and argued that the scaling should consider the Quality of Service aspects. Gervásio *et al.* [20] in their hybrid autoscaling proposal, combined a self-adaptive prediction and reactive approach for optimal configuration of the threshold values for scaling operations. Ullah *et al.* [21] presented the combination of predictive and reactive approach where Cartesian genetic programming based neural network is used for resource estimation and a rule-based scaling.

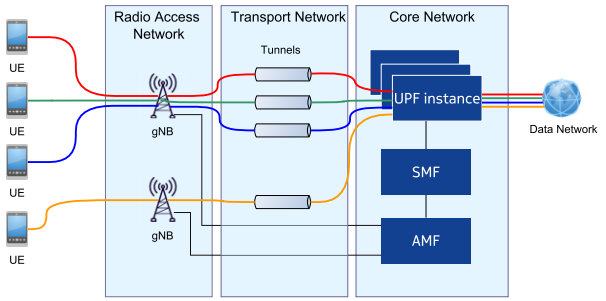


FIGURE 1. The role of the 5G UPF.

From the related literature review, it is observed that scaling algorithms reported in most of the literature works so far control the number of VM, VNF, container instances with the use of some thresholds [14]–[20]. However, to our best knowledge, no existing work on the queueing analysis for the scaling of 5G UPF instances based on threshold algorithms has been presented. Our queueing model provides a quick evaluation of scaling algorithms based on two thresholds, which could be used to set parameters and establish benchmarks for various scenarios.

III. SCALING ISSUE IN THE OPERATION OF 5G UPF

A. 5G UPF ENVIRONMENT

The 5G architecture consists of Radio Access Networks and a 5G Core part (see Figure 1). User Equipment (UEs) are are connected to a Radio Access Network (RAN) based on 5G base stations (Next Generation NodeB gNB) [1]–[4], [22]. The transport network referred to as backhaul is responsible

for ensuring the RAN and Core Network connectivity. There are diverse wireless, wireline or optical solutions for the backhaul transport network [23]. The 5G standards support the connectivity of UE with various types (IP, Ethernet, unstructured) of external data networks. The Third Generation Partnership Project (3GPP) designed the 5G core based on the Service Based Architecture and the total control and user plane separation (CUPS). That is, the 5G core consists of the control plane and the data plane. The control plane of 5G systems includes the Access and Mobility Management Function (AMF) and and the Session Management Function (SMF) that manages the authentication of users, establishes the data connection between UEs and data networks, and executes necessary procedures to handle the mobility of UEs.

5G base stations and the UPF perform the tasks of the 5G data plane, that is, they provide necessary procedures to convey data flows between end-devices and data networks. Before the communication of a specific UE and a data network, a PDU session should be started and handled by UPF. UPF is implemented in software and can be executed inside either a virtual machine or containers (termed as a UPF instance). SMF is responsible for managing user sessions, assigns a PDU session to an appropriate UPF instance. To support the mobility management of UE and hide the mobility of UE from external data networks, General Packet Radio Service (GPRS) tunnels are established between gNB and a specific UPF instance that handles data flows between UE and a specific data network.

B. THRESHOLD-BASED SCALING ALGORITHM

If the traffic load increases, more UPF instances can be started. Also, UPF idle instances can be terminated when the traffic is low. This section presents a threshold-based scaling algorithm that can be executed inside operations, administration and management (OAM) to control the number of UPF instances based on traffic. The threshold-based scaling algorithm makes scaling decisions based on the information about PDU sessions at SMF and UPF instances.

Since a limited resource can be allocated for one virtual machine or container, we assume that one UPF instance can handle a maximum of C number sessions. To manage the resource for the Quality of Service provision, a best practice approach for operators classifies UPF. Each group handle PDU sessions with the same requirement. Due to the limit on the available capacity of physical servers, the maximum number of UPF instances that can be launched is L. Also, the operator may set the minimum number (M) of UPF instances that should be established. Let $I(t)$ denote the number of sessions that are being served and $J(t)$, $(M \leq J(t) \leq L)$, be the number of UPF instances at time instant t. The maximum number of sessions that can be served at time t is $N_{J(t)} = J(t) \times C$.

A scaling algorithm applies two thresholds (T_1 and T_2) to control the number of UPF instances.

- If $(I(t) = J(t) \times C - T_1 - 1)$ and $J(t) < L$ upon a request for a new session arrival, the request (termed the scaling-out action) for a new instance is initiated. Note that if the request is fulfilled, $J(t) = J(t) + 1$.
- If the number of free slots would be equal to T_2 and $J(t) \geq M+1$ upon the departure of a session, the request is launched for the termination of an idle UPF instance. Note that if the request (termed the scaling-in action) is fulfilled, $J(t) = J(t) - 1$. It is worth emphasizing that session migration should be performed to increase the chance of having an idle instance.

TABLE 1. Parameters of the operation rule.

Notation	Description
T_1	Threshold value for the scaling-out decision
T_2	Threshold value for the scaling-in decision
C	Number of maximum sessions served by one UPF instance
M	Initial number of UPF instances
L	Maximum number of UPF instances

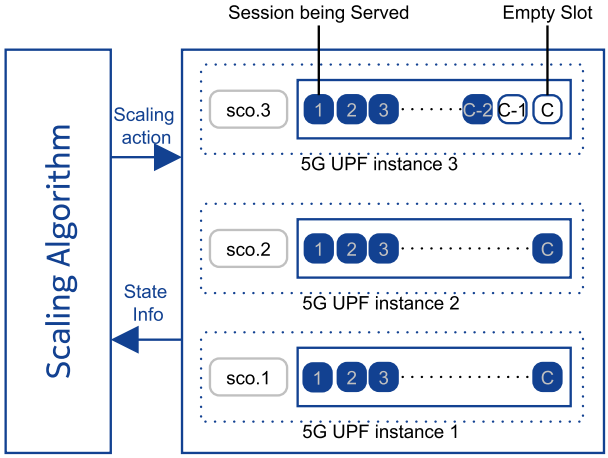


FIGURE 2. An example for the assignment of sessions to UPF instances.

When $J(t) = j$ UPF instances are deployed, the scaling-out action happens if there are $I(t) = N_j - T_1 - 1$ PDU sessions upon the arrival of a new session in the system. The scaling-out action results in a situation with $j + 1$ UPF instances and $N_j - T_1$ served sessions. The scaling-in action for a situation with $j+1$ UPF instances takes place when one of ongoing $N_{j+1} - T_2 + 1$ PDU sessions departs. Therefore, $N_{j+1} - T_2 < N_j - T_1 - 1$ should hold to avoid the oscillation (the scaling-in/out actions may immediately follow each other). This condition is equivalent to $T_2 - C - T_1 > 1$ because $N_{j+1} - N_j = C$.

Figure 2 shows an example where $J(t) = 3$ UPF instances are started, (instances $sco.1 - sco.3$) and there are $I(t) = 3 \times C - 2$ sessions (plain circles). Instance $sco.3$ has two empty slots left for arriving sessions (empty circle).

IV. ANALYTICAL MODEL

We assume that the arrival of new sessions follows the Poisson process with rate λ and the session durations are exponentially distributed with mean $1/\mu$. Then the system is described by a two-dimensional Continuous Time Markov Chain (CTMC), $\{(I(t), J(t)), t \geq 0\}$. The following types

TABLE 2. Summary of main notations.

Notation	Description
$I(t)$	Number of sessions that are being served
$J(t)$	the number of deployed UPF instances at time instant t
λ	The rate of session arrivals
μ	The reciprocal of the average holding time of sessions
$N_j = j \times C$	The maximum free slots when j UPF instances are deployed for $M \leq j \leq L$
$p_{i,j}$	The steady state probability of state (i, j)

of transitions are possible between the states of CTMC $\{(I(t), J(t)), t \geq 0\}$.

