Engineering Journal issn: 0125-8281

Article

Modeling of a Threshold-Based Scaling and Fractional Admission Controlling Problem for NSSI Blocks to Ensure QoS and Optimize Resource Utilization in 5G Network Slicing

Ly Cuong Hoa[1,2,a] **and Thanh Chuong Dang**[1,b,*]

1 University of Sciences, Hue University, Hue City, Vietnam
2 College of Information and Communication Technology, Can Tho University, Can Tho City, Vietnam
E-mail: [a]cuonghl@hueuni.edu.vn, [b,*]dtchuong@hueuni.edu.vn (Corresponding author).

Abstract. In the 5G core network, network functions can be flexibly scaled out/in network slices proactively. The autoscaling process increases effectiveness by scaling out network function instances and minimizes expenses by scaling NSSI (Network Slice Subnet Instance) blocks. 5G network functions have to deploy or terminate multiple NFIs (Network Function Instances) simultaneously; it significantly affects the system's cost efficiency and ensures QoS (Quality of Service). In the paper, we will propose a Markov chain-based analytical model for the Threshold-based Scaling and Fractional Admission Controlling problem of NSSI blocks (called TSFAC-NB) within a 5G network slice. The model will incorporate two thresholds related to the scaling-out/scaling-in of NSSI blocks. We will also propose a FAC (Fractional Admission Controlling) mechanism in the model with two thresholds added to control NSSI blocks by the probability to optimize resource utilization. A threshold-based scaling and fractional admission controlling (TSFAC) algorithm is developed and implemented in Kubernetes-based Open5GS to evaluate the performance of TSFAC-NB experimentally. The simulation results show a similarity between the analytical and experimental results, in which the analytical model helps to determine the admission thresholds for the best performance of TSFAC-NB.

Keywords: 5G network slicing, NSSI block, TS, FAC, Q-TSFAC-NB.

ENGINEERING JOURNAL Volume 29 Issue 1
Received 11 September 2024
Accepted 10 January 2025
Published 31 January 2025
Online at https://engj.org/
DOI:10.4186/ej.2025.29.1.57

DOI:10.4186/ej.2025.29.1.57

1. Introduction

Commencing with 3GPP (3rd Generation Partnership Project) Release 15 (R15) [1-5] 5G broadband is designed to serve not only humans but everything, including various types of machines. Moreover, network functions in 5G core networks will be initiated not only by network operators. When users instantiate a separate mobile network, control plane network functions can still be provided by the operators. Users only own UPF (User Plane Function) on their request [1]. NFV (Network Function Virtualization) is a network architecture model designed to virtualize network services that typically run on dedicated, separate network devices. With NFV, functions such as routing, load balancing, and firewall access control are packed as virtual machines or packed processes distributed on common hardware. Individual VNFs (Virtual Network Function) are foundational elements of the NFV architecture.

According to definitions provided by 3GPP TS 28.530 [5], a network slice comprises several NSIs (Network Slice Instance) characterized by dynamic scalability to process the capabilities of the essential network functions (Fig. 1) [1].

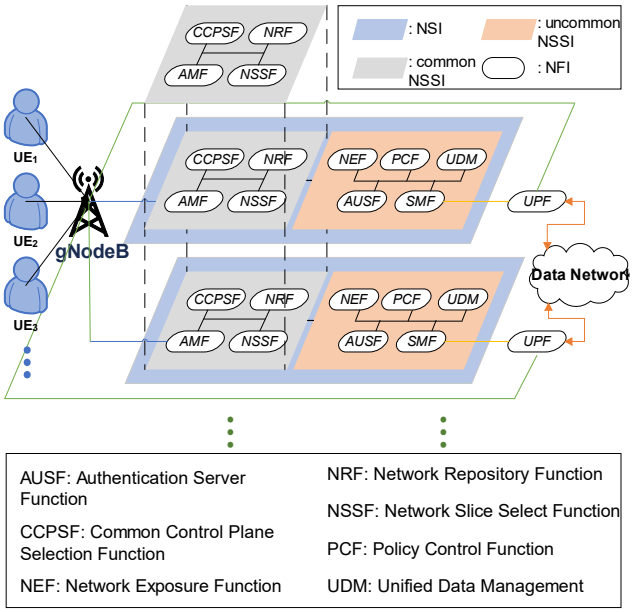


Fig. 1. Network slicing model based on 3GPP R15 standards [1, 5].

In Fig. 1, a slice network consists of multiple NSIs, such as NSI_1 and NSI_2, where a NSI can include multiple NSSIs (Network Slice Subnet Instance) or a NSI consists only of a single NSSI. For example, NSI_1 includes $NSSI_1$ and common NSSI. A NSSI includes multiple NFIs (Network Function Instance), particularly VNFs. For example, AUSF, NEF, PCF, SMF, UDM, and UPF are VNFs within $NSSI_1$. In addition, NSIs can share the functions of the control plane on a common NSSI. Figure 1 depicts the common NSI shared between NSI_1 and NSI_2. The common NSSI covers control plane functions. Bottleneck phenomena that occur on unimplemented user planes are often incorporated and leveraged by NSSI, such as $NSSI_1$ and $NSSI_2$. During the initiation of a connection, a user sends an initial communication to authenticate to the control plane. After successfully initiating the session, data can be conveyed to UPFs for data processing and forwarding, bypassing the control plane. However, when a user unexpectedly submits enormous data, an operator can automatically set more UPFs to minimize system traffic overload. NFIs in 5G core networks can be scaled flexibly to allocate resources within network slices. Improving performance processes scales out NSSIs, and operating costs are reduced by scaling in NSSIs.

As previously mentioned, a network slice in 5G comprises three different layers: NFI, NSSI, and NSI [1]. To establish slices, we must analyze a NSSI, which includes some NFIs and can be integrated with other NSSIs to form a new NSI. This means that network scaling established by the 3GPP specifications requires taking into account the components and combinations of each layer. Moreover, as stipulated in 3GPP Release 15, a high degree of interdependence exists between 5G network functions.

In the article, we have modeled the problem of NSSI blocks with multiple VNFs forming a single NSI block of an NSI and deploying multiple NSSIs in a network slice according to dynamic incoming UE traffic based on scaling out or scaling in thresholds for the 5G core network using the Markov queueing model [1, 6]. The model also incorporates a threshold control mechanism proposed in this paper to manage resources more cost-effectively while ensuring system utilization.

The main contributions of the article include:

- Modeling of Threshold-based Scaling for NSSI Blocks (TS-NB);
- Proposing a FAC mechanism to efficiently respond to UE requests;
- Integrating a FAC mechanism in the model TS-NB to scale NSSI Blocks effectively. This innovative model is referred to as TSFAC-NB;
- Building a Queueing model for TSFAC-NB (Q-TSFAC-NB) to evaluate the impact of control thresholds on the performance of TSFAC-NB; and
- Developing a TSFAC algorithm and implementing simulation on Kubernetes with Open5GS to evaluate the performance of TSFAC-NB experimentally.

The remainder of the paper is organized as follows. Section 2 introduces the related research work. The TSFAC-NB model and queueing-based performance analysis are presented in Section 3. Experimental implementation and result analysis are described in Section 4. Finally, Section 5 presents the conclusion and future.

2. Related Works

Queueing models have emerged as a valuable tool for analyzing resource allocation in network slicing, offering insights into system performance, and enabling optimization strategies. Several studies have employed

queueing models to address network slicing challenges [7, 8]. Adou et al. [7] investigate a retrial queue to model network slicing in 5G wireless networks with an unlimited buffer in order to ensure QoS (Quality of Service). The generating function approach has been applied in solving this model. Kochetkova et al. [8] propose three metrics for evaluating the effectiveness of dynamic network slicing. These metrics consider the trade-off between resource allocation efficiency, signaling overhead, and QoS.

The issue of analyzing system performance for automatic load-balancing mechanisms has also been proposed by researchers using the DBCA [9], DASA [10], and ASA [11]. However, these studies focus on providing a single instance, which is inadequate for analyzing 5G network slicing. In 5G, there can be multiple NFIs within an NSSI, and multiple NSSIs are implemented to create a network slice. While bulk schemes can consider the responses of multiple servers/jobs, they still lack the flexibility and efficiency to explore how to deploy numerous small instances in more sophisticated strategies.

Autoscaling strategies in NSSIs offer the potential to enhance operational flexibility, optimize resource utilization, and guarantee performance requirements. However, frequent scaling can not only degrade system performance but also lead to significant cost increases. Therefore, the authors in [1] consider the impact of pre-provisioning NFIs to prevent sudden traffic spikes, as well as designing block instantiation, i.e., modeling multiple VNFs as a block and considering deploying multiple blocks using a threshold-based scaling strategy. With a threshold-based approach, the authors in [1] utilize single/double threshold mechanisms to design block instantiation mechanisms, optimizing instantiation time and the number of NSSIs by controlling the threshold value and determining the shutdown time and number of NSSIs based on operator needs. However, threshold-based block instantiation with value m, as described in [1], may encounter latency issues concerning queued UEs, which we will solve in our proposed model in the next section.

