Chapter

Architecture and Operation Algorithms of Mobile Core Network with Virtualization

Larysa Globa, Svitlana Sulima, Mariia Skulysh,
Stanislav Dovgyi and Oleksandr Stryzhak

Abstract

The analysis of the current situation in the wireless communication market shows an increase in the workload, which leads to an increase in the need in additional resources. However, the uneven loading of the infrastructure nodes leads to their loss of use; so, there is a need in introducing technologies that both do not lead to downtime of equipment and ensure the quality of load service during the day. An overview of the NFV virtualization technology has shown that it is appropriate to build wireless networks, since it provides the necessary flexibility and scalability. The method for determining the location and capacity of reserved computer resources of virtual network functions in the data centers of the mobile communication operator, method for determining the size of computing resources constant configuration time interval, and distributed method of local reconfiguration of the virtual network computing resources in the case of a failure or overload are proposed. Thus, configuration, operation, and reconfiguration processes in mobile core network with virtualized functions are described.

Keywords: network function virtualization, evolved packet core, resource allocation, reconfiguration, mobile network

1. Introduction

In the mobile cellular network, the rapid development has been observed. Modern telecommunication systems are being constructed as complex networks that involve various types of devices united into a single complex, operating in conditions of large load flows and large number of connections [1]. They can offer higher data transfer rates, with the integration of more services and guarantee of high quality of experience. Nevertheless, this development also means that the amount of data that is transferred in the mobile network is increasing and the volume of signaling traffic is increasing, respectively. According to [2], it is expected that total mobile data traffic will have increased to 77 exabytes per month by 2022, almost seven times more compared to 2017. Mobile data traffic will grow at an average annual growth rate (CAGR) equal to 46% from 2017 to 2022.

According to Shimojo et al. [3], vehicles, houses, personal devices, robots, sensors, etc. will be connected wirelessly. It means that an automatic and intelligent control system will be achieved. An increase in the number of devices will affect the

IoT market, which is estimated to be $19 trillion [4], and is expected to reach 50 billion [5]. In addition, rich content services, such as real-time streaming movies that require high resolution and tele-surgery requiring small delay must be provided (**Figure 1**).

In addition, the average signaling requirement per subscriber is up to 42% higher in LTE compared to the standard of the past generation communication [6].

Furthermore, market competition requires faster deployment of services and elasticity of changing service criteria as well as the ability to cope with higher service requirements. Therefore, there is a need to manage the signaling traffic in order to provide the necessary quality of service to end users and the proper use of resources of the network operator.

In such circumstances, operators are forced to build up the network infrastructure to ensure the process of service of telecommunication services at a given level of quality. During the day, the load differs, and according to [7], up to 80% of the computing capacity of the base stations and up to half of the capacity of the core network are unused. This leads to a low usage of resources as well as a high level of energy consumption, which reduce the cost-effectiveness of the network for mobile operators.

The emergence of the concept of network functions virtualization opens up new opportunities for the world of telecommunication systems. At the same time, there is a need for new approaches, models, and methods for organizing service handling. The use of virtual servers to solve the tasks of the mobile core network can greatly simplify the process of organizing resources on the service server and ensure its scalability and fault tolerance.

The principle of network function virtualization (NFV) [8] is aimed at transforming network architectures by deploying network functions into software that can run on a standard hardware platform. According to the ETSI [9], the network function is a functional block within a network infrastructure that has defined external interfaces and a defined functional behavior. Network functions are components of the LTE evolved packet core (EPC) network, such as MME, HSS, PGW, and SGW, which for the NFV case will be deployed on the basis of data center system, with the use of leased computing resources (CPU core, memory, disk space, and network interface card), which can be allocated and reallocated in the process of operation depending on actual load requirements.

Thus, the features of NFV can be characterized as follows [10]:

1. Separation of software from hardware. Since the network element is no longer an aggregate of integrated hardware and software entities, the evolution of both is independent of each other. This allows having separate terms of development and maintenance of software and hardware.

2. Flexible deployment of network functions. The separation of software from hardware helps to reallocate and share infrastructure resources; thus, together

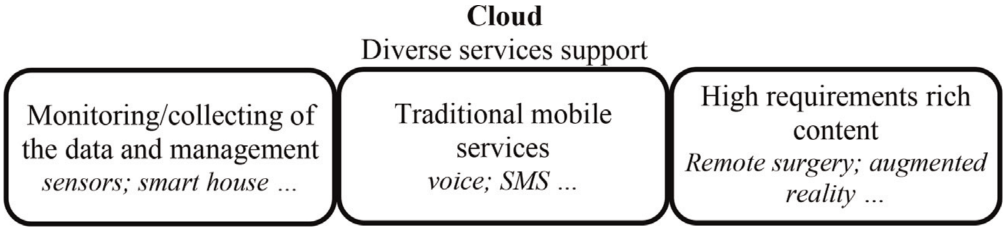

Cloud
Diverse services support
Monitoring/collecting of the data and management
sensors; smart house ...
Traditional mobile services
voice; SMS ...
High requirements rich content
Remote surgery; augmented reality ...

Figure 1.
Services in the era of future generations' networks.

hardware and software can perform various functions at different times. It helps network operators to deploy new network services faster on the same physical platform. Consequently, the components can be created in any NFV-compliant device on the network and their connections can be installed on a flexible basis.

3. Dynamic scaling. Dividing the functionality of the network function into created software components provides greater flexibility in scaling the actual performance of the virtual network function (VNF) more dynamically and with greater details, for example, according to the actual traffic for which the network operator should provide capacity.

At present, numbers of problems remain unresolved. You need to consider hybridity of the service environment, where flexible, well-scalable, virtual servicing entities located in rented cloud-based databases operate along with specialized hardware with limited features. Therefore, the task to organize the computing resources of service nodes and flows between them in a hybrid environment, which consists of hardware telecommunications and virtual computing entities, is important.

Unlike the existing static architecture of the LTE EPC network, a system (**Figure 2**) in which service flows are processed by hardware, and in the case of expected overload, the redistribution of flows happens and takes into account the expansion of the service network by adding virtual service facilities located in the leased clouds of the data centers is proposed (**Figure 3**). After organizing a hybrid service environment, there is a need to adapt the computing resources of the system in the process of operation to ensure a high-quality service, and also it is necessary to consider the features of the reconfiguration process and the costs associated with it. So far, there has not been any comprehensive solution to the task of controlling the computing resources of the hybrid telecommunication environment. The peculiarities of the load distribution of resources of network elements, hardware or virtual ones, have it been considered yet either.

Thus, the chapter proposes a structured approach to the management of resources of network functions through sequential control of the following stages:

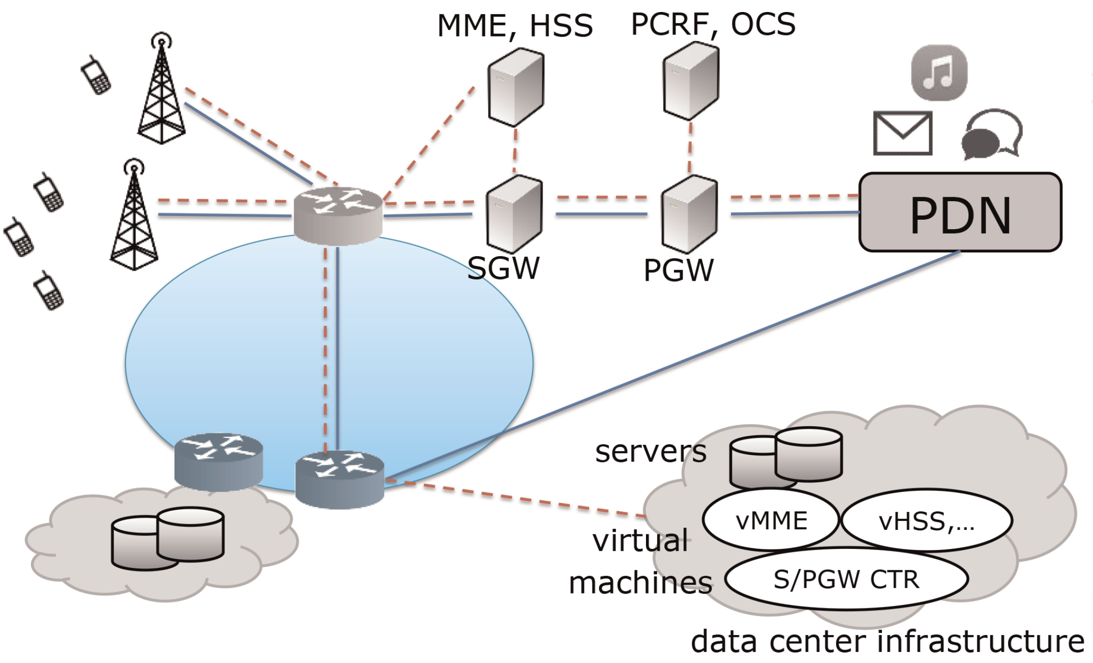


Figure 2.
LTE EPC network architecture with the use of NFV (variable dependent on load volumes).