- State transition $(i, j) \Rightarrow (i+1, j+1)$ for $i+1 = N_j - T_1$ and $M \leq j < L$, is caused by the acceptance of a new user session and the launch of a new UPF instance based on the operation rule.
- State transition $(i, j) \Rightarrow (i+1, j)$ for
  - either $N_j - T_2 + 1 < i + 1 \leq N_j - T_1 - 1$ and $M < j < L$,
  - or $0 < i + 1 \leq N_M - T_1 - 1$ and $j = M$,
  - or $N_L - T_2 + 1 < i + 1 \leq N_L$ and $j = L$

  happens when there is a free slot and a new user session arrives, and no scaling is performed.
- State transition $(i, j) \Rightarrow (i-1, j)$ for
  - either $N_j - T_2 + 1 \leq i - 1 < N_j - T_1 - 1$ and $M < j < L$,
  - or $0 \leq i - 1 < N_M - T_1 - 1$ and $j = M$,
  - or $N_L - T_2 + 1 \leq i - 1 < N_L$ and $j = L$

  takes place when a user session departs and no scaling is performed.
- State transition $(i, j) \Rightarrow (i-1, j-1)$ for $i - 1 \leq N_j - T_2$ and $M < j \leq L$, is due to the departure of a user session and the termination of a UPF instance based on the operation rule.

As a consequence, the state space S of CTMC $\{(I(t), J(t)), t \geq 0\}$ is expressed as

$$\begin{aligned} S = &\{(i, M) : \ 0 \leq i \leq N_M - T_1 - 1\} \\ &\cup \{(i, j) : \ N_j - T_2 + 1 \leq i \leq N_j - T_1 - 1, M < j < L\} \\ &\cup \{(i, L) : \ N_L - T_2 + 1 \leq i \leq N_L\}. \end{aligned}$$

The states when j UPF instances are launched are termed as level j states. As the number of the states is $N_M - T_1 + (T_2 - T_1 - 1)(L - M - 1) + T_2 = M \times C + T_2 - T1 + (T_2 - T_1 - 1)(L - M - 1)$, the direct methods [24] to find the steady state probabilities have the complexity of at most $\mathcal{O}((M \times C + T_2 - T1 + (T_2 - T_1 - 1)(L - M - 1)^3)$, which is the complexity of solving a system of linear equations. In what follows, we present a derivation that leads to a solution with the complexity of $\mathcal{O}(M \times C + T_2 - T1 + (T_2 - T_1 - 1)(L - M - 1))$.

If j, $M < j < L$, UPF instances are launched, due to the operation rule there are two special states.

- State $(N_j + C - T_2, j)$ can be reached from state $(N_j + C - T_2 + 1, j + 1)$ due to the departure of a session and the scaling-in action performed to terminate one UPF instance.
- State $(N_j - C - T_1, j)$ is the result of the scaling-out action from state $(N_j - C - T_1 - 1, j - 1)$ due to the arrival of a session.

In this paper, we distinguish two cases based on the relation between $N_j + C - T_2$ and $N_j - C - T_1$. In the first case, $N_j + C - T_2 \geq N_j - C - T_1$ holds (i.e., $T_2 - T_1 \leq 2C$) The second case corresponds to $T_2 - T_1 > 2C$. The state transition diagrams of CTMC $\{(I(t), J(t)), t \geq 0\}$ in two cases are illustrated in Fig 3 and Fig 4, respectively.

Let us denote the steady state probabilities of CTMC $\{(I(t), J(t)), t \geq 0\}$ as follows

$$p_{i,j} = \lim_{t \to \infty} \Pr(I(t) = i, J(t) = j), \quad (i, j) \in S.$$

To determine the steady state probabilities, we apply Proposition 1.

Proposition 1: All the probabilities ($p_{i,j}$'s) can be expressed in $p_{N_L - T_2 + 1, L}$.

A proof for case $T_2 - T_1 \leq 2C$ is presented in Section IV-A, while a proof for case $T_2 - T_1 > 2C$ is show in Appendix.

A. PROOF FOR CASE $T_2 - T_1 \leq 2C$

In what follows we provide a proof for Proposition 1.

From Figure 3, we can observe that states can be classified into subsets S_1, S_2, S_3, S_4, S_5 and S_6.

- Subset $S_1 = \{(N_j - T_2 + 1, j) : M < j \leq L\}$ includes states where the scaling-in decision takes place upon the departure of a session. The balance equation of these states can be written as

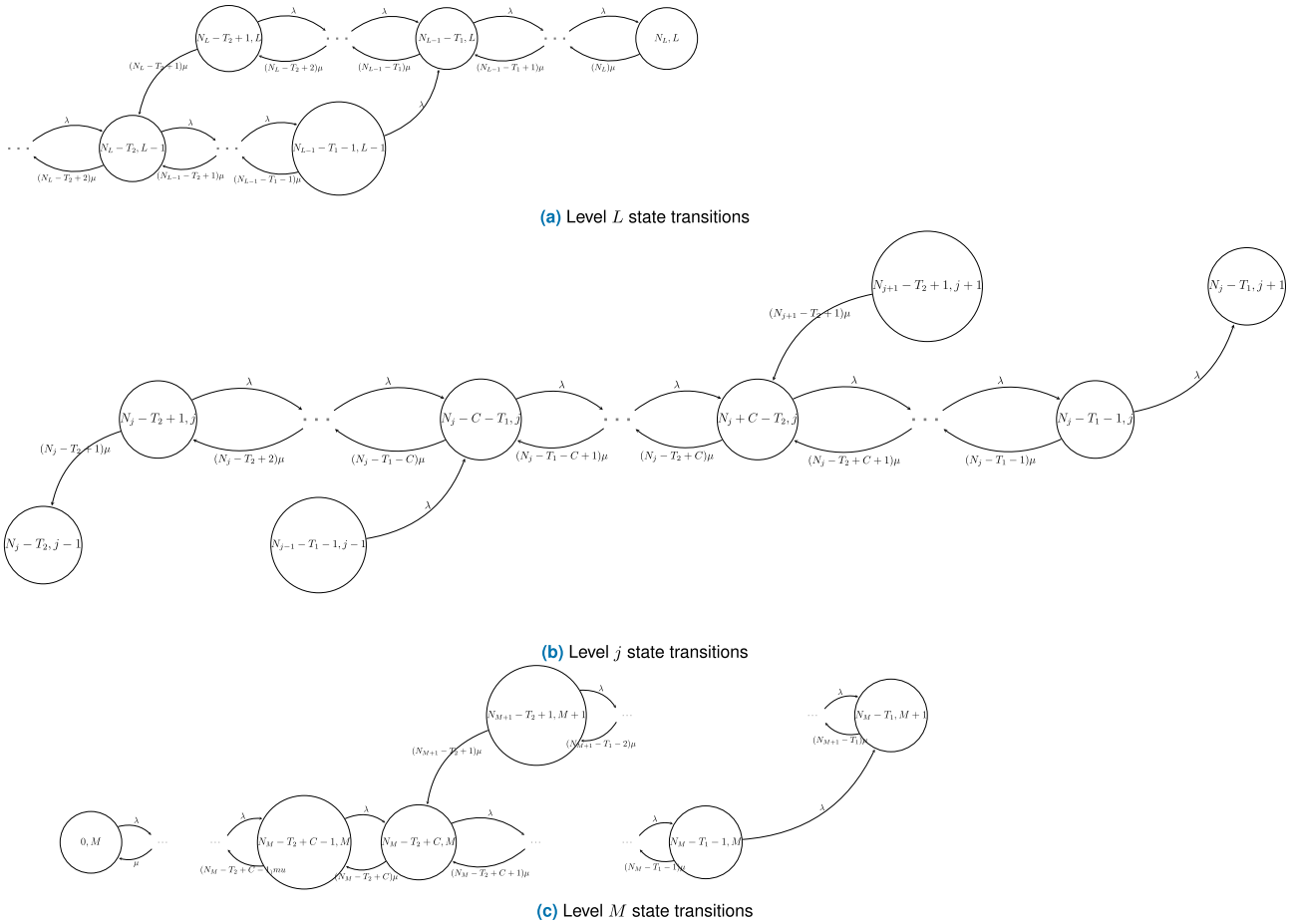
$$p_{k,j}(\lambda + k\mu) = p_{k+1,j}(k + 1)\mu, \quad (k, j) \in S_1. \quad (1)$$