In 5G, an UPF is utilized to transmit data. Specifically, an UPF maintains routing tables and establishes GTP (General Packet Radio Service Tunneling Protocol) tunnels to forward packets. Before transmission, an UPF decodes packet headers and searches for routing protocols in its dataset. In 3GPP R15, a SMF (Session Management Function) can control and manage multiple UPFs. A MAPE (Monitor-Analyze-Plan-Execute) method was also introduced by Nguyen et al. [12] to manage the autoscaling of UPFs. Establishing a vast number of UPFs at the lowest expenditure while fulfilling user demands remains challenging for network operators. Accordingly, Rotter and Do [6] introduced a queuing model for the UPF in the 5G core network based on a threshold-based scaling algorithm to manage UPFs efficiently. This model is highly promising in its potential application to NSSI blocks. System performance enhancement can be achieved by Guard Channel (GC) and Fractional Guard Channel (FGC) mechanisms. Cruz-Perez and Ortigoza-Guerrero [13] overviewed call admission control mechanisms to ensure QoS in mobile networks. Controlling the admission of UE requests is therefore necessary to improve the efficiency of resource use and maintain stable system performance.

In the article [14], schemes for network slicing based on mathematical models with SDN and NFV under an operational and management architecture were introduced. Debbabi et al. [15] summarized and categorized algorithms related to NS; however, the issue of NFI scaling in NS was not addressed. Banchs et al. [16] conducted a survey on network slicing with two approaches: one based on reservation and the other on resource sharing for network slices. Nguyen et al. [17] applied a horizontal scaling algorithm for VNFs using a reinforcement learning approach to enhance operational and management efficiency in NFV. To reduce the transmission power of base stations in mobile wireless networks, Tun and Kunavut [18] proposed a continuous cell zooming algorithm with Time-Adaptive Periodic Update and Location-Adaptive Periodic Update to ensure QoS. Saha et al. [19] proposed a distribution and scheduling algorithm for energy, frequency, and time in the architecture that separates the control plane and the user plane to address power demand and energy efficiency, particularly for smart cells in the control and user plane separation architecture. In [20], Saha et al. focused on addressing two issues related to transitioning the RAN architecture from centralized to distributed and subsequently to virtualized RAN. They then utilized a proof-of-concept (PoC) evaluation to understand and assess the technologies.

In this study, we will model TS-NB by applying T_1 and T_2 thresholds to the dynamic scaling problem of NSSI blocks [1,6]. Additionally, the model will integrate a fractional admission controlling mechanism (TSFAC-NB) to control the admission of incoming UE requests, thereby more effectively managing the deployment/termination of NSSI blocks. A queueing model for analyzing TSFAC-NB is also developed, and implementing TSFAC-NB on Kubernetes with Open5GS is also deployed. The following section will describe our contributions in detail.

3. Analytical Model Q-TSFAC-NB with FAC Mechanism

3.1. Modeling of TS-NB

The model of TS-NB within a NSI during a network slice is modeled as a Markov queueing model. Assumed that the K virtual NFIs within a NSI are divided into a maximum of L NSSI blocks, where each block contains k virtual NFIs; however, there may be a case where NSSI blocks from 1 to $L-1$ each have k virtual NFIs and the last block (L^{th} block) having only r virtual NFIs ($r \leq k$), so $L = \lceil K/k \rceil$. The system initializes the minimum number of M blocks $(M < L)$, including $M * k$ virtual

NFIs, which are always ready to serve users. That is, M NSSI blocks are always on. The remaining blocks ($(L-M)$ blocks) that are set up dynamically (scaling-out or scaling-in depending) on the traffic of the UE requests [1, 6].

The challenge in the analytical model is to achieve resource utilization efficiency by analyzing the issues of reservation, setup, and dynamic block termination. Unlike conventional cloud computing, network slices are implemented with a hierarchical structure where numerous NFIs are established to create a NSSI and numerous NSSIs are implemented to form a NSI [1]. To achieve this goal, we consider k NFIs as a block to delineate a mutual reliance between NFIs and NSSIs and institute separate blocks to describe the relationship between NSSIs and NSIs. Additionally, frequent scaling will significantly reduce system performance or setup cost, so the model also reserves the minimum number of M NSSI blocks and appropriately controls the scaling-out and scaling-out threshold values T_1 and T_2, respectively, to ensure precise resource allocation.

- Block size (k) and last block size (r): Whenever instances are needed, we set up an NSSI block with k NFIs and r NFIs for the last block. An instance corresponds to a NFI and a block consists of NFIs equivalent to a NSSI. By establishing several blocks, we can flexibly configure NSIs with different numbers of NSSIs. In the following sections, we will denote N_j as the maximum number of virtual NFIs when j NSSI blocks have been deployed $(M \le j \le L)$, subject to $N_j = \begin{cases} j \times k, (M \le j < L) \\ (j-1) \times k + r, (j = L) \end{cases}$. For computational simplicity, we have combined this into a general formula, $N_j = \left\lceil \frac{j \times (L-1)}{L} \right\rceil \times k + \left\lfloor \frac{j}{L} \right\rfloor \times r, (M \le j \le L)$.
- Scaling-out threshold T_1: In this system model, we leverage the threshold T_1 to identify the most cost-effective way to deploy a block, adhering to operator cost-saving strategies, known as the scaling-out decision threshold [6]. Utilization of the threshold T_1 here overcomes the problem of using the threshold m as in the model in [1]. According to [1], when an UE request arrives but all n_0 NFIs have been allocated, the UE must be queued, which can increase UE latency. Instead, our model applies the threshold T_1 to allow the reservation of blocks based on incoming traffic, ensuring that when UEs arrive, resources are always available for allocation.
- Scaling-in threshold T_2: The operator-specified block size (k or r) is also used as a certain threshold, known as the scaling-in decision threshold, denoted by T_2 [6]. Specifically, to avoid excessive switching, set-up instances must be maintained for some time even when they are not deployed. The block size k can also be applied as a terminating threshold, i.e., when $T_2 = k$ or $T_2 = r$ for the last block, the system will terminate a block of NSSIs if k or r for the last block is idle [1].

Let $I(t)$ denote the number of UE requests being served by NFIs and $J(t)$ $(M \le J(t) \le L)$ be the number of blocks NSSIs containing virtual NFIs that are established at time t. The maximum number of UE requests that can be served at time t is $N_j = \left\lceil \frac{J(t) \times (L-1)}{L} \right\rceil \times k + \left\lfloor \frac{J(t)}{L} \right\rfloor \times r$. In this analytical model, the deployment and termination of blocks are performed dynamically based on incoming UE traffic and the thresholds T_1 and T_2. Specifically, the model will deploy a block when the number of UEs being served reaches $N_j - T_1 - 1$ according to the threshold T_1. Additionally, the model will also terminate a block when whole k instances in a block are idle, corresponding to the case when the number of UEs being served reaches the value $N_j - T_2 - 1$ according to the threshold T_2. Controlling the threshold values for scaling out and scaling in T_1 and T_2, respectively, can be done by applying scaling algorithms to manage NSSIs in a reasonable way. The threshold-based scaling problem can be illustrated in Fig. 2 and Fig. 3.

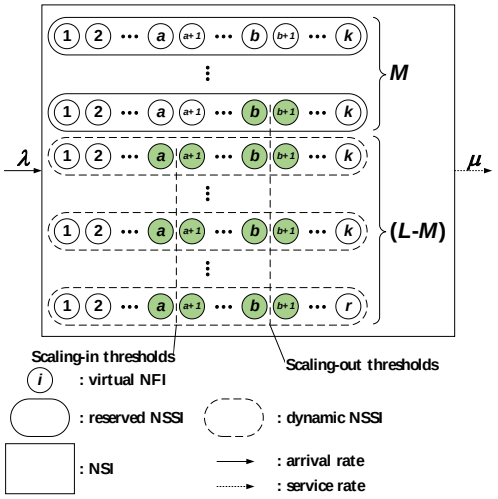


Fig. 2. Modeling operation of the Q-TS-NB in an NSI slice network.

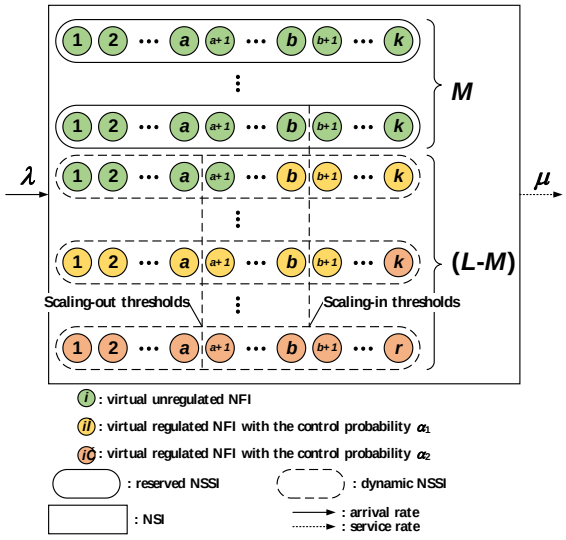


Fig. 3. Modeling operation of the Q-TSFAC-NB in an NSI slice network.

The following cases are considered [6]:

- If $I(t) = N_{J(t)} - T_1 - 1$ and $J(t) < L$ based on incoming UE requests for an NFI, a new NSSI block will be deployed and $J(t) = J(t) + 1$.
- If the number of idle NFIs within each NSSI block is equal to T_2 and $J(t) \geq M + 1$ when an UE leaves, a request will also be processed to terminate an idle block. Note that if the scaling-in request is performed, then $J(t) = J(t) - 1$. Note that NFI migration between blocks can be performed to increase the chance of having a block with all k or r idle instances.