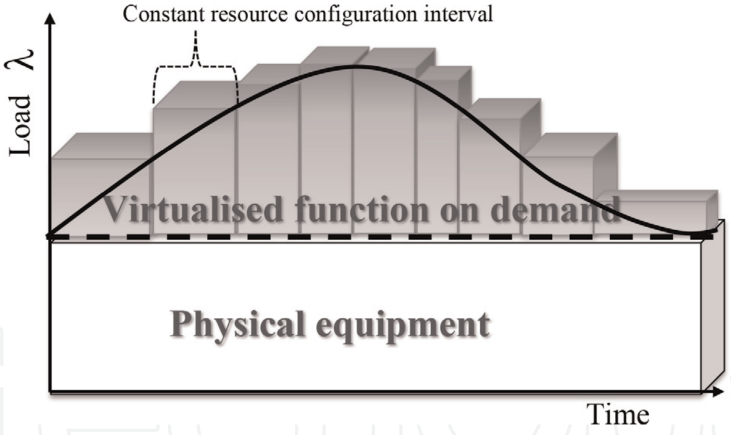

Constant resource configuration interval
Load λ
Virtualised function on demand
Physical equipment
Time

Figure 3.
Distribution of load in hybrid network.

monitoring, forecasting, controlling the sufficiency of resources, and controlling the given level of quality of telecommunication services.

Having analyzed the research and development processes of telecommunication networks of the next generation, we may argue that the existence of powerful data centers greatly expands the possibilities of organizing the process of providing services. One of the key aspects of network virtualization is the allocation of physical resources to virtual network functions. This involves mapping virtual networks on physical networks, as well as managing dedicated resources throughout the life cycle of a virtual network. The optimality and flexibility of resource allocation are key factors for successful network virtualization.

Most of the existing methods for solving the tasks of organizing hardware and virtual resources offer a static distribution of resources, in which, when computing and telecommunication environment is organized, the reallocation of resources does not occur throughout its life cycle. As the network traffic is not static, this may result in improper use of shared computing resources. It is important to organize monitoring of virtual nodes and provide the resources on the basis of their real needs.

2. Method for determining the location and required capacity of virtual reserved computing resources in case of an overload of the physical network

The method is based on the shared embedding concept [11] of the individual virtualized services of the core network on the physical network. We suppose that the virtual network functions of the mobile core network have the same functionality and interfaces as the network components of the 3GPP LTE EPC architecture.

The number of service chains must be determined in advance. The extreme case would be consideration of one service chain for the mobile phone/eNodeB. Since realistic scenarios for mobile networks are up to 10,000 eNodeBs, the resulting optimization model will be enormous and quite long computation time is required to solve it. Therefore, we accept reasonably large clusters of eNodeBs and assume that each of these eNodeB clusters refers to a single service chain of the core network.

Consider the situation when the provider of telecommunication services already has an existing topology of base stations. You need to determine a subset of the network nodes where the load aggregation blocks will be placed which will generate the requests to the same virtualized EPC service. After that, for each base station site, we assign a node of aggregation (traffic aggregation point – TAP).

Let x_i be a binary variable that is equal to 1 if we need to place TAP at point i and equals 0 in the other case. In addition, we define y_{ji} as a binary variable that is equal to 1 if the base station j sends the load to the i TAP and equals 0 in the other case. We need to define the values of x_i and y_{ji} in order to find the optimal value of the objective function.

Objective function (1) aims to minimize network latency. Objective function (2) represents the total cost of placing aggregation nodes and the cost of establishing channels between base stations and the respective TAPs. The objective function (3) aims to leave more free bandwidth on each physical channel. The residual bandwidth of all channels is maximized, since high-downloaded channels can lead to network overload; so, it is advisable to get a solution where more free channels are left.

These optimization goals can be useful for network operators to plan the best deployment strategy.

$$\min_{\mathrm{x_i},\, \mathrm{y_{ji}}} \left(\sum_{\mathrm{i}} \sum_{\mathrm{j}} \mathrm{y_{ji}} \cdot \mathrm{L_{ji}} \right), \tag{1}$$

$$\min_{\mathrm{x_i},\, \mathrm{y_{ji}}} \left(\sum_{\mathrm{i}} \mathrm{x_i} \cdot \mathrm{cost_i} + \sum_{\mathrm{i}} \sum_{\mathrm{j}} \mathrm{y_{ji}} \cdot \mathrm{costl_{ji}} \right), \tag{2}$$

$$\max_{\mathrm{x_i},\, \mathrm{y_{ji}}} \left(\sum\nolimits_{\mathrm{i}} \sum_{\mathrm{j}} \mathrm{y_{ji}} \cdot \left(\mathrm{c_{ji}} - \mathrm{B_{ji}}\right) \right), \tag{3}$$

where L_{ji} is the delay of the communication channel between the site j and TAP i;

$cost_i$ is the cost that consists of two parts: the fixed initial cost f_i, which is responsible for fixed investments such as space and installation of equipment, and the additional costs – $costN_i$ – per unit of processing power on the computing node, where d_i is the amount of computational resources of processing: $cost_i = f_i + costN_i \cdot d_i$;

$costl_{ji}$ is the cost of establishing a connection between the site j and TAP i, and it is determined as a linear combination of the initial fixed cost fl_{ji} and the variable part dependent on the bandwidth B_{ji}, which is necessary for the channel, and the cost of the unit of capacity $costLj$: $costl_{ji} = fl_{ji} + costL_i \cdot B_{ji}$;

c_{ji} – available bandwidth throughput.

It is possible to use the linear combination (4) of Eqs. (1)–(3) with weights a, b, and c, which can be applied not only to give importance the component but also in order to scale the values of these equations for the purpose of converting to comparable values and have meaningful summation:

$$\min_{x_i,\, y_{ji}} \left(\begin{array}{r} a \cdot \sum\limits_i \sum\limits_j y_{ji} \cdot L_{ji} + b \cdot \left(\sum\limits_i x_i \cdot costN_i + \sum\limits_i \sum\limits_j y_{ji} \cdot costL_{ji} \right) - \\ -c \cdot \left(\sum\nolimits_i \sum\limits_j y_{ji} \cdot \left(c_{ji} - B_{ji}\right) \right) \end{array} \right) \tag{4}$$

Subject to:

$$\sum_i y_{ji} = 1 \forall j, \tag{5}$$

$$y_{ji} \le x_i \forall j \forall i, \tag{6}$$

$$\sum_i x_i \le p, \tag{7}$$

$$\sum_j y_{ji} \cdot d_j \le p_i \forall i, \tag{8}$$

$$\sum_i y_{ji} \cdot \left(c_{ji} - B_{ji}\right) \ge 0 \forall j, \tag{9}$$

$$\sum_i y_{ji} \cdot L_{ji} \le T_j \forall j. \tag{10}$$

Restriction (5) ensures that each base station will be connected only one TAP. Restriction (6) ensures that a channel is created between the base station site j and the TAP i only if i was placed.

Restriction (7) ensures that the maximum TAP does not exceed budget p, while (8) is a capacity limit that ensures that the general requirements for processing of all base stations assigned to a specific TAP do not exceed the actual physical resources installed. Restriction (9) makes sure the sufficiency of channel resources for the establishment of channels, and (10) ensures admissibility of delay value, i.e., not exceeding the threshold T_j.

Below we describe the method for solving the problem of placement and the capacity of reserved computing resources of virtual network functions.

Physical network is given in the form of graph $SN = (N, NE)$, where N is a set of physical nodes and L is a set of channels. Each channel $l = (n_1, n_2) \in NE, n_1, n_2 \in N$ has a maximum capacity of $c(n_1, n_2)$ and each node $n \in N$ is associated with certain resources $c_n^{\,i}$, i ∈ R, where R is the set of resource types (CPU cores, memory, disk space, and network interface card). The set of all traffic aggregation points (TAPs), i.e., eNodeB clusters, is denoted as $K \subseteq N$. For each node $n \in N$, $suit_n^{\,k,j}$ is a binary parameter that indicates whether it is administratively possible to deploy a function $j \in V$ on the node n, where V is the set of types of network functions, k service, where $k \in K$.

A virtual mobile core network is represented by a set of services (one service per TAP) which are embedded in the physical network.