- Subset $S_2 = S - S_1 - S_3 - S_4 - S_5 - S_6$ consists of state (k, j), $M \leq j < L$, where only either session request or departure happens.
  The balance equation of these states can be written as

$$p_{k,j}(\lambda + k\mu) = p_{k-1,j}\lambda + p_{k+1,j}(k + 1)\mu, \quad (k, j) \in S_2. \quad (2)$$

- Subset $S_3 = \{(i, L) : N_L - C - T_1 < i \leq L\}$, consists of state (i, L) where only either session request or departure can happen. The balance equation of these states can be written as

$$p_{k,j}k\mu = p_{k-1,j}\lambda, \quad (k, j) \in S_3. \quad (3)$$



(a) Level L state transitions

(b) Level j state transitions

(c) Level M state transitions

FIGURE 3. Case $T_2 - T_1 \leq 2C$ - State transition diagrams for levels L, j and M.

- Subset $S_4 = \{(N_j - C - T_1, j) : M < j \leq L\}$ includes state $(N_j - C - T_1, j)$ that can be reached from state $(N_j - C - T_1 - 1, j - 1)$ after the scale-out decision upon the arrival a new session request. Note that $(N_j - C - T_1, j)$ can be reached from either $(N_j - C - T_1 - 1, j)$ or $(N_j - C - T_1 + 1, j)$ as well. The balance equation of these states can be written as

$$p_{k,j}(\lambda + k\mu) = p_{k-1,j-1}\lambda + p_{k-1,j}\lambda + p_{k+1,j}(k+1)\mu, \quad (k,j) \in S_4. \tag{4}$$

- Subset $S_5 = \{(N_j - T_1 - 1, j) : M \leq j < L\}$ consists of state $(N_j - T_1 - 1, j)$ where the scale-out decision takes place upon the arrival of a session request. The balance equation of these states can be written as

$$p_{k,j}(\lambda + k\mu) = p_{k-1,j}\lambda, \quad (k,j) \in S_5. \tag{5}$$

- Subset $S_6 = \{(N_j + C - T_2, j) : M < j \leq L\}$ includes state $(N_j + C - T_2, j)$ that can be reached from state $(N_j + C - T_2 + 1, j + 1)$ when the scale-in decision is initiated by the departure of a session. The balance equation of these states can be written as

$$p_{k,j}(\lambda + k\mu) = p_{k-1,j}\lambda + p_{k+1,j}(k+1)\mu + p_{k+1,j+1}(k+1)\mu, \quad (k,j) \in S_6. \tag{6}$$

1) THE STEADY STATE PROBABILITY OF STATE $(*, L)$

We categorize the states when L UPF instances are launched into two subsets: $\{(k, L) : N_L - T_2 + 1 < k \leq N_L - C - T_1\}$ (i.e., the left-hand side of Figure 3a) and $\{(i, L) : N_L - C - T_1 < i \leq N_L\}$ (i.e., the right-hand side of Figure 3a).

(1) We proceed from the left-hand side to the right-hand side of subset $\{(k, L) : N_L - T_2 + 1 < k \leq N_L - C - T_1\}$.

From equation (1) for state $(N_L - T_2 + 1, L)$, we get

$$p_{N_L - T_2 + 2, L} = p_{N_L - T_2 + 1, L} \frac{(\lambda + \mu(N_L - T_2 + 1))}{(N_L - T_2 + 2)\mu}. \tag{7}$$

If $T_2 - C - T_1 \geq 3$, from equation (2) for state (k, L), $N_L - T_2 + 2 \leq k \leq N_L - C - T_1 - 1$, we obtain

$$p_{k+1,L} = p_{k,L} \frac{(\lambda + k\mu)}{(k+1)\mu} - p_{k-1,L} \frac{\lambda}{(k+1)\mu}. \tag{8}$$

From equations (7) and (8), $p_{N_L - T_2 + k, L}$ can be expressed in $p_{N_L - T_2 + 1, L}$ for $2 \leq k \leq T_2 - C - T_1$, $T_2 - C - T_1 \geq 3$. Note that if $T_2 - C - T_1 = 2$, $p_{N_L - T_2 + 2, L}$ is expressed in $p_{N_L - T_2 + 1, L}$ from (7).

(2) We proceed from right-hand side to left-hand side through balance equations of states in subset $\{(k, L) : N_L - C - T_1 + 1 \leq k \leq N_L\}$.