The analytical model is a continuous-time Markov chain (CTMC) $\{(I(t), J(t)), t \geq 0\}$ for resource management operations for a single NSI slice, where the rate of incoming UE requests is assumed to follow a Poisson distribution λ, and the service time is presumed to comply with an exponential distribution $1/\mu$ (μ is the service rate). The state space $S = \{(i,j) | i \in I(t), j \in J(t)\}$ is defined, and the corresponding state transitions in the model can be depicted as follows:

- Transition from (i,j) to $(i+1,j)$: When a new UE request arrives and the system can allocate a NFI to serve it, an NSSI block is not deployed if:
  - either $0 \leq i \leq N_M - T_1 - 2, j = M$;
  - or $N_j - T_2 < i \leq N_j - k - T_1 - 2$, $M < j < L$;
  - or $N_L - T_2 < i \leq N_L - 1, j = L$.
- Transition from (i,j) to $(i+1, j+1)$: When a new UE request arrives and the system can allocate a NFI to serve it, and a new NSSI block is deployed based on scaling out. At this time, $i = N_j - T_1 - 1, M \leq j < L$.
- Transition from (i,j) to $(i-1, j)$: When an UE completes and departs, and the system returns a NFI instance, and no scaling is performed if:
  - either $0 \leq i \leq N_M - T_1, j = M$;
  - or $N_j - T_2 + 2 < i \leq N_j - T_1 - 2, M < j < L$;
  - or $N_L - T_2 + 2 < i \leq N_L, j = L$.
- Transition from (i,j) to $(i-1, j-1)$: When an UE completes and departs, the system also terminates a NSSI block based on the scaling-in. At this time, $i \leq N_j - T_2 + 1, M < j \leq L$. In the case of choosing $T_2 = k$, we can rewrite $i \leq N_j - k + 1$ or $i \leq k \times (j-1) + 1$, with $M < j \leq L$.

The state space of the CTMC process $\{(I(t), J(t)), t \geq 0\}$ is denoted by [6]:

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

The number of states is:

$$|S| = N_M - T_1 + (T_2 - T_1 - 1)(L - M - 1) + T_2 + k - r = M \times k + T_2 - T_1 + (T_2 - T_1 - 1)(L - M - 1) + k - r \quad (1)$$

The states when there are j NSSI blocks deployed are called level j. If with j, $M < j < L$, the deployed NSSI blocks, due to operating rules, will have two special states occur:

- The states $(N_j + k - T_2, j)$ can be reached from the states $(N_j + k - T_2 + 1, j+1)$ $\left((N_j + k - T_2 + 1, j+1) \rightarrow (N_j + k - T_2, j)\right)$ due to the departure of an UE and a *scaling-in* behavior executed to terminate a NSSI block.
- The states $(N_j - k - T_1, j)$ are the consequences of a *scaling-out* procedure from the previous states $(N_j - k - T_1 - 1, j-1)$ $\left((N_j - k - T_1 - 1, j-1) \rightarrow (N_j - k - T_1, j)\right)$) when an UE request arrives and a NSSI block is deployed.

Here, two distinct scenarios can be differentiated based on the mathematical relationship between $N_j + k - T_2$ and $N_j - k - T_1$. In the first case, $N_j + k - T_2 \geq N_j - k - T_1$ (i.e., $T_2 - T_1 \leq 2k$). The second case is characterized by the satisfaction of the inequality $T_2 - T_1 > 2k$. Within the scope of the paper, we only consider the case $N_j + k - T_2 \geq N_j - k - T_1$, or $T_2 - T_1 \leq 2k$. The other case has been proven as in [6].

3.2. Modeling of TSFAC-NB

3.2.1. System modeling

The utilization of the two thresholds T_1 and T_2 enables the flexible and efficient deployment and termination of NSSI blocks based on incoming UE traffic. However, this threshold pair is considered locally for each NSSI to scale out or scale in when the number of UE requests reaches them. This approach clearly does not consider the entire system's remaining NSSI blocks when scaling. In the case of a sudden increase in the number of UE requests, while the number of NSSI blocks is almost exhausted, it may be impossible to satisfy UE requests; congestion will occur, and as a result, the service efficiency of the system is seriously degraded. Controlling the admission of UE requests is therefore necessary to improve the efficiency of resource use and maintain stable system performance. To solve this limitation, we will propose an improved model of TS-NB, in which a fractional admission controlling mechanism is integrated to control the admission of incoming EU requests, thereby more effectively managing the deployment and termination of NSSI blocks (called TSFAC-NB). A queueing model for analyzing TSFAC-NB is also developed (Q-TSFAC-NB), and implementing simulation TSFAC-NB on Kubernetes with Open5GS is also deployed.

In the model TSFAC-NB, we introduce two probabilities α_1 and α_2 $(0 < \alpha_2 \leq \alpha_1 \leq 1)$ for each state $\beta_{i,j}$ of each session according to the thresholds H_1 and H_2 $(1 \leq H_1 \leq H_2)$ to minimize the number of idle

NSSIs. On the other hand, the fractional admission controlling mechanism ensures a sufficient number of activated NSSI blocks for redundancy in case of sudden traffic spikes and guarantees that system performance does not degrade (Fig. 3). The proposed threshold values H_1 and H_2 are completely independent of the scaling-out threshold T_1 and the scaling-in threshold T_2. This implies a gradual throttling of requests until the number of requests in the system attains the threshold H_1 and then a strict limit when the threshold H_2 is reached (Table 1).

Table 1. Notations used in Q-TS-NB and Q-TSFAC-NB models.

Symbols	Descriptions
k	Number of virtual NFIs within a NSSI block (from block 1 to block $(L-1)$)
r	Number of virtual NFIs in the last NSSI block (block L^{th}) ($r \leq k$)
K	Total number of virtual NFIs within a NSI slice network
M	Number of reserved (initial) NSSI blocks
L	Maximum number of NSSI blocks including reserved and dynamic virtual NFIs ($L = \lceil K/k \rceil$)
T_1	The scaling-out decision threshold
T_2	The scaling-in decision threshold
λ	Average arrival rate of UEs
μ	Average service rate of UEs
N_j	Maximum number of idle virtual NFIs with j deployed NSSI blocks ($M \leq j \leq L$)
$p_{i,j}$	The steady state probability of state (i,j)
H_1	The strict threshold of FAC
H_2	The more stringent threshold of FAC
α_1	The probability of triggering H_1
α_2	The probability of triggering H_2
$\beta_{i,j}$	The control probability of state (i,j)
$I(t)$	Number of UE requests served by NFIs at time t
$J(t)$	Number of NSSI blocks containing virtual NFIs deployed at a given time t

3.2.2. FAC mechanism

The FGC mechanism proposed in [13] enables the limitation of new incoming requests using a control parameter when system resources have reached the threshold limit. Based on the FGC mechanism, combined with the addition of probability $\beta_{i,j}$ in each state for each request according to the aforementioned thresholds H_1 and H_2, the FAC mechanism we propose in this paper's model is defined as follows.

Definition FAC. *The FAC mechanism is a control scheme where resources are managed based on the number of requests being served in the system to segment by establishing two thresholds* H_1 *and* H_2 *corresponding to two probabilities* α_1 *and* α_2. *These parameters are used to determine the probability* $\beta_{i,j}$ *of system state* (i,j) *in a two-dimensional Markov chain* $\{(I(t), J(t)), t \geq 0\}$.

The FAC mechanism is designed to regulate incoming requests through control parameters H_1 and H_2 when system resources have reached a certain threshold. Based on the number of requests being served, the FAC mechanism sets the appropriate thresholds H_1 and H_2 with the probabilities α_1 and α_2, respectively. The FAC mechanism can be considered a generalized case of FGC when not limited by control parameters. However, parameters may be limited to a specific range in certain systems. In Definition FAC, the parameter values are not constrained and can have a certain range depending on the deployed system.

3.2.3. TSFAC algorithm

Based on Definition FAC, we further refine TS-NB by introducing two thresholds and their corresponding probabilities to limit deployed resources and ensure QoS for the system, as presented in Algorithm TSFAC. The purpose of admission controlling is to prevent uncontrolled deployment, system underutilization, and waste of resource.