The requirements to the bandwidth between two functions, $j1$ and $j2$, $(j1,j2)$ E, referring to the TAP k K service are denoted as $d_k^{(j1,j2)}$. $d_k^{j,i}$ is the amount of computing resource type i allocated to the network function j in the service k. $s_{n,i}^{\,k,j}$ specifies the processing time for the type resource i of the virtual network function j for the service k with one resource unit on node n. The requirements to the admissible processing time of the network function j related to the service k are designated as P_k^j. T_k – the maximum delay for $k \in K$, $L(n_1,n_2)$ is the network latency for the channel $(n_1,n_2) \in NE$.

The goal of optimization is to find the location of the virtualized services of the core network (i.e., the allocation of network functions and the allocation of resources, as well as definition of the ways to transfer traffic between them), so as to minimize the cost of the occupied resources of channels and nodes in the physical network, while satisfying the load requirements $\lambda^{k,j}$. Let us formulate an objective function (Eq. (11)) in the form of a linear combination of two value expressions: the occupied capacity of computing node resources, where the value of resource unit i on node n is denoted by $costN(i,n)$, and the occupied bandwidth of the channels, where *costL(n₁,n₂)* is the cost of the unit of bandwidth of the physical channel *(n₁, n₂)∈NE*.

The following Eqs. (11)–(20) represent the formulation of the optimization problem of mixed integer nonlinear programming. The Boolean variables $x_n{}^{k,j}$ indicate whether the network function *j* associated with the service *k* is located on the physical node *n*. For *j* = TAP, $x_n{}^{k,TAP}$ are not variables but input parameters that indicate where TAP *k* is, i.e.,

$$\mathrm{x_n}^{\mathrm{k,TAP}} = \begin{cases} 1 \text{ if } \mathrm{k = n}, \\ 0 \text{ else} \end{cases}.$$

Similarly, Boolean variables $f_{(n_1,n_2)}{}^{k,(j_1,j_2)}$ indicate whether the physical channel *(n₁,n₂)∈NE* is used for the path between *j₁* and *j₂* for service *k*.

$$\min_{x_n{}^{k,j},\, f_{(n_1,n_2)}{}^{k,(j_1,j_2)},\, d_t{}^{j,i}} \left(\sum_{k\in K}\sum_{j\in V}\sum_{n\in N}\sum_{i\in R} x_n{}^{k,j}\cdot d_k{}^{j,i}\cdot costN(i,n) \right.$$
$$\left. + \sum_{(n_1,n_2)\in L} costL\left(n_1,n_2\right)\cdot \sum_{k\in K}\sum_{(j_1,j_2)\in E} f_{(n_1,n_2)}{}^{k,(j_1,j_2)}\cdot d_k{}^{(j_1,j_2)} \right) \tag{11}$$

$$\text{Subject to } \sum_{n\in N} x_n{}^{k,j} = 1 \forall k\in K, j\in V \tag{12}$$

$$x_n{}^{k,j} \le suit_n{}^{k,j} \forall k\in K, j\in V, n\in N \tag{13}$$

$$\sum_{(w,n)\in NE}\sum_{k\in K}\sum_{(j_1,j_2)\in E} f_{(w,n)}{}^{k,(j_1,j_2)}\cdot d_k{}^{(j_1,j_2)} \le c_n{}^{bdw} \forall n\in N \tag{14}$$

$$\sum_{k\in K}\sum_{j\in V} x_n{}^{k,j}\cdot d_k{}^{j,i} \le c_n{}^{i} \forall n\in N, i\in \{R\backslash bdw\} \tag{15}$$

$$\sum_{k\in K}\sum_{(j_1,j_2)\in E} f_{(n_1,n_2)}{}^{k,(j_1,j_2)}\cdot d_k{}^{(j_1,j_2)} \le c(n_1,n_2) \forall (n_1,n_2)\in NE \tag{16}$$

$$\sum_{(n,w)\in NE} f_{(w,n)}{}^{k,(j_1,j_2)} - f_{(n,w)}{}^{k,(j_1,j_2)} = x_n{}^{k,j_1} - x_n{}^{k,j_2} \forall k\in K, n\in N, (j_1,j_2)\in E \tag{17}$$

$$x_n{}^{k,j}, f_{(n_1,n_2)}{}^{k,(j_1,j_2)} \in \{0,1\} \forall k\in K, j\in V, n\in N, (j_1,j_2)\in E, (n_1,n_2)\in NE \tag{18}$$

$$\sum_{(j_1,j_2)\in E}\sum_{(n_1,n_2)\in L} f_{(n_1,n_2)}{}^{k,(j_1,j_2)}\cdot L(n_1,n_2) \le T_k \forall k\in K \tag{19}$$

$$\sum_{n\in N} x_n{}^{k,j} \sum_{i\in R} \left(\frac{1}{\frac{d_k}{s_{n,i}{}^{k,j}} - \lambda^{k,j}} \right) \le P_k{}^{j} \forall t\in T, j\in V \tag{20}$$

Eq. (12) ensures that for each TAP/service, only one network function of each type is placed. Eq. (13) ensures that the allocation of resources is carried out on physical nodes, which have an administrative opportunity to locate the corresponding network functions. Eqs. (14)–(16) represent restriction for the available resources of physical nodes and channels. Eq. (17) represents a restriction for flow conservation of all paths in the physical network. Eq. (18) ensures that the variables in the task of locating network functions and displaying a path are Boolean.

In order to limit the delays on channels, the delay limit shown in Eq. (19) is also added. And to take into account the necessary performance of the virtual network function, the restrictions for the value of the processing time of the request determined in Eq. (20) are necessary.

It is supposed to solve the problem (11)–(20) in the offline mode at the initial stage. According to the solution, each network function reserves a certain number of resources of the virtual network function based on the assessment of its greatest resource requirements. The instantaneous needs of different network functions are dynamically satisfied by activating the necessary configuration of virtual machines during execution in such a way as to satisfy the guarantees provided for each network function.

3. Method for determining the size of the time interval of the constant configuration of computing resources

The decision when to provide resources depends on the dynamics of traffic loads. Telecommunication loads undergo long-term changes, such as hourly effects or seasonal effects, as well as short-term fluctuations such as unexpected crowds. While long-term fluctuations can be predicted in advance, observing changes in the past, short-term fluctuations are less predictable, and in some cases, unpredictable. The proposed method uses two different approaches for working in conditions of changes that are observed at different time scales. Proactive resource management is used to assess the load and corresponding management, as well as reactive resource management is used to correct long-term errors or to respond to unforeseen overload.

We propose to apply a mechanism which implies dynamic change in the duration of the constant configuration of the resources of the virtual network function, depending on the difference between the maximum load value at a certain base interval and the minimum one. Eq. (21) describes the principle:

$$\mathrm{Int(t)} = max\left(\begin{array}{r} Int_{base}\cdot\left(1-K\cdot\dfrac{\max_{\tau\epsilon(t-I(t-1);t)}\lambda_{basepred}(\tau)-\min_{\tau\epsilon(t-I(t-1);t)}\lambda_{basepred}(\tau)}{\max\lambda_{basepred}}\right);\\ Int_{\min\ base}\end{array}\right), \tag{21}$$

Int is the interval during which the appropriate specified resources will be allocated; Int_{base} is the base value of the interval calculated according to the load discretization approach described below; *K* is the coefficient of the duration change of constant configuration determined by the network operator according to the experiment; $\lambda_{basepred}(t)$ is the average predicted arrival rate in the period t, and $Int_{minbase}$ is the minimum acceptable value of the base interval.

To do this, you need to define the base interval. The goal is to present a daily load pattern, sampling its requests into successive, non-overlapping time intervals with a single representative value in each interval. Load discretization: having a time series X in the interval $[v, \tau]$, time series Y on the same interval is the discretization of the load X, if $[v, \tau]$ can be divided into m consecutive non-overlapping time intervals, $\{[v, \tau_1],[\tau_1, \tau_2],...,[\tau_{m-1}, \tau]\}$, so that $X(j) = r_i$, for all j in i-th interval, $[\tau_{i-1}, \tau_i]$.

The solution for time series discretization (Eq. 22) is given as follows:

$$\sum_{i=1}^{m}\left[\sum_{\tau=\tau_{i-1}}^{\tau_i} u(r_i - X(\tau))\right] + f(m) \rightarrow \min. \tag{22}$$

Eq. (22) is an objective function which has to be minimized, where X is the time series and $f(m)$ is a function of the value of the number of changes or intervals, m. The purpose of Eq. (22) is to simultaneously minimize the load representation error and number of changes. Basic interval is calculated as $Int_{base} = \frac{\tau_m}{m}$. In order to determine the optimal value of the interval, we set different values of the number of intervals, calculate the value of the Eq. (22) and choose the best, i.e., the minimum one, having for each interval:

$$r_i = \max_{\tau\epsilon(\tau_{i-1},\,\tau_i)} X(\tau). \tag{23}$$

At the same time, it is proposed to continuously monitor the values of the request arrival rate and use the predicted values if the load does not exceed the threshold; otherwise, current trends are evaluated and resources are scaled on the basis of the new forecast.