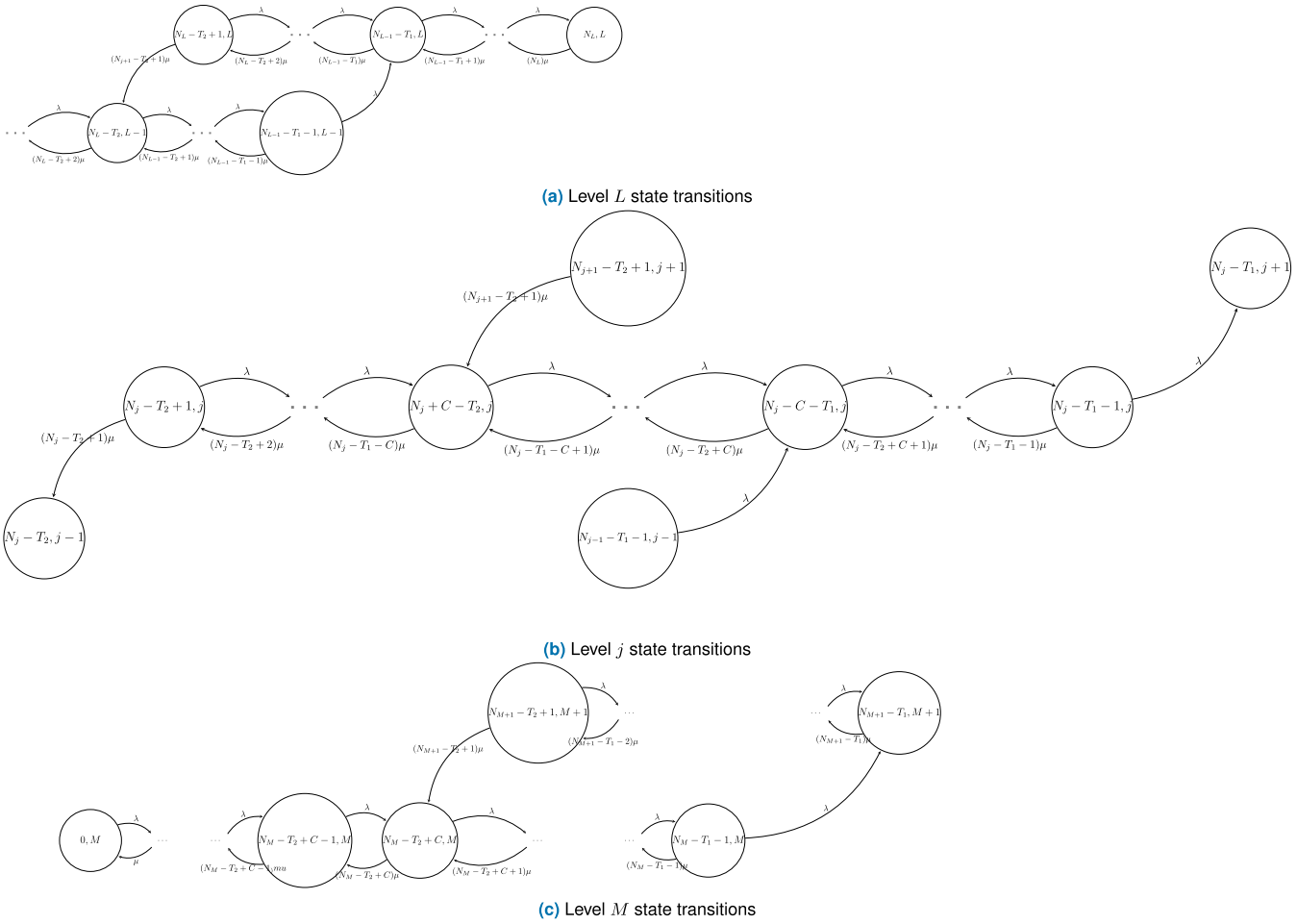


FIGURE 4. Case $T_2 - T_1 > 2C$ - State transition diagrams for levels L, j and M.

From equation (3) for state $\{(k, L) : N_L - C - T_1 + 1 \leq k \leq N_L\}$, we attain

$$p_{k-1,L} = p_{k,L} \frac{k\mu}{\lambda} \quad \forall\, N_L - C - T_1 + 1 \leq k \leq N_L. \tag{9}$$

As a consequence, we get

$$p_{N_L-C-T_1+k,L} = p_{N_L-C-T_1,L} \left(\frac{\lambda}{\mu}\right)^k \frac{1}{\prod_{u=1}^{k}(N_L - C - T_1 + u)}, \quad 1 \leq k \leq C + T_1, \tag{10}$$

which means that $p_{N_L-C-T_1+k,L}$ is expressed in $p_{N_L-C-T_1,L}$. Note that $p_{N_L-C-T_1,L}$ can be expressed in $p_{N_L-T_2+1,L}$. Therefore, $p_{i,L}$, $N_L - T_2 + 1 < i \leq N_L$, can be expressed in $p_{N_L-T_2+1,L}$.

2) THE STEADY STATE PROBABILITY OF STATE $(*, j)$, $j = L - 1, \ldots, M + 1$

We categorize the states when j UPF instances are launched into two subsets: $\{(k, j) : N_j - T_2 + 1 < k \leq N_j - C - T_1\}$ (i.e., the left-hand side of Figure 3b) and $\{(i, j) : N_j - C - T_1 < i \leq N_j - T_1 - 1\}$ (i.e., the right-hand side of Figure 3b).

(1) We proceed from the right-hand side to the left-hand side of subset $\{(i, j) : N_j - T_1 - 1 < i \leq N_j - C - T_1\}$. From equation (4) for state $(N_j - T_1, j + 1)$ we get

$$\begin{aligned} p_{N_j-T_1-1,j} = {} & p_{N_j-T_1,j+1}(\lambda + (N_j - T_1)\mu)/\lambda \\ & - p_{N_j-T_1-1,j+1} - p_{N_j-T_1+1,j+1} \\ & \times (N_j - T_1 + 1)\mu/\lambda. \end{aligned} \tag{11}$$

If $T_2 - T_1 = 2C$ then level $j + 2$ state probabilities should appear in the balance equation for state $(N_j - T_1, j + 1)$, $M < j < L - 1$. Equation (12) is used to calculate $p_{N_j-T_1-1,j}$ instead of equation (11) as

$$\begin{aligned} p_{N_j-T_1-1,j} = {} & p_{N_j-T_1,j+1}(\lambda + (N_j - T_1)\mu)/\lambda \\ & - p_{N_j-T_1-1,j+1} \\ & - p_{N_j-T_1+1,j+1}(N_j - T_1 + 1)\mu/\lambda \\ & - p_{N_j-T_1+1,j+2}(N_j - T_1 + 1)\mu/\lambda. \end{aligned} \tag{12}$$

From equation (5) for state $(N_j - T_1 - 1, j)$ we obtain

$$p_{N_j-T_1-2,j} = p_{N_j-T_1-1,j} \frac{\lambda + (N_j - T_1 - 1)\mu}{\lambda}. \tag{13}$$

If $T_2 - C - T_1 \geq 3$, from equation (2) for state (k, L), $N_j + C - T_2 + 1 \leq k \leq N_j - T_1 - 2$, we get

$$p_{k-1,j} = p_{k,j}\frac{(\lambda + k\mu)}{\lambda} - p_{k+1,j}\frac{(k+1)\mu}{\lambda}. \quad (14)$$

If $T_2 - T_1 \leq 2C - 1$, from equation (6) for state $(N_j + C - T_2, j)$ we obtain

$$\begin{aligned} p_{N_j+C-T_2-1,j} = {} & p_{N_j+C-T_2,j}\frac{(\lambda + (N_j + C - T_2)\mu)}{\lambda} \\ & - p_{N_j+C-T_2+1,j}\frac{(N_j + C - T_2 + 1)\mu}{\lambda} \\ & - p_{N_{j+1}-T_2+1,j+1}\frac{(N_{j+1} - T_2 + 1)\mu}{\lambda}. \end{aligned} \quad (15)$$

If $T_2 - T_1 \leq 2C - 2$, from equation (2) for state (k, j), $N_j - C - T_1 + 1 \leq k \leq N_j + C - T_2 - 1$ we obtain

$$p_{k-1,j} = p_{k,j}\frac{\lambda + k\mu}{\lambda} - p_{k+1,j}\frac{(k+1)\mu}{\lambda}. \quad (16)$$

From equations (11), (13), (14), (15) and (16) we conclude that, $p_{k,j}$ can be expressed in $p_{N_L-T_2+1,L}$ for $N_j - C - T_1 \leq k \leq N_j - T_1 - 1$.

(2) We proceed from the left-hand side to the right-hand side of subset $\{(i, j) : N_j - C - T_1 \leq i \leq N_j - T_2 + 1\}$. From equation (1) for state $(N_j - T_2 + 1, j)$, we get

$$p_{N_j-T_2+2,j} = p_{N_j-T_2+1,j}\frac{(\lambda + \mu(N_j - T_2 + 1))}{(N_j - T_2 + 2)\mu}. \quad (17)$$

If $T_2 - C - T_1 \geq 3$, from equation (2) for state (k, j) $N_j - T_2 + 2 \leq k \leq N_j - C - T_1 - 1$, we obtain

$$p_{k+1,j} = p_{k,j}\frac{(\lambda + k\mu)}{(k+1)\mu} - p_{k-1,j}\frac{\lambda}{(k+1)\mu}. \quad (18)$$

Equations (17) and (18) follow that $p_{k,j}$ can be expressed in $p_{N_j-T_2+1,j}$ for $N_j - C - T_1 \leq i \leq N_j - T_2 + 1$. Note that $p_{N_j-C-T_1,j}$ was calculated in equation (14) and expressed in $p_{N_L-T_2+1,L}$. As a result, $p_{N_j-T_2+k,j}$ is expressed in $p_{N_j-T_1-C,j}$, so it can be expressed in $p_{N_L-T_2+1,L}$.