Algorithm TSFAC: Threshold-based Scaling and Fractional Admission Controlling

Input: $M, L, k, r\ \lambda, \mu, T_1, T_2, H_1, H_2, \alpha_1, \alpha_2$
Output: $I(t), J(t), \lambda_{i,j}$ ($\lambda_{i,j}$ are arrival rates at states (i,j))

$I(t) \leftarrow 0$
$J(t) \leftarrow M$
While $I(t) < (L-1) \times k + r$ **do**
  $N_{J(t)} \leftarrow \left\lfloor \frac{J(t) \times (L-1)}{L} \right\rfloor \times k + \left\lfloor \frac{J(t)}{L} \right\rfloor \times r$
  $\beta_{i,j} \leftarrow 1$
  If $H_1 < I(t) \leq H_2$ **then**
    $\beta_{i,j} \leftarrow \alpha_1$
  End if
  If $H_2 < I(t)$ **then**
    $\beta_{i,j} \leftarrow \alpha_2$
  End if
  $\lambda_{i,j} \leftarrow \lambda \times \beta_{i,j}$
  If a new UE requests a NFI **then**
    $I(t) \leftarrow I(t) + 1$
    If $I(t) = N_{J(t)} - T_1 - 1$ and $J(t) < L$ **then**
      $J(t) \leftarrow J(t) + 1$; // *scaling-out*
    End if
  End if
  If a NFI in the system departs **then**
    $I(t) \leftarrow I(t) - 1$
    If $I(t) = N_{J(t)} - T_2 + 1$ and $J(t) \geq M + 1$ then
      $J(t) \rightarrow J(t) - 1$; // *scaling-in*
    End if
  End if
End While
Return $I(t), J(t), \lambda_{i,j}$.

Algorithm TSFAC has a complexity of $O(k \times L)$. The algorithm considers the probability control value for the states (i, j) as $\beta_{i,j}$ with initial default values of 1. The algorithm then compares the states (i, j), and if $H_1 < i \leq H_2$ then $\beta_{i,j}$ is α_1, and if $i > H_2$ then $\beta_{i,j}$ is α_2.

Algorithm TSFAC manages NSSI blocks dynamically based on predefined thresholds, which helps maintain better performance without over-provisioning or under-utilizing NSSI blocks and wasting of resource.

3.2.4. Operation of Q-TSFAC-NB

TSFAC-NB is implemented to adjust the number of deployed NSSI blocks corresponding to the condition of applying two thresholds, H_1 and H_2. The operation of TSFAC is depicted in Fig. 3, where scaling and fractional admission controlling are performed by the algorithm TSFAC-NB.

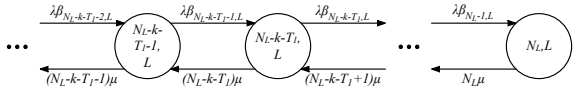
Accordingly, the system has at most L deployed NSSI blocks and each deployed NSSI block has at most k NFI instances serving k UE requests. Thus, there is a maximum of $k \times L$ UEs being served in the system. b and a are the controlled positions for H_1 and H_2, respectively. As shown in Fig. 3, the *uncontrolled cells* with *green color* with the probabilities $\beta_{i,j} = 1$. The *yellow cells controlled* with the probabilities $\beta_{i,j} = \alpha_1$. The *red cells controlled* with the probabilities $\beta_{i,j} = \alpha_2$. The queueing model of TSFAC-NB (Q-TSFAC-NB) is developed from the queueing model of TS-NB. Specifically, when the number of UE requests in the system reaches $N_j - T_1 - 1$, labeled as a_j in Fig. 3, the system deployed a new NSSI block. Similarly, if the number of UE requests decreases to $N_j - T_2 + 1$, labeled as $b_j + 1$ in Fig. 3, the system will terminate one NSSI block. According to **Definition FAC**, the lines 5-12 of **Algorithm TSFAC** can be described more clearly as follows:

- In the case $H_1 < H_2$,
  - If $H_2 \leq k \times L$ and $i \leq H_1$ then $\beta_{i,j} = 1$;
  - If $H_2 \leq k \times L$ and $H_1 < i \leq H_2$ then $\beta_{i,j} = \alpha_1$;
  - If $H_2 < k \times L$ and $H_2 < i$ then $\beta_{i,j} = \alpha_2$;
- In the case $H_1 = H_2$,
  - If $i \leq H_1$ then $\beta_{i,j} = 1$;
  - If $i > H_1$ then $\beta_{i,j} = \alpha_2$.

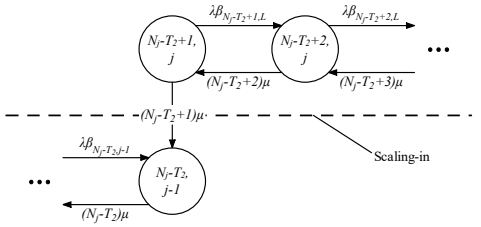
For exceptional cases, if $H_1 = H_2 = k \times L$ or $\alpha_1 = \alpha_2 = 1$ then $\beta_{i,j} = 1$; Q-TSFAC-NB becomes to Q-TS-NB.

3.2.5. State diagram and system state equilibrium equations

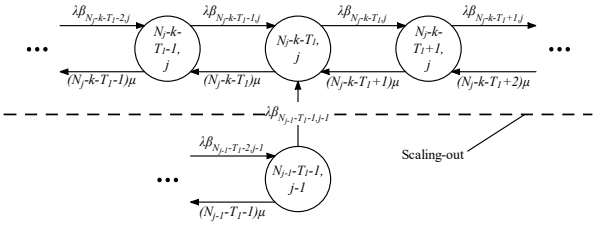
From Q-TSFAC-NB in Fig. 3, state equilibrium equations and state transition schemes are as the following equations, where equilibrium probabilities of the two-dimensional Continuous Time Markov Chain $(I(t), J(t)), t \geq 0$ are denoted as. The state diagram of the Q-TSFAC-NB model is described in Fig. 4 and consists of 6 state transition groups.



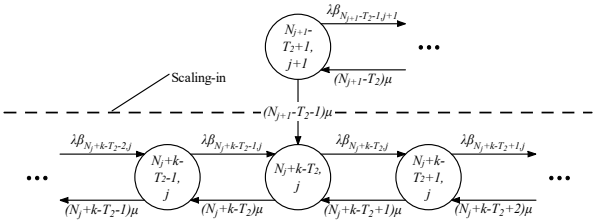
(a) State transition subdiagram for the Eq. (2).



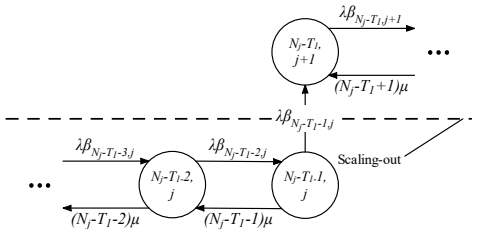
(b) State transition subdiagram for the Eq. (3).



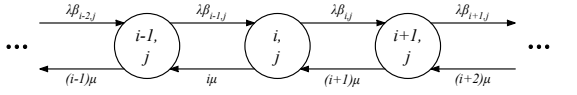
(c) State transition subdiagram for the Eq. (4).



(d) State transition subdiagram for the Eq. (5).



(e) State transition subdiagram for the Eq. (6).



(f) State transition subdiagram for the Eq. (7).

Fig. 4. State transition diagrams of Q-TSFAC-NB.

From Q-TSFAC-NB in Fig. 3 and state transition schemes in Fig. 4, state equilibrium equations from (1) to (6) corresponding with figures from Fig. 4a to Fig. 4f are follows, where equilibrium probabilities of the two-

dimensional Continuous Time Markov Chain $(I(t), J(t)), t \geq 0$ are denoted as

$$p_{i,j} = \lim_{t \to +\infty} P(I(t) = i, J(t) = j), (i,j) \in S$$

where S is the set of states of the system. The cardinality of the set S is determined as $|S|$ in (1)

We have state transition equations:

$$p_{i,L} i\mu = p_{i-1,L} \lambda \beta_{i-1,L}, (N_L - k - T_1 < i \leq N_L) \quad (2)$$

$$p_{N_j - T_2 + 1, j} \left[\lambda \beta_{N_j - T_2 + 1, j} + (N_j - T_2 + 1)\mu\right] = p_{N_j - T_2 + 2, j}(N_j - T_2 + 2)\mu, (M < j \leq L) \quad (3)$$

$$p_{N_j - k - T_1, j} \left[\lambda \beta_{N_j - k - T_1 j} + (N_j - k - T_1)\mu\right] = p_{N_j - k - T_1 - 1, j-1} \lambda \beta_{N_j - k - T_1 - 1, j-1} + p_{N_j - k - T_1 - 1, j} \lambda \beta_{N_j - k - T_1 - 1, j} + p_{N_j - k - T_1 + 1, j}(N_j - k - T_1 + 1)\mu, (M \leq j < L) \quad (4)$$

$$p_{N_j + k - T_2, j} \left[\lambda \beta_{N_j + k - T_2, j} + (N_j + k - T_2)\mu\right] = p_{N_j + k - T_2, j} \lambda \beta_{N_j + k - T_2, j} + p_{N_j + k - T_2 + 1, j}(N_j + k - T_2 + 1)\mu + p_{N_j + k - T_2 + 1, j+1}(N_j + k - T_2 + 1)\mu, (M < j \leq L) \quad (5)$$

$$p_{N_j - T_1 - 1, j} \left[\lambda \beta_{N_j - T_1 - 1, j} + (N_j - T_1 - 1)\mu\right] = p_{N_j - T_1 - 2, j} \lambda \beta_{N_j - T_1 - 2, j}, (M \leq j < L) \quad (6)$$

$$p_{i,j}(\lambda \beta_{i,j} + i\mu) = p_{i-1,j} \lambda \beta_{i-1,j} + p_{i+1,j}(i+1)\mu, \left(i \notin \begin{Bmatrix} (N_j - T_2 + 1) \cup (N_j - k - T_1) \cup \\ \cup (N_j + k - T_2) \cup (N_j - T_1 - 1) \end{Bmatrix} \middle| M \leq j < L \right) \quad (7)$$

The equation system of (2) to (7) can be solved using a system of linear regression equations with a normalization condition by setting $p'_{i,j} = p_{i,j} / p_{N_L,L}$. In this case, $p'_{N_L,L} = 1$. At this point, the system of equations (2) to (7) can be solved by backtracking [21] since there is a defined value of $p'_{N_L,L} = 1$. After determining the $p'_{i,j}$ based on the normalization condition $\sum_{(i,j) \in S} p_{i,j} = 1$, we can calculate:

$$p_{N_L,L} = \frac{1}{\sum_{(i,j) \in S} p'_{i,j}} \quad (8)$$

We then derive the probabilities $p_{i,j}$ from $p_{i,j} = p'_{i,j} \times p_{N_L,L}$.