Load forecasting for the next time interval is carried out by taking into account long-term statistics and adjusting it according to the model of exponential smoothing, where errors of more recent past periods have a greater importance factor:

$$\lambda_{pred}(t) = \lambda_{basepred}(t) + \alpha \sum_{j=t-h}^{t-1} (1-\alpha)^{t-1-j} \cdot \left(\lambda_{obs}(j) - \lambda_{basepred}(j)\right)^{+}, \tag{24}$$

α (smoothing constant) is the coefficient that characterizes the weight rate reduction and takes values from 0 to 1; the closer the value of this parameter is to 1, the better is the consideration of the influence of the last levels of the series during the forecast. The model parameters are set by the network administrator according to the experiment. $\lambda_{obs}(t)$ is the request arrival rate on interval t, h is the interval of previous observations, which is considered by the algorithm, and x^{+} denotes $max(0,x)$.

4. Method of local reconfiguration of network computing resources in case of failure or overload

There might be situations when the resources available on the nodes will be insufficient or if the node fails. Potential failures can be physical nodes failures, failures of servers that have higher failure rates than telecommunication hardware or the infrastructure provider will perform node maintenance tasks and this will require the migration of nodes.

For this case, the methods of reconfiguration are used which seek to find the places for migration of network functions from the affected nodes, minimizing the cost of recovering the node after failure and maintaining a high level of physical performance of the network. The proposed improved recovery methods differ from existing ones by taking into account the cost of resources on the nodes and the final quality of service, as well as the case of node overload. In addition, in previous research, the problem of locating management nodes, which are coordinators of the movement of virtual network functions, remained unsolved.

MN represents a set of control nodes (hereinafter–managers), where managers $MN \quad N$ are responsible for the operation of the proposed recovery mechanism after the failure. Each control node is connected to one or more nodes in the physical network and performs the steps required to recover from the failure. Let us assume that managers can be located in nodes N. For a given number of managers A, there is a finite set of possible $\binom{|N|}{A}$ locations, so the task of placing managers is the task of multi-criteria combinatorial optimization. The purpose of the task is to determine the location of each manager at a given number of A, so that the general cost function $U_A(\{p_n{:}n \quad N\})$ can be minimized, where p_n is a Boolean variable equal to 1 if the manager is placed at the point n. The task of optimization will be given as follows:

$$\begin{aligned} &\min_{\{p_n:n\in N\}} \; U_A \\ &subject\ to \sum_n p_n = A \end{aligned}. \tag{25}$$

The main purpose of the optimal placement of managers is to minimize delays between nodes and managers in the network. However, considering only delays is not enough. The placement of managers should also take into account certain restrictions of stability. **Figure 4** shows different issues that need to be considered when evaluating the stability of the placement. Below we will briefly explain these issues and what is needed to be sustainable in relation to them. **Figure 4** shows normalized delays between nodes and arrival rate at nodes.

Let us presume that the nodes are assigned to their closest manager, using as the metric of delay, i.e., the shortest path $dl_{g1,g2}$ between the node *g1* and the manager *g2*. The number of nodes per manager may be unbalanced – the more nodes the manager has to control, the greater is the load per this manager. If the number of site requests to the manager in the network increases, additional delays probability due to the queues in the control system increases too. In order to be resilient to manager overload, the assignment of nodes to different managers should be balanced properly.

It is obvious that one manager is not enough to achieve network resilience. On the other hand, when multiple managers are hosted in the network, the logic of network management is distributed across multiple managers, and these managers must be synchronized to maintain a consistent global state. Depending on the frequency of synchronization between managers, the delay between individual managers plays an important role.

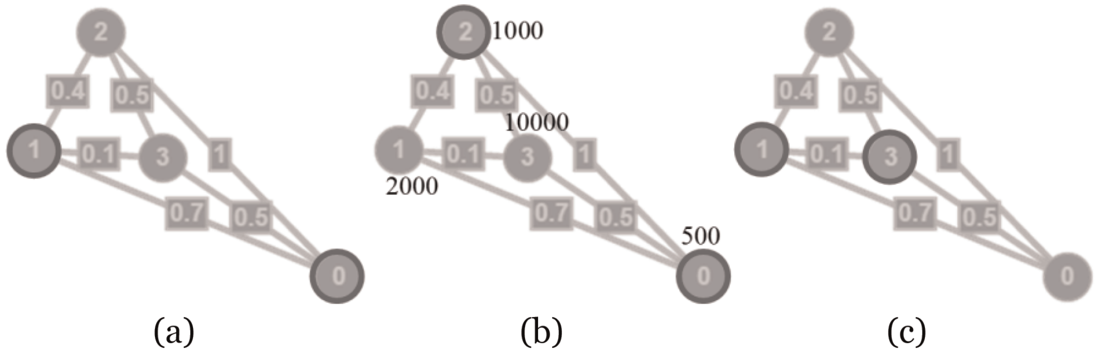


Figure 4.
Assignment according to different criteria: (a) minimal delay to the manager; (b) a minimum load imbalance of the managers; and (c) the minimum delay between managers.

Based on the *dl* matrix, which contains the distance of the shortest paths between all nodes, the maximum transmission latency between the node and the manager for a certain placement of managers can be calculated as follows:

$\mathrm{U_A}^{\mathrm{latency}}(\mathrm{p}) = \max(\mathrm{ddc_n})$,

ddc_n is the maximum transmission delay from the network node to the manager at the point n;

ddc_n is calculated as follows:

$$ddc_n = \max_{g\in N} latency_g \cdot \pi_{g,n},$$

where $latency_g$ is the delay between manager and node g,
$latency_g = \min_{\{n: n\in N\cap p_n=1\}} dl_{g,n}$;

$\pi_{g,n}$ is a Boolean variable equal to 1 if node g is served by a manager located at the point n.

We consider not the average, but the maximum delay value, since the average hides the values of the worst case which are important when resiliency needs to be improved.

Depending on the situation, it may be desirable to have an approximately equal load for all managers, so that no manager is overloaded, while others have little work. Next, we consider the balanced distribution of nodes between managers. As a formal metric, we introduce the balance of placement, or rather, the imbalance, $U_A{}^{imbalance}$, i.e., the deviation from the fully balanced distribution, as the difference between the load for the most downloaded manager and the least downloaded manager.

$U_A{}^{imbalance}$ is calculated as follows:

$U_A{}^{imbalance}(p) = max(ldc_n) – min(ldc_n)$, де $ldc_n > 0$,

ldc_n – manager load at n;

ldc_n is given as follows:

$$ldc_n = \sum_{g\in N} load_g \cdot \pi_{g,n},$$

where $load_g$ is the load factor for the node g.

As the last aspect of a resilient placement of managers, let us consider how the delay between managers can be taken into account when choosing managers. Formally, the delay between managers $U_A{}^{interlatency}$ is defined as the greatest delay between any two managers at a given placement:

$$U_A{}^{interlatency}(p) = \max_{\{n,g: n,g\in N\cap p_n=1, p_g=1\}} dl_{g,n}.$$

In general, placement with a delay between managers' considerations tends to place all managers closer to each other. This increases the maximum delay from nodes to managers.