3) THE STEADY STATE PROBABILITY OF STATE $(*, M)$

We divide the states when M UPF instances are launched into two subsets: $\{(k, M) : 0 \leq k \leq N_0 + C - T_2\}$ (i.e., the left-hand side of Figure 3c) and $\{(i, 0) : N_M + C - T_2 < i \leq N_M - T_1 - 1\}$ (i.e., the right-hand side of Figure 3c).

(1) We proceed from the right-hand side to the left-hand side of subset $\{(i, M) : N_M + C - T_2 + 1 \leq i \leq N_M - T_1 - 1\}$. Note that we can use equation (11) to calculate $p_{N_M-T_1-1,}$ from equation (4) for state $(N_M - T_1, 1)$. From equation (5) for state $(N_M - T_1 - 1, M)$, we get

$$p_{N_M-T_1-2,M} = p_{N_M-T_1-1,M}(\lambda + (N_M - T_1 - 1)\mu)/\lambda. \quad (19)$$

If $T_2 - C - T_1 \geq 3$, from equation (2) for state (k, M) $N_M + C - T_2 + 1 \leq k \leq N_M - T_1 - 2$, we get

$$p_{k-1,M} = p_{k,M}\frac{(\lambda + k\mu)}{\lambda} - p_{k+1,M}\frac{(k+1)\mu}{\lambda}. \quad (20)$$

(2) We proceed from the right-hand side to the left-hand side of subset $\{(i, M) : 0 \leq i \leq N_M + C - T_2 - 1\}$. From equation (6) for state $(N_M + C - T_2, M)$, we obtain

$$\begin{aligned} p_{N_M+C-T_2-1,M} = {} & p_{N_M+C-T_2,M}\frac{(\lambda + (N_M + C - T_2)\mu)}{\lambda} \\ & - p_{N_M+C-T_2+1,M}\frac{(N_M + C - T_2 + 1)\mu}{\lambda} \\ & - p_{N_{M+1}-T_2+1,M+1}\frac{(N_{M+1} - T_2 + 1)\mu}{\lambda}. \end{aligned} \quad (21)$$

From equation (2) for state (k, j), $1 \leq k \leq N_j + C - T_2 - 1$, we get

$$p_{k-1,M} = p_{k,M}\frac{\lambda + k\mu}{\lambda} - p_{k+1,M}\frac{(k+1)\mu}{\lambda}. \quad (22)$$

Based on equations (20), (21) and (22) we conclude that $p_{i,M}$ for $0 \leq i < N_M - T_1 - 1$ can be expressed in $p_{N_M-T_1-1,M}$. Note that $p_{N_M-T_1-1,M}$ can be expressed in $p_{N_L-T_2+1,L}$, therefore $p_{i,M}$ can be expressed in $p_{N_L-T_2+1,L}$.

To compute the stationary probabilities, we utilize the normalization equation $\sum_{(i,j)\in S} p_{i,j} = 1$ and Proposition 1.

B. PERFORMANCE MEASURES

The performance measures can be determined as follows:

- the blocking probability of sessions equals to the steady state probability of state (N_L, L)

$$P_{bS} = p_{N_L,L}, \quad (23)$$

- the average number of UPF instances is

$$V_d = \sum_{(i,j)\in S} j \times p_{i,j}, \quad (24)$$

- the average number of busy UPF instances in the system is

$$V_b = \sum_{(i,j)\in S} \lceil i/C \rceil \times p_{i,j}, \quad (25)$$

- the average number of idle instances in the system is

$$V_i = V_d - V_b, \quad (26)$$

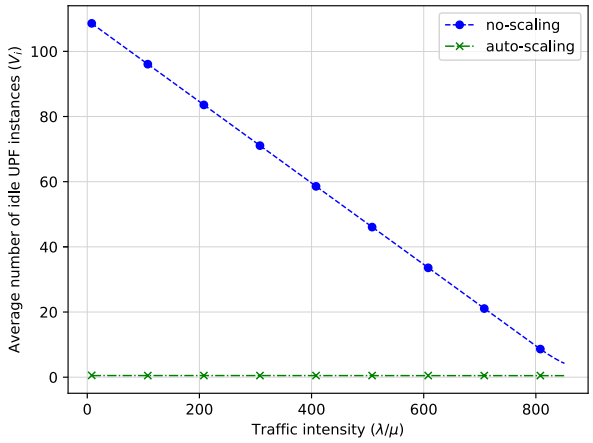
- the utilization is

$$U = \sum_{(i,j)\in S} \frac{i}{j \times C} \times p_{i,j}. \quad (27)$$

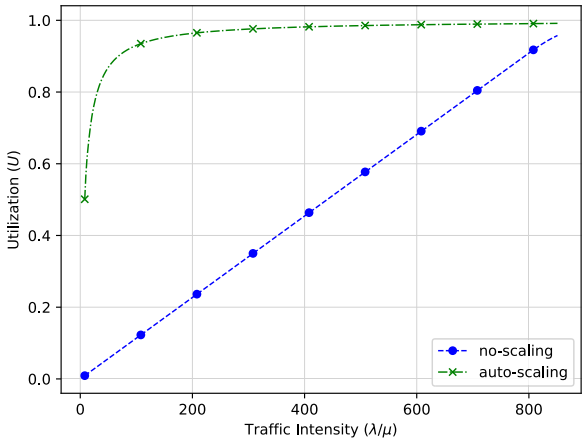
V. NUMERICAL RESULTS

For a numerical evaluation, we assume that

- UPF instances run in five physical servers [25]. Each server has the Intel Xeon 6238R 2,2 GHz processor with 28 cores and 4 × 64 GB RAM;
- each UPF session conveys video streaming data;
- six cores on each server are allocated for OS and the container management system;
- Each UPF instance occupies one core and 2GB RAM and serve maximum $C = 8$ simultaneous video streams.

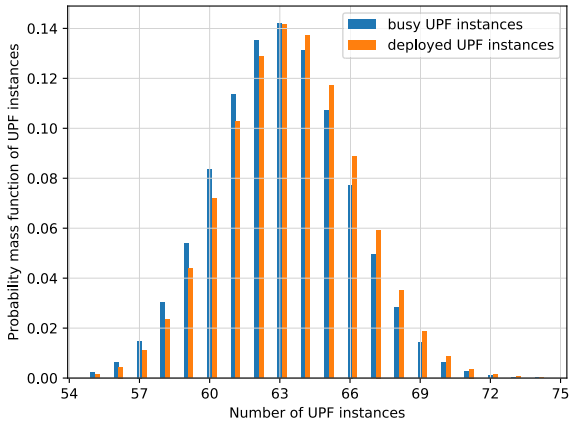


(a) Average number of idle UPF instances vs the traffic intensity

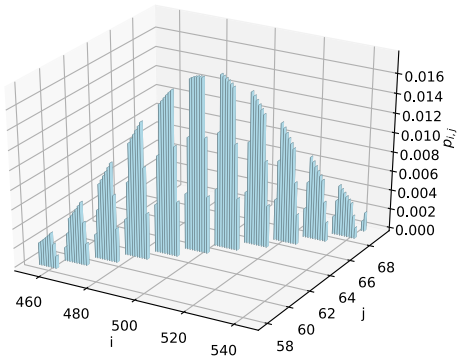


(b) Cluster utilization vs the traffic intensity

FIGURE 5. Average number of idle instances and cluster utilization.