3.3. Performance Evaluation Metrics

The performance evaluation metrics of our proposed model are as follows. Q-TSFAC-NB model adds the probabilities $\beta_{i,j}$ to achieve the minimal number of idle NSSI blocks.

- The average number of deployed NSSI blocks includes the used NSSI blocks and idle NSSI blocks.

$$V_d = \sum_{(i,j) \in S} j p_{i,j} \quad (9)$$

- The average number of busy NSSI blocks includes the deployed and used NSSI blocks.

$$V_b = \sum_{(i,j) \in S} \left\lceil \frac{i}{k} \right\rceil p_{i,j} \quad (10)$$

- The average number of NSSI blocks deployed but idle (V_{id}) is for insurance purposes. However, sometimes, the number of idle NSSI blocks exceeds the required insurance, which is called redundant NSSI blocks. The redundancy leads to a waste of resources and increased management costs. Therefore, Q-TSFAC-NB aims to minimize the number of redundant NSSI blocks.

$$V_{id} = V_d - V_b \quad (11)$$

- Utilization is the ratio of used resources to allocated resources.

$$U = \sum_{(i,j) \in S} \frac{i}{jk} p_{i,j} \quad (12)$$

The probabilities $\beta_{i,j}$ in Q-TSFAC-NB directly affect the equilibrium probabilities $p_{i,j}$, which improves the performance parameters of Q-TSFAC-NB compared to Q-TS-NB. We concern ourselves with equations from (9) to (12), as they are important metrics for determining system optimality. The next section will clarify the advantages of our models.

4. Results and Discussions

We conduct performance evaluations with respect to changes in thresholds. The system's performance will be assessed through the following metrics: the average number of idle NSSIs V_{id} and utilization U. In some cases, the parameters are reset to suit the simulation objectives.

In the following sections, we perform evaluations based on the thresholds T_1 and T_1, the thresholds H_1 and H_2, and the probabilities α_1 and α_2. Additionally, we conduct simulations to assess the accuracy of the model. The parameters we will analyze/simulate in the system include: $k = 8, M = 1, L = 60$. The parameters such as the thresholds T_1, T_2, H_1, and H_2, the probabilities α_1 and α_2, traffic λ/μ, and the number of NFIs in the last NSSI r will be adjusted to fit our comparison scenarios.

The performance evaluation metrics are as mentioned in Section 3.3:

- The average number of idle NSSI blocks (noted V_{id}): results are computed using formula (11) according to the theory and are averaged from 100 to 300 simulations, and
- The utilization (noted U): results are computed using formula (12) according to the theory and are averaged from 100 to 300 simulations.

We first examine the variation in the number of NFIs in the last NSSI block with parameters $T_1 = 1$ and $T_2 = 11$ as a function of traffic λ/μ. The results shown in Fig. 5 reveal that as the value of r increases from 1 to 8, the average number of idle NSSIs tends to remain stable with traffic λ/μ ranging from 100 to 400, and only exhibits significant changes when λ/μ exceeds 400. Specifically, when $\lambda/\mu > 400$, the average number of idle NSSIs is inversely related to the value of r. This occurs because as r decreases, the value of $(k - r)$ increases, where $(k - r)$ represents the number of vacant NFI positions in the last NSSI block, leading to an increase in the average number of idle NSSIs. Conversely, the position required to perform the termination of an NSSI block is $2k -$

$T_2 + 1 = 6$. Thus, when $r \leq 2$, the average number of idle NSSIs tends to increase, whereas when $r > 2$, it tends to decrease. Utilization also generally improves with increasing values of r (see Fig. 6).

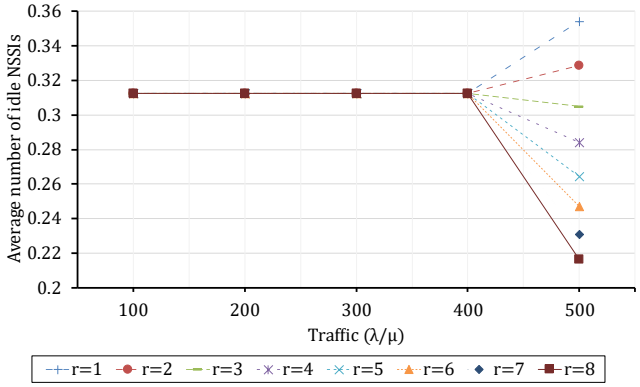


Fig. 5. Analysis results of TS-NB based on the NFI number in the last NSSI for the average number of idle NSSIs.

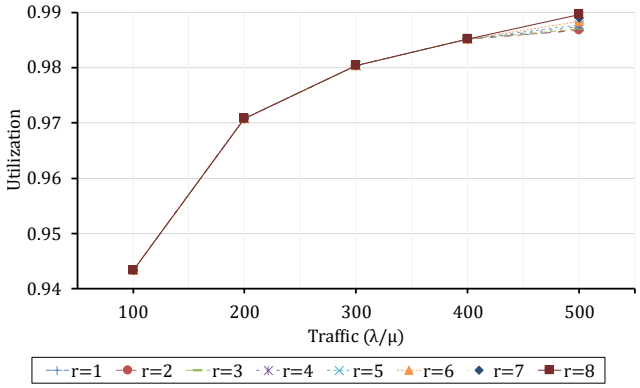


Fig. 6. Analysis results of TS-NB based on the NFI number in the last NSSI for utilization.

In order to assess the impact of scaling thresholds according to Fig. 7 and Fig. 8, we present the average number of idle NSSIs and utilization as a function of the parameters T_1 and T_2, with values $\lambda/\mu = 500$, $L = 60$, $M = 1$, $C = 8$, and $r = 8$ when applying the TS-NB model. As observed, a low average number of idle NSSIs is generally associated with high utilization. The configuration $T_1 = 1$ and $T_2 = 11$ shows the best performance, as this setting allows NFIs to fill the NSSI blocks more effectively. Consequently, the average number of idle NSSIs decreases and utilization improves.

As discussed in the theoretical model above, incorporating the FAC mechanism into the model helps guarantee that the system maintains optimal performance while neither over-allocating nor using resources inefficiently, resulting in resource waste. As a result, we will analyze the model with the FAC mechanism to show that it is more efficient than the model without it.

Next, in this section, we will compare the above TS-NB model with a model that integrates the FAC mechanism, i.e., the TSFAC-NB model. To evaluate the effectiveness of the TSFAC-NB model when compared with the TS-NB model, we assess both the V_{id} and U metrics. The following analysis results will show that the V_{id} value in the TSFAC-NB model will be smaller than V_{id} in TS-NB, while U value in TSFAC-NB has a larger value than U in TS-NB. This means that the TSFAC-NB model has deployed enough of the required number of idle NSSI blocks when integrating the FAC mechanism, thereby decreasing redundant deployed NSSI blocks and helping to reduce system costs.

Moreover, when applying the FAC mechanism to the TS-NB method, resulting in TSFAC-NB, and comparing it with the TS-NB model without the FAC mechanism using similar parameters as shown in Fig. 7 and Fig. 8, we observe significant improvements. Specifically, with settings $H_1 = \frac{1}{3}|S|$, $H_2 = \frac{2}{3}|S|$, $\alpha_1 = 0.9$, and $\alpha_2 = 0.8$, results in Fig. 9 and Fig. 10 demonstrate that TSFAC-NB achieves a lower average number of idle NSSIs and higher utilization compared to the TS-NB model. This indicates that the FAC mechanism provides a clear advantage in reducing the average number of idle NSSIs and enhancing utilization.

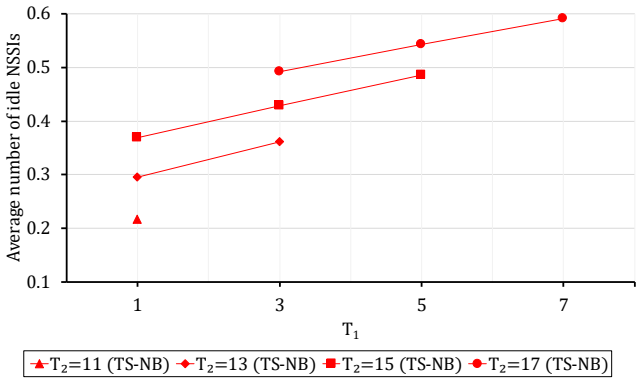


Fig. 7. Analysis results of TS-NB based on the thresholds T_1 and T_2 for the average number of idle NSSIs

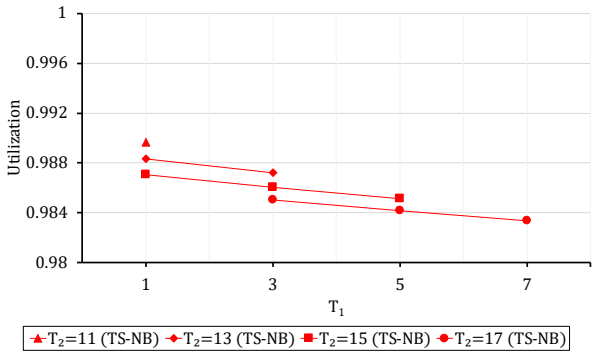


Fig. 8. Analysis results of TS-NB based on the thresholds T_1 and T_2 for utilization.