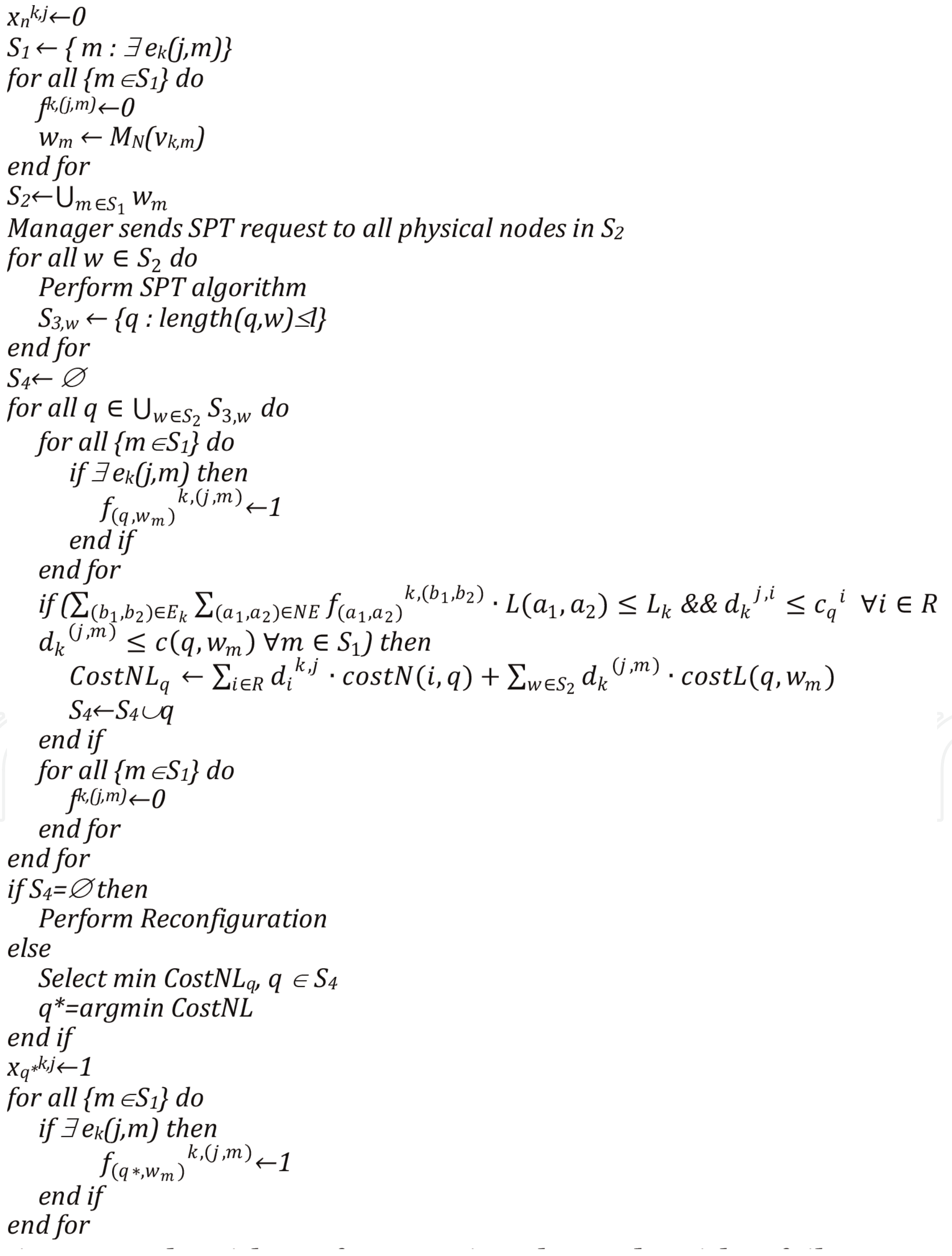
Thus, the target optimization function is given by:

$U_A = wu^{latency} \times U_A{}^{latency}(p) + wu^{imbalance} \times U_A{}^{imbalance}(p) + +wu^{interlatency} \times U_A{}^{interlatency}(p)$,

where wu is the set of importance coefficients.

The recovery algorithm is based on prototype described in [12] but considers modified problem formulation and expands the solution on node overload case.

The physical network is given in the form of a graph $SN = (N,NE)$, where N is a set of physical nodes and NE is a set of channels. Each channel $(n_1,n_2)\in NE$, n_1, $n_2\in N$ has a maximum throughput of $c(n_1,n_2)$ and a network delay $L(n_1,n_2)$, and each node *n N* is associated with certain resources c_n^i, $i \in R$, where *R* is the set of types of resources. The communication network is represented by a set of services (or virtual network requests) *K* that are embedded into the physical network. The virtual network request *k, k*∈*K,* can be given as a graph $G_k = (V_k, E_k)$, where V_k is the set of virtual nodes containing h_k elements and denoted as $V_k = (v_{k,1}, v_{k,2}, \ldots, v_{k,hk})$, where $v_{k,j}$ indicates the *j-th* network function in the service chain of *k*. E_k is the set of virtual channels $e_k(v_{k,j}, v_{k,g}) \in E_k$. The channel throughput requirements between the two functions, *j1* and *j2*, referring to the *k*∈*K* service are marked as $d_k^{(j1,j2)}$, $d_k^{j,i}$ is the number of resource type *i* allocated to the network function *j* in the *k-th* service. The Boolean variables $x_n^{k,j}$ indicate whether the network function *j* associated with *k*∈*K* is located on the physical node *n*, and the variables $f_{(n1,n2)}^{k,(j1,j2)}$ determine whether the physical channel *(n1, n2)* is used in the path between *j1* and *j2* for request *k*. L_k is the maximum delay for request *k*. *costN(i,n)* is the cost of the occupied resource unit on the physical node *n*, and $costL(n_1,n_2)$ is the cost per unit occupied bandwidth on the



```
$x_n^{k,j} \leftarrow 0$
$S_1 \leftarrow \{ m : \exists e_k(j,m) \}$
for all $\{m \in S_1\}$ do
    $f^{k,(j,m)} \leftarrow 0$
    $w_m \leftarrow M_N(v_{k,m})$
end for
$S_2 \leftarrow \bigcup_{m \in S_1} w_m$
Manager sends SPT request to all physical nodes in $S_2$
for all $w \in S_2$ do
    Perform SPT algorithm
    $S_{3,w} \leftarrow \{q : length(q,w) \leq l\}$
end for
$S_4 \leftarrow \varnothing$
for all $q \in \bigcup_{w \in S_2} S_{3,w}$ do
    for all $\{m \in S_1\}$ do
        if $\exists e_k(j,m)$ then
            $f_{(q,w_m)}^{k,(j,m)} \leftarrow 1$
        end if
    end for
    if $(\sum_{(b_1,b_2) \in E_k} \sum_{(a_1,a_2) \in NE} f_{(a_1,a_2)}^{k,(b_1,b_2)} \cdot L(a_1,a_2) \leq L_k \;\&\&\; d_k^{j,i} \leq c_q^i \;\; \forall i \in R$
    $d_k^{(j,m)} \leq c(q,w_m) \; \forall m \in S_1)$ then
        $CostNL_q \leftarrow \sum_{i \in R} d_i^{k,j} \cdot costN(i,q) + \sum_{w \in S_2} d_k^{(j,m)} \cdot costL(q,w_m)$
        $S_4 \leftarrow S_4 \cup q$
    end if
    for all $\{m \in S_1\}$ do
        $f^{k,(j,m)} \leftarrow 0$
    end for
end for
if $S_4 = \varnothing$ then
    Perform Reconfiguration
else
    Select min $CostNL_q$, $q \in S_4$
    $q^* = \arg\min CostNL$
end if
$x_{q^*}^{k,j} \leftarrow 1$
for all $\{m \in S_1\}$ do
    if $\exists e_k(j,m)$ then
        $f_{(q*,w_m)}^{k,(j,m)} \leftarrow 1$
    end if
end for
```

Figure 5.
Algorithm of recovering the node with a failure.

physical channel $(n_1,n_2)\in NE$. $suit_n^{k,j}$ means that the *j* function *k* can be placed on node *n*.

The process of moving the nodes of the virtual network hosted on the failed node, $v_{k,j}^{fail}$, starts when the system sends a recovery request to the corresponding host manager. The recovery process for each failed virtual node proceeds as follows: the manager sends the recovery request to all nodes of the physical network, which hosts the virtual nodes adjacent to the affected virtual nodes. Each of these nodes builds the shortest path tree (SPT) to all nodes of the physical network at a distance of not more than *l* (the threshold is set by the service provider) from the node, where the SPT root is the node. The manager uses these paths to select the node with the optimal distance to all nodes in the physical network, where the nodes of the virtual network are located adjacent to the failed node. This node eventually becomes the best candidate for hosting the affected virtual host. In addition, the capacity of the end nodes of the paths with the SPT should be at least the capacity of the virtual node located on the failed node. We select a node with a minimum cost of the path to all root nodes in the SPT trees and the minimum processing cost. **Figure 5** contains a description of the pseudo-code of the recovery algorithm (**Figure 6**) after failure and is applied for all $\{v_{k,j}: x_n^{k,j} = 1 \,\&\, n = failed\}$.

There is also a probability of the node failure due to overload. To perform a recovery in an overloaded network, the reconfiguration procedure is performed to migrate the virtual nodes hosted on the overloaded physical node.