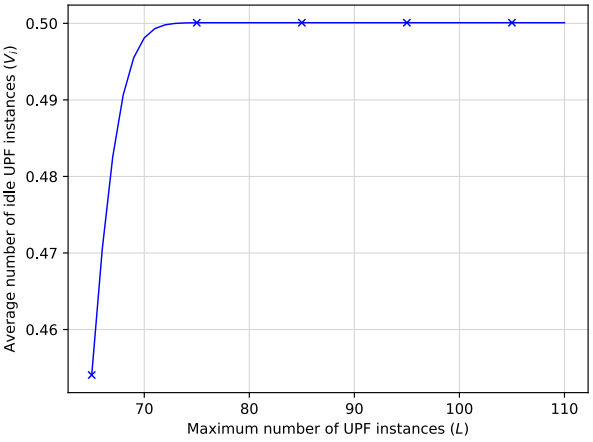


(a) The probability mass function (pmf) of UPF instances

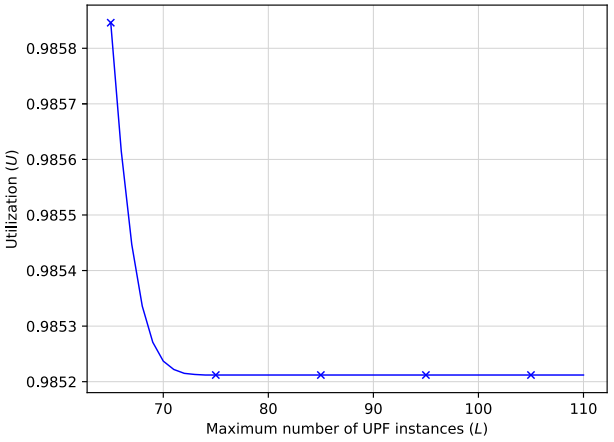


(b) The stationary probabilities of states

FIGURE 6. The probability distribution of UPF instances and PDU sessions, configuration of scaling algorithm with $\lambda/\mu = 500.0$, $L = 110$, $M = 1$, $C = 8$, $T_1 = 3$, $T_2 = 13$.



(a) The impact of L on the average number of idle UPF instances

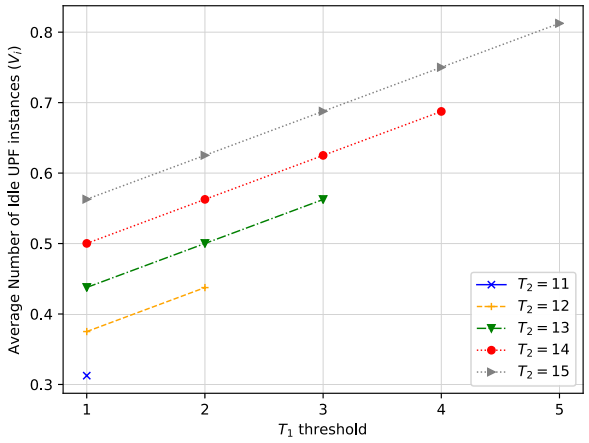


(b) The impact of L on the cluster utilization

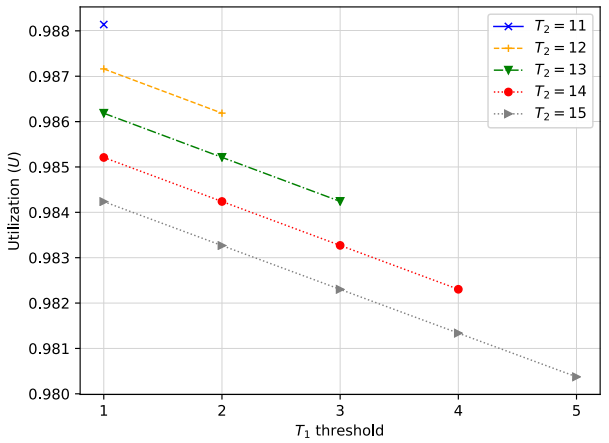
FIGURE 7. The performance measures versus L and λ/μ for $M = 1$, $C = 8$, $T_1 = 2$, $T_2 = 13$.

Therefore, the maximum number of simultaneous video streams is $110 \times 8 = 880$.

To check whether the threshold-based algorithm can manage the resource requirement in response to the change of traffic load, we compare two scenarios for the load (λ/μ) below 851 (due to the maximum capacity of the servers this is the maximum load value where the blocking probability of arriving sessions is below 1%).

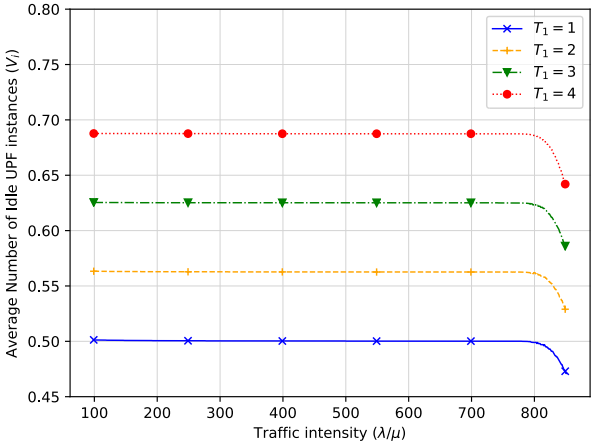


(a) The impact of T_1 and T_2 on the average number of idle UPF instances

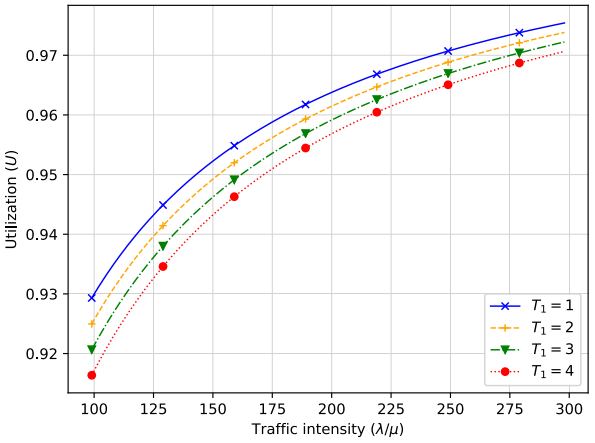


(b) The impact of T_1 and T_2 on the cluster utilization

FIGURE 8. The performance measures versus T_1 and T_2 for $\lambda/\mu = 500.0$, $L = 110$, $M = 1$, $C = 8$.



(a) The impact of T_1 on the average number of idle UPF instances



(b) The impact of T_1 on the cluster utilization

FIGURE 9. The performance measures versus T_1 and λ/μ for $L = 110$, $M = 1$, $C = 8$, $T_2 = 14$.

- *No-scaling* scenario assumes that all the UPF instances are launched. That is, 110 UPF instances are always available. Of course, this scenario wastes the resource when the traffic is low. Numerical results for this scenario are computed with the use of the Erlang B formula [26].
- *Auto-scaling* scenario is applied when the scaling algorithm is responsible for adjusting the number of UPF instances within the range of M to 110. Numerical results for this scenario are computed based on the analysis presented in this paper with $L = 110$.

We plot the average number of idle UPF instances vs traffic intensity (λ/μ), and the system utilization vs traffic intensity in Figure 5. It can be observed that the auto-scaling could save resource consumption and achieve high utilization. The average number of idle UPF instances is better on higher traffic intensities, which results in a 20% decrease in the average number of idle UPF instances. Similarly, the utilization is 50% for lower intensities increases to 99% for higher traffic rates.