DOI:10.4186/ej.2025.29.1.57

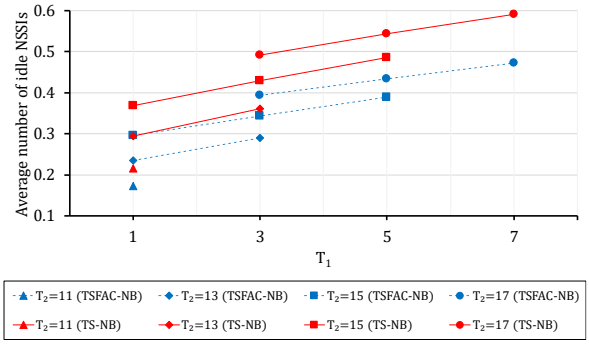


Fig. 9. Analysis results of the model the TS-NB and TSFAC-NB based on thresholds T_1 and T_2 for the average number of idle NSSIs.

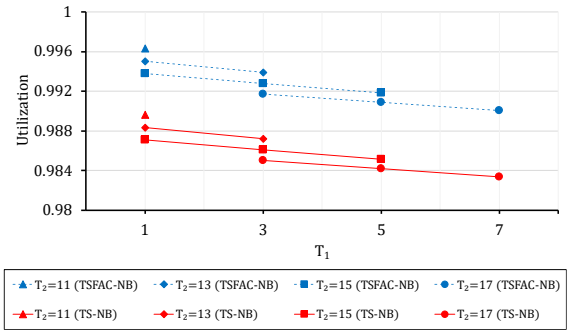


Fig. 10. Analysis results of the model the TS-NB and TSFAC-NB based on thresholds T_1 and T_2 for utilization.

We present the average number of idle NSSIs (Fig. 11) and utilization (Fig. 12) with parameters $\lambda/\mu = 500$, $k = r = 8$, $T_1 = 1$, $T_2 = 11$, $\alpha_1 = 0.9$, and $\alpha_2 = 0.8$ while varying the thresholds H_1 and H_2. Although thresholds-based scaling according to the TS-NB model has contributed to reducing the average number of idle NSSIs, the addition of the FAC mechanism provides even greater benefits. Specifically, from the early stages, the FAC mechanism can further conserve resources (by reducing the average number of idle NSSIs) and enhance utilization. The average number of idle NSSIs performs better when the threshold $H_2 < |S|$, indicating that applying the FAC mechanism not only improves resource allocation but also maintains stable utilization (see Fig. 12).

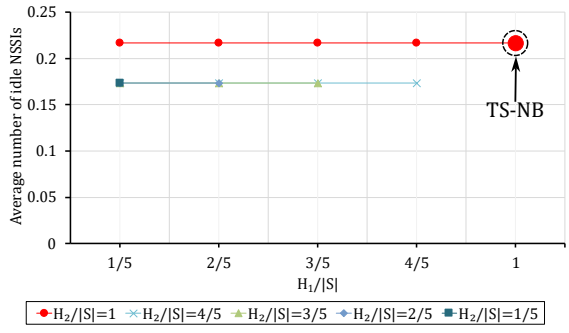


Fig. 11. Analysis results of the model applying the model TSFAC-NB based on the thresholds H_1 and H_2 for the average number of idle NSSIs. In this case, the model TS-NB is the special case where $H_1/|S| = H_2/|S| = 100\%$, represented by the large red marker.

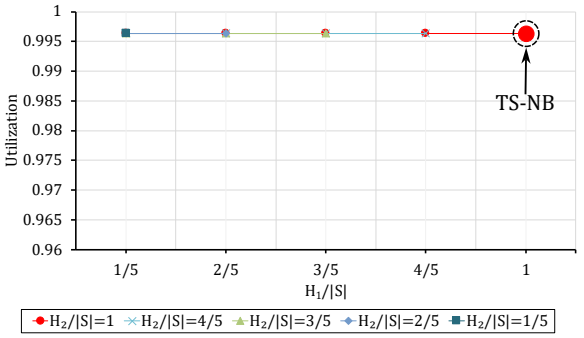


Fig. 12. Analysis results of the model applying the model TSFAC-NB based on the thresholds H_1 and H_2 for utilization. In this case, the model TS-NB is the special case where $H_1/|S| = H_2/|S| = 100\%$, represented by the large red marker.

We conduct an analysis with the parameters $T_1 = 1$, $T_2 = 11$, $H_1 = \frac{1}{3}|S|$ and $H_2 = \frac{2}{3}|S|$, while varying the values of α_1 and α_2. The case where $\alpha_1 = \alpha_2 = 1$ represents a special scenario of the TSFAC-NB model (that is TS-NB model), as discussed in Section 3.2.4. The results presented in Fig. 13 indicate that as α_2 decreases, the TSFAC-NB model achieves a lower average number of idle NSSIs. Notably, the TSFAC-NB model generally performs better than or at least comparably to the TS-NB model. Furthermore, utilization remains stable (see Fig. 14). This suggests that a more stringent control can enhance resource consumption efficiency while maintaining high and stable system utilization.

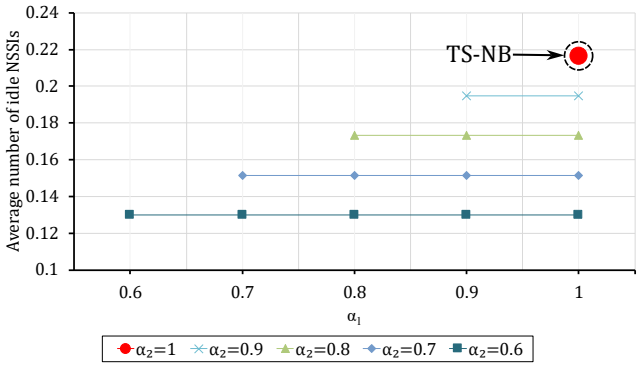


Fig. 13. Analysis results of the model TSFAC-NB applying the FAC mechanism based on probabilities α_1 and α_2 for the average number of idle NSSIs. In this context, the model TS-NB is the special case with $\alpha_1 = \alpha_2 = 1$, represented by the large red marker.

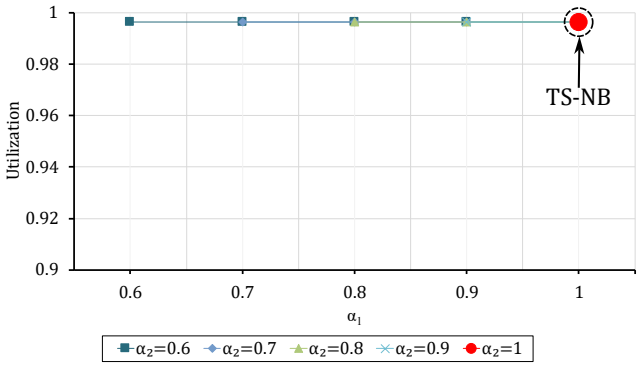


Fig. 14. Analysis results of the model TSFAC-NB applying the FAC mechanism based on probabilities α_1 and α_2 for utilization. In this context, the model TS-NB is the special case with $\alpha_1 = \alpha_2 = 1$, represented by the large red marker.

We examine the variation in the number of NFIs in the last NSSI block with parameters $T_1 = 1$, $T_2 = 11$, $H_1 = \frac{1}{3}|S|$ and $H_2 = \frac{2}{3}|S|$, $\alpha_1 = 0.9$, and $\alpha_2 = 0.8$ as a function of traffic λ/μ. The findings displayed in Fig. 15 demonstrate that the average number of idle NSSIs tends to remain constant as r increases from 1 to 8 for traffic λ / μ between 100 and 400 and only shows notable variations when λ / μ surpasses 400. In particular, the average number of idle NSSIs is inversely correlated with the value of r when $\lambda / \mu > 400$. This happens because the number of available NFI positions in the last NSSI block, denoted by $(k - r)$, increases as r decreases, increasing the average number of idle NSSIs. On the other hand, $2k - T_2 + 1 = 6$ is the position needed to complete the termination of a NSSI block. Hence, the average number of idle NSSIs tends to rise when $r \leq 2$, while it tends to decrease when $r > 2$. Utilization also tends to increase gradually and shows a distinct differentiation for $\lambda/\mu > 400$ (Fig. 16).