The recovery process begins with sorting all the virtual nodes located on the overloaded physical node. The criterion (CRT in **Figure 7**) used to sort these nodes in a virtual network is the capacity of the virtual nodes. Then, the recovery

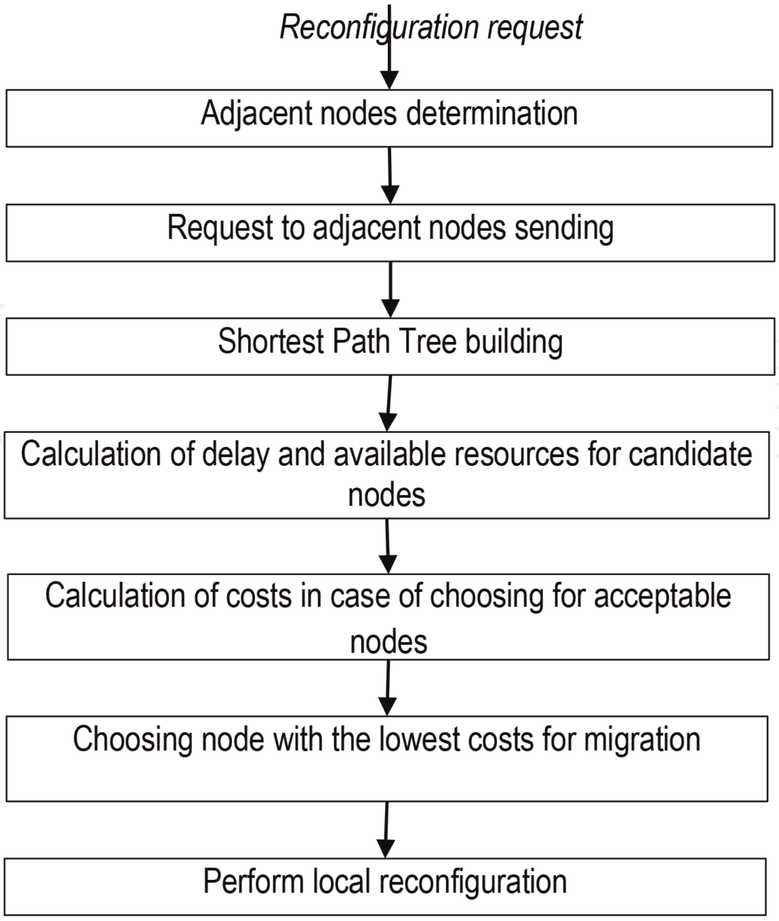
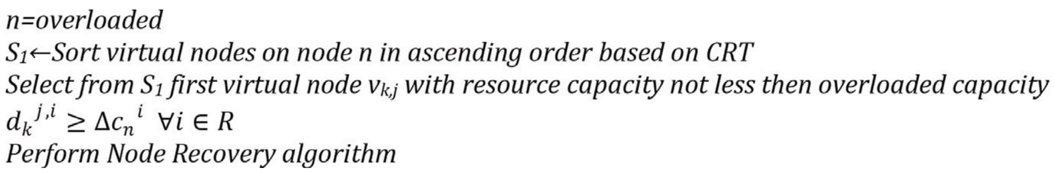

Reconfiguration request
Adjacent nodes determination
Request to adjacent nodes sending
Shortest Path Tree building
Calculation of delay and available resources for candidate nodes
Calculation of costs in case of choosing for acceptable nodes
Choosing node with the lowest costs for migration
Perform local reconfiguration

Figure 6.
Recovering the node with a failure.



```
n=overloaded
S_1←Sort virtual nodes on node n in ascending order based on CRT
Select from S_1 first virtual node v_{k,j} with resource capacity not less then overloaded capacity
$d_k^{j,i} \geq \Delta c_n^{\,i} \;\; \forall i \in R$
Perform Node Recovery algorithm
```

Figure 7.
Recovering the node with overload.

procedure is performed on the first sorted virtual network node, which has a capacity equal to the overloaded, to migrate to the new node of the physical network.

When the load or resources change, some virtual network functions (VNFs) may have to be moved. There is a probability that finding a new node candidate for a node of a virtual network hosted on a failed site will not be possible. In this case, the reconfiguration procedure is performed to migrate one or more virtual nodes. Let us consider the problem of migration as an optimization problem, which is aimed at minimizing the general migration costs with the limits of permissible delay and computational resources.

The goal of optimization is to find the location of virtual network functions (i.e., the location of network functions and resource allocation as well as channels to transfer traffic between them), so as to minimize the cost of the occupied resources of channels and nodes in the physical network, while satisfying the requirements of traffic. Let us give the objective function (26) in the form of a linear combination (with weighted coefficients a, b, c, and e) of the cost expressions.

Let us determine the binary variable $x_n^{k,j} \in \{0,1\}$ to indicate that VNF j is associated with the service chain k placed on the node n after migration. The indicator $x_n^{k,j} = 0$ means that VNF j is not placed on node n after migration; otherwise, j is placed on node n after migration.

Then, we enter the binary variable $y_n^{k,j}$ to display the network status before migration. It is similar to variable $x_n^{k,j}$, $y_n^{k,j} = 0$ means that VNF is not located on node n before migration; otherwise, j is located on node n before migration.

Thus, we can use the $I^{k,j}$ indicator to indicate whether the VNFj by k service was moved in the current migration process.

$$I^{k,j} = \sum_{n \in N} x_n^{\,k,j} \cdot y_n^{\,k,j}$$

$I^{k,j} = 0$ indicates that the VNF has been moved in the current migration process, and $I^{k,j} = 1$ indicates that the VNF has not been moved.

x_n denotes whether the n physical server works or not after migration.

$$x_n = \begin{cases} 1 \text{ (server n launched)} \\ 0 \text{ (otherwise)} \end{cases}$$

y_n indicates whether the n physical server works or not before migrating.

$$y_n = \begin{cases} 1 \text{ (server n launched)} \\ 0 \text{ (otherwise)} \end{cases}$$

In order to consider the resources that are consumed while migrating, we introduce the following equations:

B_n indicates the required b_n costs to launch the *n-th* server:

$$B_n = b_n x_n (x_n - y_n);$$

$L_i^{k,j}(n \to n')$ denotes the use of resource *i* for the migration of VNF *j* from the service chain *k* from the server *n* to the server *n*':

$$L_i^{k,j}(n \to n') = l_i\left(d^{k,j}\right) + l'_i\left(d^{k,j}\right),$$

where $l_i(x)$ is the function of using resource *i* for migration from the server and $l'_i(x)$ is the use of resource *i* for migration to the server.

The objective function will be calculated as follows:

$$\begin{aligned} MCost = a \cdot \sum_{n \in N} (B_n + x_n \cdot cost(n)) + b \cdot \sum_{n \in N} \sum_{k \in K} \sum_{j \in V} \sum_{i \in R} x_n^{k,j} \cdot d_i^{k,j} \cdot costN(i,n) + c \\ \cdot \sum_{(n_1,n_2) \in NE} costL(n_1,n_2) \cdot \sum_{k \in K} \sum_{(j_1,j_2) \in E} f_{(n_1,n_2)}^{k,(j_1,j_2)} \cdot d_k^{(j_1,j_2)} + e \\ \cdot \sum_{n \in N} \sum_{n' \in N} L_i^{k,j}(n \to n') x_{n'} (x_{n'} - y_n) \end{aligned} \tag{26}$$

Taking everything into account, we formulate the problem as follows.

Objective function:

Min MCost.

With constraints:

$$\sum_{n \in N} x_n^{k,j} = 1 \forall k \in K, j \in V, \tag{27}$$

$$x_n^{k,j} \leq suit_n^{k,j} \forall k \in K, j \in V, n \in N, \tag{28}$$

$$\begin{aligned} \sum_{k \in K} \sum_{j \in V} x_n^{k,j} \cdot d_i^{k,j} + y_n^{k,j} \cdot \left(1 - I^{k,j}\right) \cdot l_i\left(d^{k,j}\right) + x_n^{k,j} \cdot \left(1 - I^{k,j}\right) \\ \cdot l'_i\left(d^{k,j}\right) \leq c_n^i \forall n \in N, i \in R, \end{aligned} \tag{29}$$

$$\sum_{t \in K} \sum_{(j_1,j_2) \in E} f_{(n_1,n_2)}^{k,(j_1,j_2)} \cdot d_k^{(j_1,j_2)} \leq c(n_1,n_2) \forall (n_1,n_2) \in NE, \tag{30}$$

$$\begin{gathered} \sum_{(n,w) \in L} f_{(w,n)}^{k,(j_1,j_2)} - f_{(n,w)}^{k,(j_1,j_2)} = x_n^{k,j_1} - x_n^{k,j_2} \\ \forall k \in K, n \in N, (j_1,j_2) \in E, \end{gathered} \tag{31}$$

$$x_n^{k,j}, f_{(n_1,n_2)}^{k,(j_1,j_2)} \in \{0,1\} \forall k \in K, j \in V, n \in N, (j_1,j_2) \in E, (n_1,n_2) \in NE, \tag{32}$$

$$\sum_{(j_1,j_2) \in E} \sum_{(n_1,n_2) \in NE} f_{(n_1,n_2)}^{k,(j_1,j_2)} \cdot L(n_1,n_2) \leq L_k \forall k \in K, \tag{33}$$

$$\sum_{n \in N} x_n^{k,j} \sum_{i \in R} \left(\frac{1}{\frac{d_k}{s_{n,i}^{k,j}} - \lambda^{k,j}} \right) \leq P_k^j \forall k \in K, j \in V \tag{34}$$

Hence, the objective function (26) is a linear combination of four equations which aims to minimize: the cost of starting and using a server, using server resources, communication channels, and resources for migration. Eq. (27) ensures the one-time allocation of network functions, and Eq. (28) is the administrative possibility of placement on the node. Eqs. (29) and (30) represent a limit for the resources of physical nodes and channels, i.e., they ensure that the amount of resources involved in a node does not exceed the amount of available resources. Eq. (31) represents a flow conservation limit, i.e., the input stream at the node is equal to the output stream. Eq. (32) ensures that the variables in the problem are Boolean. Eqs. (33) and (34) represent a limit for the time of transmission by telecommunication channels and time of processing by service nodes, respectively, and ensure compliance with the specified time requirements for the service.