The auto-scaling approach automatically adjusts the number of UPF instances without any intervention and the measurement of the traffic intensity. In Figure 6a the probability mass distribution (pmf) of the number of the deployed UPF instances and the busy UPF instances (which can be derived based on the stationary distribution of the states, see Fig. 6b) for $\lambda/\mu = 500.$, $L = 110$, $M = 1$, $C = 8$, $T_1 = 3$, $T_2 = 13$ are illustrated. We can observe that the pmf of the number of the deployed UPF instances is quite close to the pmf of the number of busy UPF instances. This mean the threshold-based scaling algorithm can keep the number of deployed instances quite close to the number of UPF instances necessary to serve PDU sessions.

We depict the average number of idle UPF instances vs L and the system utilization vs L for for $\lambda/\mu = 500.$, $M = 1$, $C = 8$, $T_1 = 3$, $T_2 = 13$ in Figure 7. It can be observed that the average number of idle UPF instances (7a) and the utilization (7b) will reach constant value while the maximum number of deployed UPF instances is higher than 75, which means that the scaling algorithm is keeping

the number deployed UPF instances to 75 regardless of the L in this case.

In Figure 8, we plot the average number of idle UPF instances and the utilization versus T_1 and T_2 for $\lambda/\mu = 500.0$, $L = 110$, $M = 1$, $C = 8$, $T_1 = 1$ and the autoscaling. As expected, the high utilization and the low number of average idle UPF instances come together. In the present setting the choice of $T_1 = 1, T_2 = 11$ could achieve the best performance, because the setting can give a chance to fill UPF instances, so the average number of idle UPF instances is low and high utilization is possible. Note that the same results can be obtained for other values of λ/μ as well (see Figure 9).

VI. CONCLUSION

We have proposed a queueing model for a threshold-based algorithm that controls the number of User Plane Function instances in 5G systems. We have provided an efficient procedure to compute the steady-state probabilities and performance measures. Numerical results showed that the threshold-based scheduling algorithm could automatically adjust the number of UPF instances in response to the change of traffic load, save the resource consumption and keep the high utilization of the requested resource. In the paper, the assumption of the Poisson arrival process is a common practice to make the queueing model mathematically tractable. As a future work, we will consider the application of other stochastic processes like [27]–[29] to model non-Poisson arrivals as well.

At present, various algorithms could be considered for the autoscaling of 5G UPF instances. The advantage of the variants of threshold-based algorithms is the simple implementation and operation. The threshold-based algorithm presented in this paper and the efficient evaluation could be used as a reference in benchmarking where the comparison of algorithms can be performed to select an appropriate solution and to check whether a certain configuration has enough capacity to provide the Quality of Service for PDU sessions. Furthermore, artificial intelligence-based solutions [30] could be applied and results obtained by the efficient computation method presented in this paper can be used in the training phase as well.

APPENDIX

PROOF FOR CASE $T_2 - T_1 > 2C$

In what follows we provide a proof for Proposition 1 for case $T_2 - T_1 > 2C$.

A. THE STEADY STATE PROBABILITY OF STATE $(*, L)$

The states when there are L dynamic instances are categorized into two subsets: $\{(k, L) : N_L - T_2 + 1 < k \leq N_L - C - T_1\}$ (i.e., the left-hand side of Figure 4a) and $\{(i, L) : N_L - C - T_1 < i \leq N_L\}$ (i.e., the right-hand side of Figure 4a).

The state space and the state transitions on level L are exactly the same as in the case described in IV-A1 so we use exactly the same steps used in IV-A1 to prove that level L state probabilities can be expressed with $p_{N_L-T_2+1,L}$.

B. THE STEADY STATE PROBABILITY OF STATE $(*, j)$, $j = L-1, \ldots, M+1$

The states when there are j dynamic instances are categorized into two subsets: $\{(k, j) : N_j - T_2 + 1 < k \leq N_j - C - T_1\}$ (i.e., the left-hand side of Figure 4b) and $\{(i, j) : N_j - C - T_1 < i \leq N_j - T_1 - 1\}$ (i.e., the right-hand side of Figure 4b). Note, this case is similar to the case presented in IV-A2, the main difference is that the position of state $(N_j + C - T_2, j)$ and the position of state $(N_j - C - T_1, j)$ are switched.

(1) We proceed from the right-hand side to the left-hand side of subset $\{(i, j) : N_j - C - T_1 < i \leq N_j - T_1 - 1\}$. From equation (4) for state $(N_j - T_1, j + 1)$, we attain

$$\begin{aligned} p_{N_j-T_1-1,j} = {} & p_{N_j-T_1,j+1}(\lambda + (N_j - T_1)\mu)/\lambda \\ & - p_{N_j-T_1-1,j+1} - p_{N_j-T_1+1,j+1} \\ & \times (N_j - T_1 + 1)\mu/\lambda. \end{aligned} \tag{28}$$

From equation (5) for state $(N_j - T_1 - 1, j)$, we get

$$p_{N_j-T_1-2,j} = p_{N_j-T_1-1,j}\frac{\lambda + (N_j - T_1 - 1)\mu}{\lambda}. \tag{29}$$

If $C \geq 2$, from equation (2) for state (k, L) $N_j - C - T_1 + 1 \leq k \leq N_j - T_1 - 2$, we obtain

$$p_{k-1,j} = p_{k,j}\frac{(\lambda + k\mu)}{\lambda} - p_{k+1,j}\frac{(k+1)\mu}{\lambda}. \tag{30}$$

The consequence of equations (28),(29) and (30) is that $p_{k,j}$ can be expressed in $p_{N_L-T_2+1,L}$ for $N_j - C - T_1 \leq k \leq N_j - T_1 - 1$.

(2) We proceed from the left-hand side to the right-hand side of subset $\{(i, j) : N_j - T_2 + 1 \leq i \leq N_j - C - T_1\}$. From equation (1) for state $(N_j - T_2 + 1, j)$, we get

$$p_{N_j-T_2+2,j} = p_{N_j-T_2+1,j}\frac{(\lambda + \mu(N_j - T_2 + 1))}{(N_j - T_2 + 2)\mu}. \tag{31}$$

If $C \geq 2$, from equation (2) for state (k, j) $N_j - T_2 + 2 \leq k \leq N_j + C - T_2 - 1$, we attain

$$p_{k+1,j} = p_{k,j}\frac{(\lambda + k\mu)}{(k+1)\mu} - p_{k-1,j}\frac{\lambda}{(k+1)\mu}. \tag{32}$$

If $T_2 - T_1 \geq 2C + 1$, from equation (6) for state $(N_j + C - T_2, j)$, we obtain

$$\begin{aligned} p_{N_j-T_2+C+1,j} = {} & p_{N_j-T_2+C,j}\frac{\lambda + \mu(N_j - T_2 + C)}{\mu(N_j - T_2 + C + 1)} \\ & - p_{N_j-T_2+C-1,j}\frac{\lambda}{\mu(N_j - T_2 + C + 1)} \\ & - p_{N_j+C-T_2+1,j+1}. \end{aligned} \tag{33}$$

If $T_2 - T_1 \geq 2C + 2$, from equation (2) for state (k, j) $N_j + C - T_2 + 1 \leq k \leq N_j - C - T_1 - 1$, we obtain

$$p_{k+1,j} = p_{k,j}\frac{(\lambda + k\mu)}{(k+1)\mu} - p_{k-1,j}\frac{\lambda}{(k+1)\mu}. \tag{34}$$

From equations (31),(32),(33) and (34), $p_{k,j}$ can be expressed in $p_{N_j-T_2+1,j}$ for $N_j - T2 + 1 \leq k \leq N_j - C - T_1$. Note that $p_{N_j-C-T_1,j}$ can be expressed in $p_{N_L-T_2+1,L}$, so $p_{k,j}$ can be expressed in $p_{N_L-T_2+1,L}$ for $N_j - T_2 + 1 \leq k \leq N_j - T_1 - 1$.