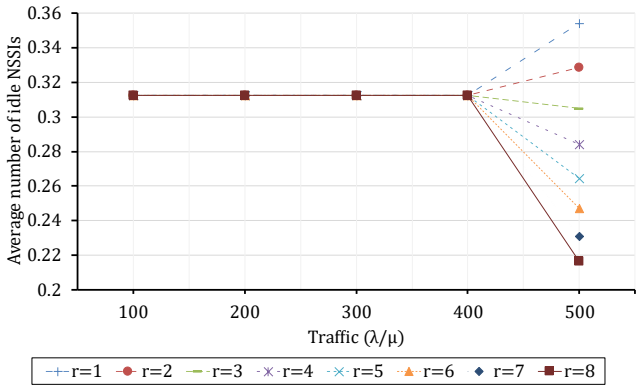


Fig. 15. Analysis results of TSFAC-NB based on the NFI number in the last NSSI for the average number of idle NSSIs.

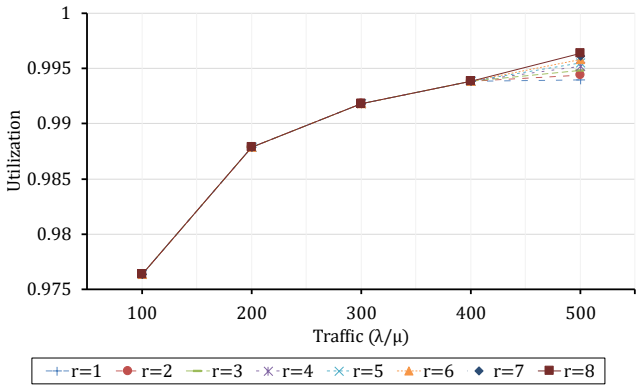


Fig. 16. Analysis results of TSFAC-NB based on the NFI number in the last NSSI for utilization.

In addition to evaluating the model results through analytical methods, as described above, by solving Eq. (2) - (7) to compute the probabilities $p_{i,j}$ and subsequently determining the performance metrics according to Eq. (11) - (12), this paper also assesses the results through simulation to validate the accuracy of the proposed model. In both methods, the algorithm TSFAC will be utilized for deployed/terminate ratio and fraction admission control based on thresholds to ensure the system maintains optimal performance in resource management within dynamic environments.

In our experimental simulation model, Open5GS [22] nodes have been deployed on Kubernetes [23], one of the most widely used container orchestration systems [24]. Accordingly, the virtual functions of the control panel in a 5G network, such as NSSIs, are virtualized using container technology and organized into Pods within Kubernetes, with resources and execution of each Pod managed by Kubernetes. This setup facilitates isolation of the execution environment and optimizes resource utilization. Kubernetes manages these Pods as the fundamental unit of deployment. In this paper, we consider a Pod running a single container that hosts a NSSI image; thus, a Pod can be referred to as a NSSI Pod and a NSSI instance [25]. In the scope of this paper, we use Open5GS deployed on Kubernetes to simulate the threshold-based scaling and fractional admission controlling algorithm (TSFAC algorithm) and compare the results with the analytical outcomes [6]. The simulations are conducted with an average runtime of 10,000 seconds, and due to the Poisson arrival process and exponential service process, the results of each run exhibit slight but not significant variations. We compared simulation results with analytical results based on traffic λ/μ.

In the case of no scaling, it is assumed that all NSSI blocks are deployed $(L = 60)$. This scenario results in significant resource wastage, particularly when the traffic is low, since all NSSI blocks are deployed regardless of their usage. In our simulation, the TS-NB model adjusts the number of NSSI blocks from M $(M = 1)$ to L $(L = 60)$ based on the thresholds T_1 and T_2. In contrast, the TSFAC-NB model employs the TSFAC algorithm to

manage NFIs according to the segments H_1 and H_2, aiming to optimize resource usage.

We present the average number of idle NSSIs and utilization as a function of traffic λ/μ across the no scaling, TS-NB, and TSFAC-NB models, illustrated in Fig. 17 through Fig. 20. For clarity, we have separated the results into individual figures. Figure 17 demonstrates that the average number of idle NSSIs of the model TS-NB is smaller when not using scaling (no scaling), and the model TSFAC-NB with mechanism FAC provides better results for both the average number of idle NSSIs (Fig. 18).

Furthermore, the utilization results highlight the superiority of the TSFAC-NB model over both the TS-NB model and the case of no scaling, as shown in Fig. 19 and Fig. 20.

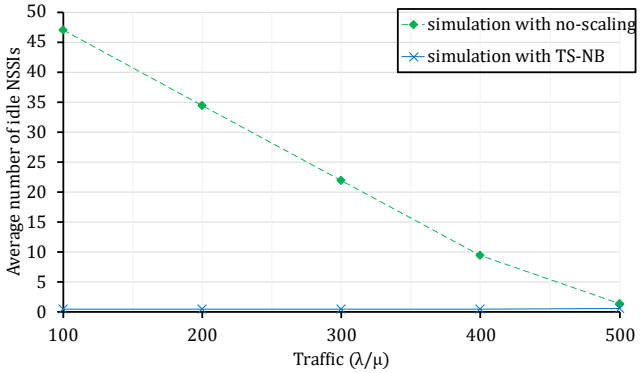


Fig. 17. Simulation comparison results between the model with no scaling and the model TS-NB for the average number of idle NSSIs.

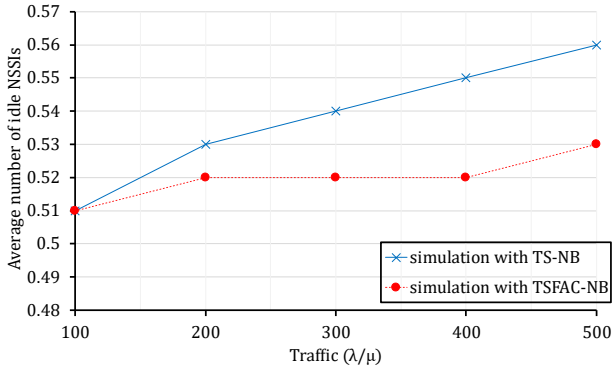


Fig. 18. Simulation comparison results between the model TS-NB and the model TSFAC-NB with the FAC mechanism for the average number of idle NSSIs.

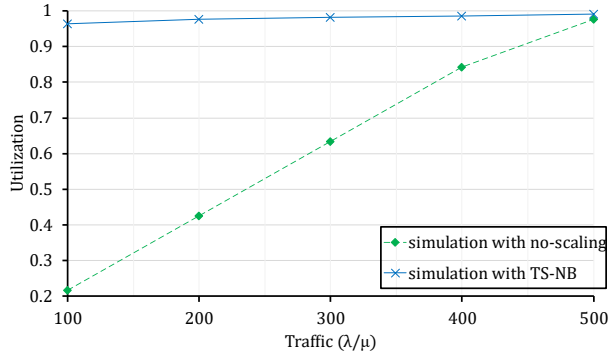


Fig. 19. Simulation comparison results between the model with no scaling and the model TS-NB for utilization.

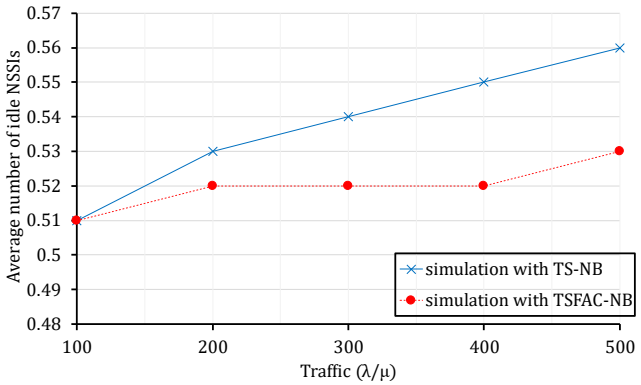


Fig. 20. Simulation comparison results between the model TS-NB and the model TSFAC-NB with the FAC mechanism for utilization.

To verify the accuracy of the model, we conducted an evaluation and comparison between the analytical results and the simulation results based on traffic λ/μ (Fig. 21 and Fig. 22). The results from both analyses and simulations indicate that our TSFAC-NB model aligns well with both theoretical expectations and practical observations. Specifically, the average number of idle NSSIs and utilization metrics from the simulations closely match those observed in practice, with an accuracy exceeding 97%.

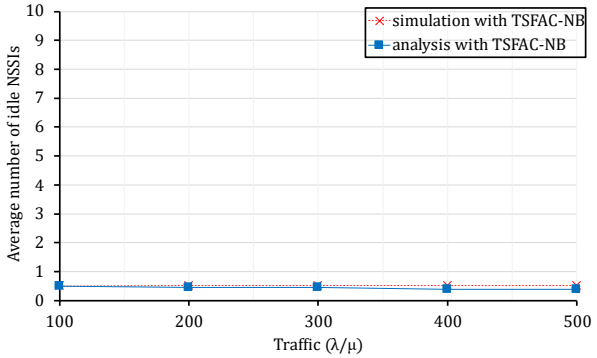


Fig. 21. Comparison results between simulation and analysis for the the model TSFAC-NB regarding the average number of idle NSSIs.