5. Operating scheme of the resource management system

Thus, before operation starting, it is necessary to have statistics on the requests arrival rate for the network function and the probability characteristics of the request servicing. According to the allocation method, the binding of each network function of the traditional network to the data center and the amount of resources that should be reserved for the corresponding virtualized network function is determined. Next, it is necessary to divide the lifecycle of the network function into intervals during which its configuration will remain unchanged and a certain amount of resources will be activated in accordance with the method of determining the size of the resources constant configuration time interval, while taking into account the expected load. When a mobile network operates, a physical node may not be able to continue to handle an incoming load due to lack of resources or due to its failure, and in this case, a distributed local reconfiguration of resources that re-distributes virtual nodes is triggered.

The general resource management system is shown in **Figure 8**.

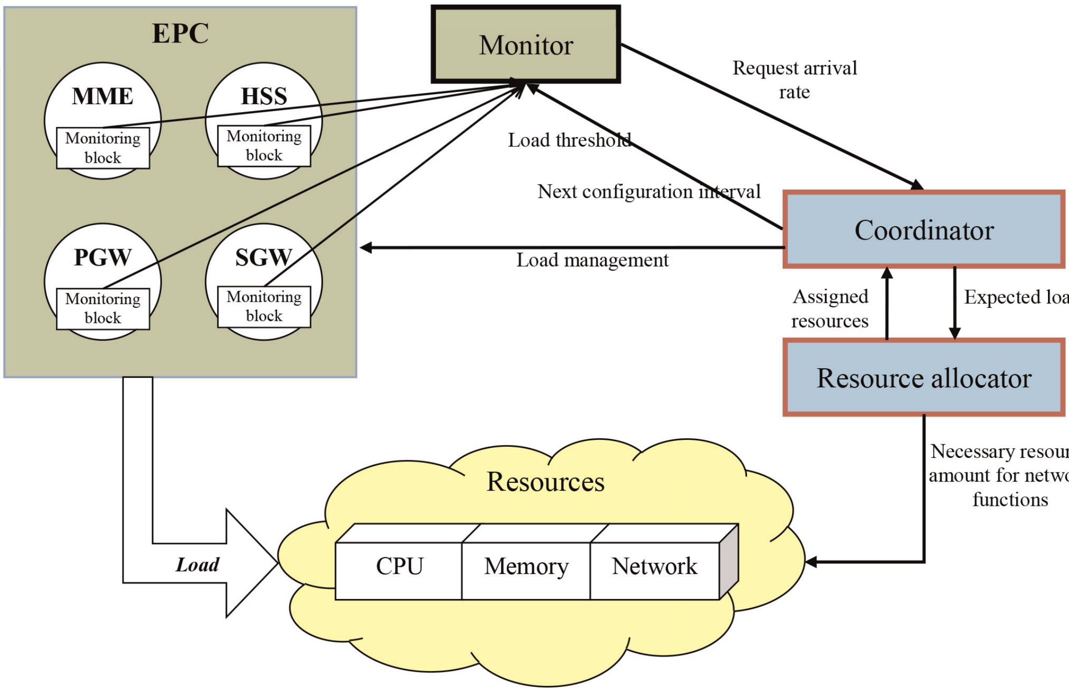


Figure 8.
Modified resource management system.

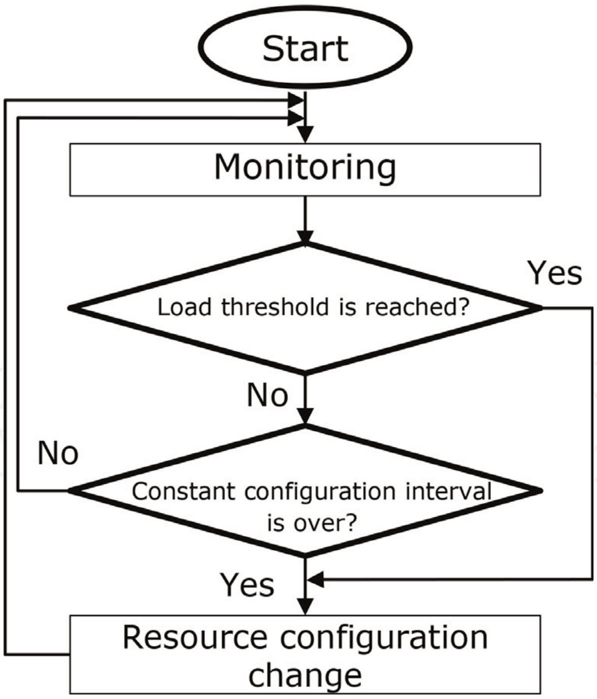

Start
Monitoring
Load threshold is reached?
Yes
No
No
Constant configuration interval is over?
Yes
Resource configuration change

Figure 9.
The method of system resource management.

The monitoring system tracks traffic and counts the number of requests. The monitoring system sets the threshold for the number of requests and sends a message to the coordinator if detecting an overload. When the coordinator accepts an overload message from the monitoring system, the resource allocation unit calculates the required amount of resources to process the applications properly and dynamically distributes the estimated volume. Then, the coordinator redirects the requests and the overload is eliminated.

The coordinator is launched periodically. To predict the base load, one can take the average value of historic daily load. The coordinator sends an incoming load to a data center, which maintains excessive workload, and also exchanges data with a resource allocation unit to provide information about the predicted input load.

The resource distribution module is responsible for distributing the appropriate amount of resources needed to handle the load with the specified quality indicators. During the direct operation of the system, this module is started when the actual load exceeds the base predicted value of the load in order to provide additional resources for excessive load. Since the resource distribution module and coordinator do not start when the actual load is lower than the predicted one, the resource reconfiguration procedure creates minimal additional costs associated with this process.

The general operating scheme of the resource management system is shown in **Figure 9**.

6. Analysis

Quantitative and qualitative analysis (**Figure 10**) of the proposed methods showed a reduction of the cost associated with reserved resources up to 15%, which contributes to increasing the efficiency of load processing, saving computing resources.

The examples of representation of time series values, i.e., loads that illustrate the accuracy of representation, depending on the selected interval of the constant

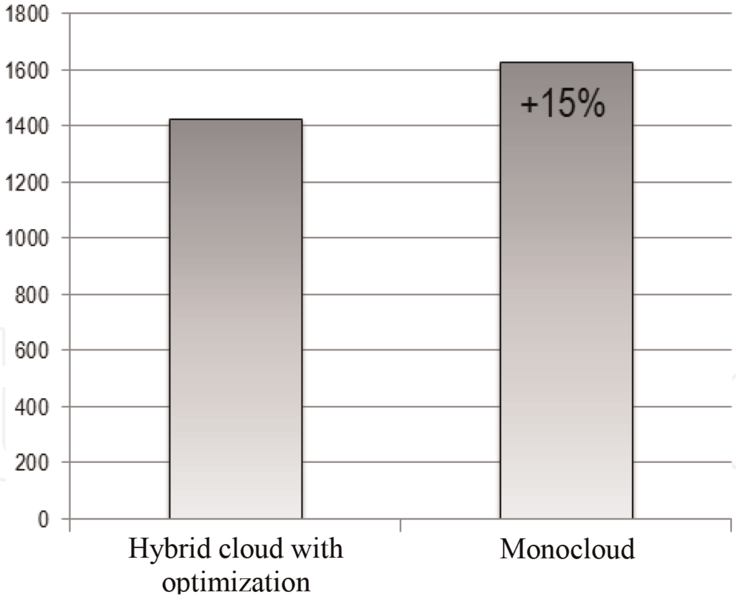


Figure 10.
Consumption of system resources by using the method of allocating virtual network functions and without it.