C. THE STEADY STATE PROBABILITY OF STATE $(*, M)$

The states when there are M UPF instances are categorized into two subsets: $\{(k, M) : 0 \leq k \leq N_0 + C - T_2\}$ (i.e., the left-hand side of Figure 4c) and $\{(i, M) : N_M + C - T_2 < i \leq N_M - T_1 - 1\}$ (i.e., the right-hand side of Figure 4c).

The state space and the state transitions on level M are exactly the same as in the case described in IV-A3 so we use exactly the same steps used in IV-A3 to prove that level M state probabilities can be expressed in $p_{N_L-T_2+1,L}$.

REFERENCES

[1] S. Redana and O. Bulakci, Eds., "View on 5G architecture," 5GPPP Archit. Work. Group, White Paper 22.891, Version 14.2.0, Jul. 2018.
[2] *5G; Study on Scenarios and Requirements for Next Generation Access Technologies*, document TS 38.913, Version 16.0.0, Release 16, 3GPP, Jul. 2020.
[3] *Technical Specification Group Services and System Aspects; System Architecture for the 5G System (5GS); Stage 2*, document TS 23.501, Version 16.7.0, 3GPP, Dec. 2020.
[4] D. Chandramouli, R. Liebhart, and J. Pirskanen, *5G for the Connected World*. Hoboken, NJ, USA: Wiley, 2019.
[5] A. Osseiran, J. F. Monserrat, and P. Marsch, *5G Mobile and Wireless Communications Technology*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[6] B. Familiar, *Microservices, IoT, and Azure: Leveraging DevOps and Microservice Architecture to Deliver SaaS Solutions*. New York, NY, USA: Apress, Oct. 2015.
[7] V. Farcic, *The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline With Containerized Microservices*, 1st ed. Scotts Valley, CA, USA: CreateSpace Independent Publishing Platform, Feb. 2016.
[8] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A review of auto-scaling techniques for elastic applications in cloud environments," *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, Dec. 2014.
[9] G. Járó, A. Hilt, L. Nagy, M. A. Tündik, and J. Varga, "Evolution towards telco-cloud: Reflections on dimensioning, availability and operability: (Invited paper)," in *Proc. 42nd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2019, pp. 1–8.
[10] H. Tang, D. Zhou, and D. Chen, "Dynamic network function instance scaling based on traffic forecasting and VNF placement in operator data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 3, pp. 530–543, Mar. 2019.
[11] D. Kumar, S. Chakrabarti, A. S. Rajan, and J. Huang, "Scaling telecom core network functions in public cloud infrastructure," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2020, pp. 9–16.
[12] J. Herrera and G. Moltó, "Toward bio-inspired auto-scaling algorithms: An elasticity approach for container orchestration platforms," *IEEE Access*, vol. 8, pp. 52139–52150, 2020.
[13] Y. Guo, A. L. Stolyar, and A. Walid, "Online VM auto-scaling algorithms for application hosting in a cloud," *IEEE Trans. Cloud Comput.*, vol. 8, no. 3, pp. 889–898, Sep. 2020.
[14] A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, "Providing performance guarantees for cloud-deployed applications," *IEEE Trans. Cloud Comput.*, vol. 8, no. 1, pp. 269–281, Jan. 2020.
[15] S. Taherizadeh and V. Stankovski, "Dynamic multi-level auto-scaling rules for containerized applications," *Comput. J.*, vol. 62, no. 2, pp. 174–197, Feb. 2019.
[16] C. Wu, V. Sreekanti, and J. M. Hellerstein, "Autoscaling tiered cloud storage in anna," *VLDB J.*, vol. 30, no. 1, pp. 25–43, Sep. 2020.
[17] L. Baresi, S. Guinea, A. Leva, and G. Quattrocchi, "A discrete-time feedback controller for containerized cloud applications," in *Proc. 24th ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, Nov. 2016, pp. 217–228.
[18] F. Zhang, X. Tang, X. Li, S. U. Khan, and Z. Li, "Quantifying cloud elasticity with container-based autoscaling," *Future Gener. Comput. Syst.*, vol. 98, pp. 672–681, Sep. 2019.
[19] E. Casalicchio, "A study on performance measures for auto-scaling CPU-intensive containerized applications," *Cluster Comput.*, vol. 22, no. 3, pp. 995–1006, Jan. 2019.
[20] I. Gervásio, K. Castro, and A. P. F. Araújo, "A hybrid automatic elasticity solution for the IaaS layer based on dynamic thresholds and time series," in *Proc. 15th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2020, pp. 1–6.
[21] Q. Z. Ullah, G. M. Khan, and S. Hassan, "Cloud infrastructure estimation and auto-scaling using recurrent Cartesian genetic programming-based ANN," *IEEE Access*, vol. 8, pp. 17965–17985, 2020.
[22] *5G; Technical Specifications and Technical Reports for a 5G Based 3GPP System*, document TS 23.205, Version 16.0.0, 3GPP, Aug. 2020.
[23] *Study on Integrated Access and Backhaul*, document TS 38.874, Version 16.0.0, 3GPP, Dec. 2018.
[24] W. J. Stewart, *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton, NJ, USA: Princeton Univ. Press, 2009.
[25] Nokia. *AirFrame Open Edge Server*. Accessed: Mar. 26 2020. [Online]. Available: https://www.nokia.com/networks/products/airframe-open-edge-server/
[26] L. Kleinrock, *Theory: Queueing Systems*, vol. 1. Hoboken, NJ, USA: Wiley, 1975.
[27] M. R. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1981.
[28] L. P. Seelen, "An algorithm for Ph/Ph/c queues," *Eur. J. Oper. Res.*, vol. 23, no. 1, pp. 118–127, Jan. 1986.
[29] R. Chakka and T. V. Do, "The MM $\sum_{k=1}^{K} CPP_k$/GE/c/L G-queue with heterogeneous servers: Steady state solution and an application to performance evaluation," *Perform. Eval.*, vol. 64, no. 3, pp. 191–209, 2007.
[30] H. T. Nguyen, T. V. Do, A. Hegyi, and C. Rotter, "An approach to apply reinforcement learning for a VNF scaling problem," in *Proc. 22nd Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, A. Galis, F. Guillemin, R. Noldus, S. Secci, F. Idzikowski, and M. Sayit, Eds., Feb. 2019, pp. 94–99.

CSABA ROTTER received the M.Sc. degree in applied electronics from the Technical University of Oradea, in 1995, and the M.Sc. degree in IT management from Central European University, Budapest, in 2008. He joined Nokia, in 1999. In 2008, he joined Nokia Research Center (currently Nokia Bell Labs). He is currently heading the Multi Cloud Orchestration Research Department, Nokia Bell Labs. He is involved in cloud-related research topics targeting cloud-native network service operation challenges in a highly distributed multivendor environment. His particular interest includes developing solutions for the performance-related service level agreement of concurrent applications sharing the resources in distributed environments. His passion for automation started years before when he was responsible for test automation concept development in large telecommunication systems.

TIEN VAN DO received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Technical University of Budapest, Hungary, in 1991 and 1996, respectively, the Habilitation from BME, and the D.Sc. title from the Hungarian Academy of Sciences, in 2011. He is currently a Professor with the Department of Networked Systems and Services, Budapest University of Technology and Economics. He led various projects on network planning and software implementations that results are directly used for industry, such as ATM & IP network planning software for Hungarian Telekom, GGSN tester for Nokia, and performance testing program for the performance testing of the NOKIA IMS product and automatic software testing framework for Nokia Siemens Networks. His research interests include queuing theory, telecommunication networks, cloud computing, performance evaluation and planning of ICT systems, and machine learning.