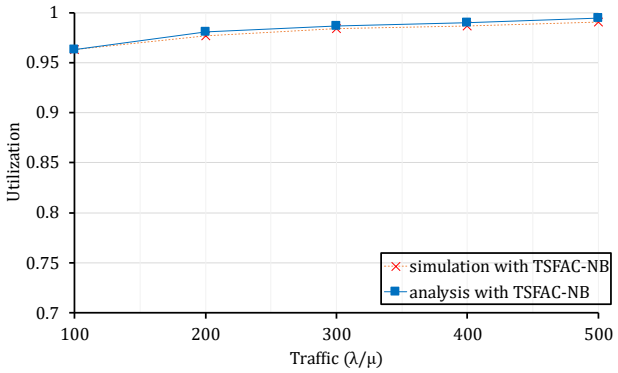


Fig. 22. Comparison results between simulation and analysis for the model TSFAC-NB regarding utilization.

5. Conclusion

In this paper, we have proposed a queueing model for a threshold-based scaling and fraction admisson controlling algorithm that controls the number of NSSI blocks (Q-TSFAC-NB model), including multiple network function versions for 5G slice networks. In addition to using the threshold-based scaling algorithm to scale NSSI blocks according to incoming UE traffic flexibly, our proposed model also integrates the fractional admission controlling mechanism with control thresholds H_1 and H_2 according to the corresponding control probabilities α_1 and α_2 to control the deployment/termination of blocks more efficiently, reduce resource waste, and cost-effectively. In addition to using a threshold-based scaling algorithm to dynamically scale NSSI blocks according to the incoming UE traffic, our proposed model also integrates a fractional admission control mechanism with control thresholds H_1 and H_2 according to the respective control probabilities α_1 and α_2 to control the deployment/termination of blocks more efficiently, reducing the average number of idle NSSI blocks while ensuring better performance, thereby helping to save system resources. Accordingly, a TSFAC algorithm is proposed in the paper and implemented via Kubernetes-based Open5GS to evaluate the effectiveness of the model by both analytical and simulation methods. In the paper, the Poisson arrival process assumption is a common method to build the queueing model mathematically tractable. In the future, we will consider applying other random processes to model non-Poisson arrival processes, as well as applying machine learning and reinforcement learning solutions to the model to obtain better results.

Acknowledgement

This work was supported by the Ministry of Education and Training (Vietnam) for the development of Science and Technology under grant number B2023-DHH-17.

References

[1] C. Y. Hsieh, T. Phung-Duc, Y. Ren and J. C. Chen, "Design and analysis of dynamic block-setup reservation algorithm for 5G network slicing," in *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5140-5154, 1 Sept. 2023, doi: 10.1109/TMC.2022.3169034.

[2] 5G PPP Architecture Working Group, "View on 5G Architecture," Version 3.0, 2020, doi: 10.5281/zenodo.3265031.

[3] 3GPP, "5G; Study on scenarios and requirements for next generation access technologies (Release 16)," 3rd Generation Partnership Project (3GPP), Tech. Rep 38.913, V 16.0.0 0, 2020.

[4] ETSI, "5G; System Architecture for the 5G System (3GPP TS 23.501 version 15.2.0 Release 15)," European Telecommunications Standards Institute, Tech. Rep. V15.2.0 15, 2018.

[5] 3GPP, "The Mobile Broadband Standard." [Online]. Available: https://www.3gpp.org/

[6] C. Rotter and T. Van Do, "A queueing model for threshold-based scaling of UPF instances in 5G core," *IEEE Access*, vol. 9, pp. 81443-81453, 2021.

[7] K. Y. Adou and E. V. Markova, "Methods for analyzing slicing technology in 5G wireless network described as queueing system with unlimited buffer and retrial group," *Communications in Computer and Information Science* (CCIS), vol. 1391, pp. 264-278, 2021.

[8] I. Kochetkova et al., "Analyzing the effectiveness of dynamic network slicing procedure in 5g network by queuing and simulation models," *Lecture Notes in Computer Science*, vol. 12525 LNCS, pp. 71-85, 2020.

[9] T. P. Duc, Y. Ren, J. C. Chen and Z. W. Yu. "Design and analysis of deadline and budget constrained autoscaling (DBCA) algorithm for 5G mobile networks," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Luxembourg, Luxembourg, 2016, pp. 94-101, doi: 10.1109/CloudCom.2016.0030.

[10] Y. Ren, T. P. Duc, J. C. Chen and Z. W. Yu, "Dynamic Auto Scaling Algorithm (DASA) for 5G Mobile Networks," *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1-6, doi: 10.1109/GLOCOM.2016.7841759.

[11] Y. Ren, T. P. Duc, Y. K. Liu, J. C. Chen and Y. H. Lin, "ASA: Adaptive VNF scaling algorithm for 5G mobile networks," in *IEEE 7th International Conference on Cloud Networking (CloudNet)*, Tokyo, Japan, 2018, pp. 1-4, doi: 10.1109/CloudNet.2018.8549542.

[12] V. G. Nguyen, K. J. Grinnemo, J. Taheri, J. Forsman, T. L. Duc and A. Brunstrom, "On auto-scaling and load balancing for user-plane gateways in a softwarized 5G network," in *2021 17th International Conference on Network and Service Management (CNSM)*, Izmir, Turkey, 2021, pp. 132-138, doi: 10.23919/CNSM52442.2021.9615536.

[13] F. A. Cruz-Pérez and L. Ortigoza-Guerrero, "Fractional resource reservation in mobile cellular systems," *Resource, Mobility, and Security Management in Wireless Networks and Mobile Communications*, pp. 349-376, 2006. doi: 10.1201/9781420013610-17.

[14] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, Z. Zhu, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," in *IEEE Network*, vol. 33, no. 6, pp. 172-179, Nov.-Dec. 2019, doi: 10.1109/MNET.2019.1900024.

[15] F. Debbabi, R. Jmal, L. C. Fourati and A. Ksentini, "Algorithmics and modeling aspects of network slicing in 5G and beyonds network: Survey," in

IEEE Access, vol. 8, pp. 162748-162762, 2020, doi: 10.1109/ACCESS.2020.3022162.

[16] A. Banchs, G. de Veciana, V. Sciancalepore, and X. Costa-Pérez. "Resource allocation for network slicing in mobile networks," *IEEE Access*, vol.8, pp. 214696-214706, 2020, doi: 10.1109/ACCESS.2020.3040949.

[17] H. T. Nguyen, T. V. Do, A. Hegyi and C. Rotter, "An approach to apply reinforcement learning for a VNF scaling problem," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Paris, France, 2019, pp. 94-99, doi: 10.1109/ICIN.2019.8685866.

[18] K. C. Tun and K. Kunavut, "Performance analysis of adaptive location update schemes for continuous cell zooming algorithm in wireless networks," *Eng. J.*, vol. 20, no. 1, pp. 137-153, Jan. 2016, doi: 10.4186/ej.2016.20.1.137.

[19] R. K. Saha, Y. Zhao, and C. Aswakul, "A novel approach for centralized 3D radio resource allocation and scheduling in dense HetNets for 5G control-/user-plane separation architectures," *Eng. J.*, vol. 21, no. 4, pp. 287-305, Jul. 2017, doi: 10.4186/ej.2017.21.4.287.

[20] R. K. Saha and Y. Kosuke, "Enabling technology and proof-of-concept evaluation for RAN architectural migration toward 5G and beyond mobile systems," *Eng. J.*, vol. 23, no. 3, pp. 51-74, May 2019, doi: 10.4186/ej.2019.23.3.51.

[21] K. R. Shah and B. K. Sinha, *Theory of Optimal Designs, Lecture Notes in Statistics*, vol. 54. Springer Science & Business Media, 2012.

[22] Open5GS. "Open5GS Document." Accessed: May 2024. [Online]. Available: https://open5gs.org/open5gs/

[23] Kubernetes. "Kubernetes Documentation." Accessed: May 2024. [Online]. Available: https://kubernetes.io/docs/home/

[24] M. N. Tran, D. D. Vu, and Y. Kim, "A survey of autoscaling in Kubernetes," in *2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, Barcelona, Spain, 2022, pp. 263-265, doi: 10.1109/ICUFN55119.2022.9829572.

[25] H. T. Nguyen, T. Van Do, and C. Rotter, "Scaling UPF instances in 5G/6G Core with deep reinforcement learning," in *IEEE Access*, vol. 9, pp. 165892-165906, 2021, doi: 10.1109/ACCESS.2021.3135315.

[26] V. Ziegler, H. Viswanathan, H. Flinck, M. Hoffmann, V. Räisänen and K. Hätönen, "6G architecture to connect the worlds," *IEEE Access*, vol. 8, pp. 173508-173520, 2020, doi: 10.1109/ACCESS.2020.3025032.

Ly Cuong Hoa procured MSc in Computer Science in 2017 from the Hue University of Science, Hue University. He is currently a PhD student majoring in computer science at the College of Science, Hue University. The areas he has engaged in comprise Queueing Theory and Wireless Networks. Email: cuonghl@hueuni.edu.vn.

Thanh Chuong Dang obtained his doctorate in Mathematical Foundation for Computers and Computing Systems in 2014 from the Institute of Information Technology, Vietnam Academy of Science and Technology (VAST). He has published over 30 research papers. His research interests are in the fields of all-optical networks with emphases on packet/burst-based switching, Contention Resolution, and Quality of Service; Queueing Theory and Retrial Queue, 5G Networks. Email: dtchuong@hueuni.edu.vn.