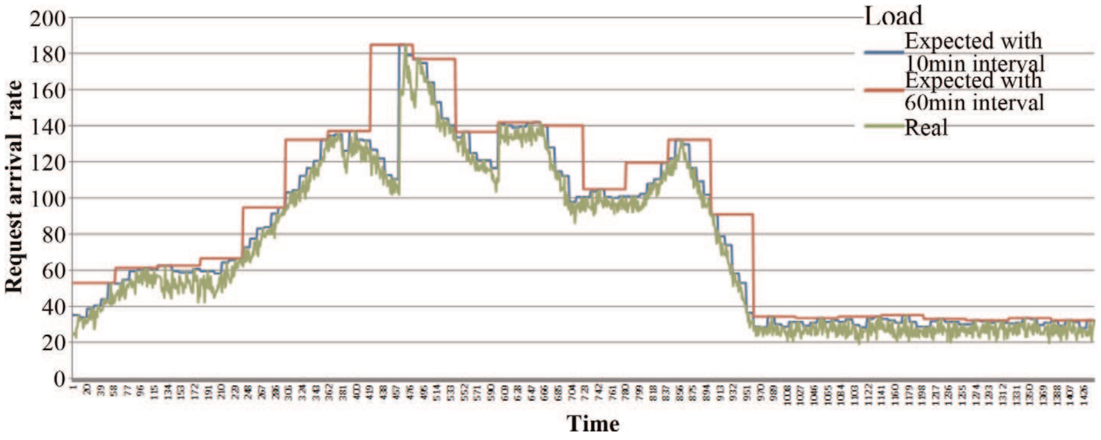


Figure 11.
Representation of load values depending on different values of the constant configuration interval.

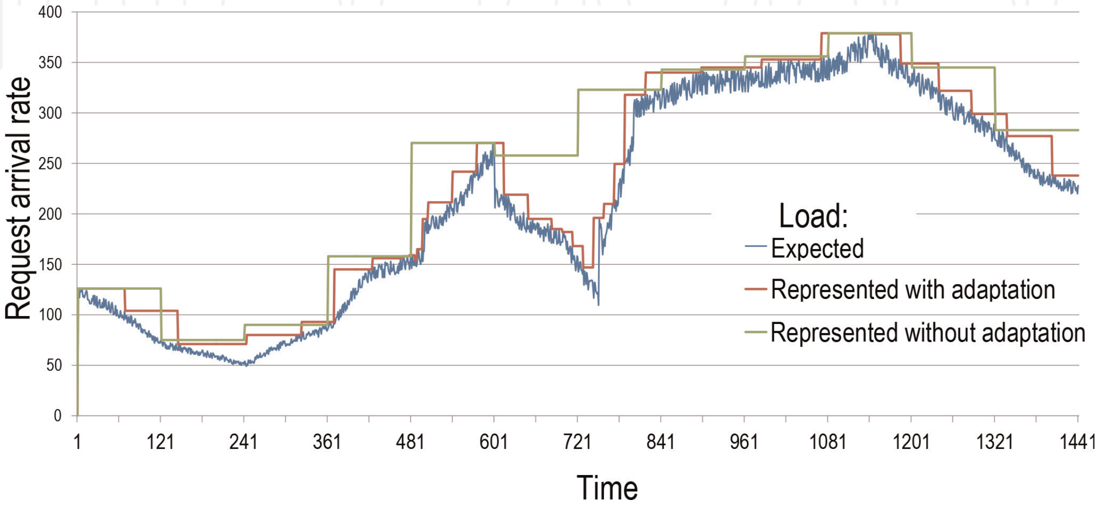


Figure 12.
Results of simulating the system with dynamic change in the value of constant configuration time interval and the system without it.

configuration, are presented in **Figure 11**, where the representation error for the case of intervals in 10 minutes is 7%, and for the case of 60 minutes—19%.

The results of simulation of the method of determining the size of the resources constant configuration time interval (**Figure 12**) showed that the difference between representational value and actual one can be 9%. If you do not apply a dynamic adjustment system to the value of the constant configuration interval, then the deviation will be 18%, i.e., 9% more, and the resources will be spent more.

In order to assess the proposed approach, the average amount of free resources per day was determined as the difference between fixed allocations, i.e., when 100% of resources were always allocated during the day, and dynamically allocated resources by using NFV. According to the results of simulation, the volume of resources allocated dynamically on average is 42% less than in the case of using the traditional distribution approach. **Figure 13** depicts the result of the dynamic distribution of resources in the virtualized EPC of the mobile network in a graphical form. A gray line illustrates the fixed allocations for the worst case scenario. The black curve shows the amount of resources distributed dynamically according to the proposed method.

According to the simulation results (**Figure 14**), the proposed local reconfiguration method showed up to 27% lower costs compared to a strategy aimed at minimizing delay, the delay being within the permissible limits but by 20% greater.

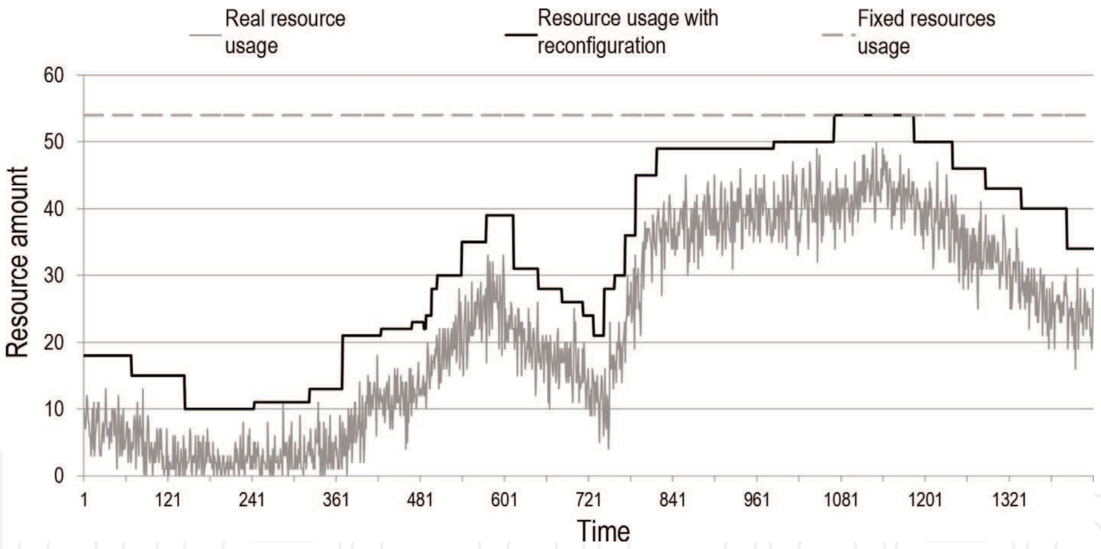


Figure 13.
Results of simulating the system with variable configuration of resources and the system with fixed resources.

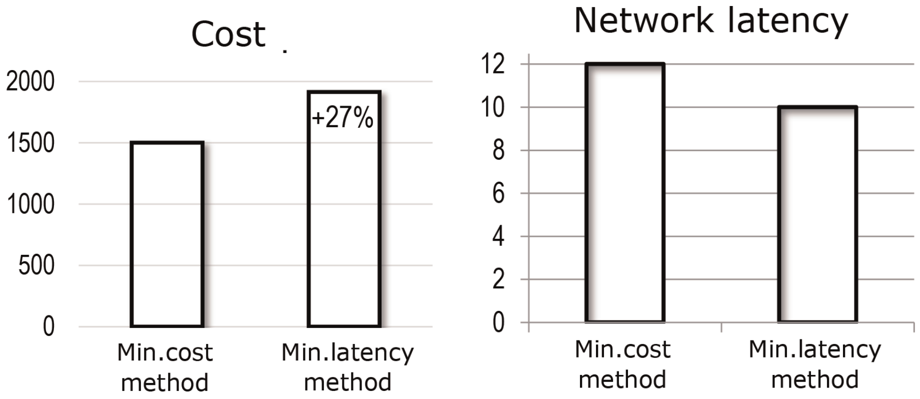


Figure 14.
Results of simulating the system with variable configuration of resources and the system with fixed resources.

7. Conclusions

The main result of the study has become the development of the method for reconfiguring resources of the core network by means of virtualization technology. As a result of the research, the following basic scientific results have been obtained.

The analysis of the current situation in the wireless communication market shows an increase in the workload, which leads to an increase in the need in additional resources. However, the uneven loading of the infrastructure nodes leads to their loss of use; so, there is a need in introducing technologies that both do not lead to downtime of equipment and ensure the quality of load service during the day.

An overview of the NFV virtualization technology has shown that it is appropriate to build wireless networks, since it provides the necessary flexibility and scalability.

We have developed the method for determining the location and capacity of reserved computer resources of virtual network functions in the data centers of the mobile communication operator, which guarantees the quality of providing telecommunication services with the minimum necessary resources by determining their sufficient configuration in a heterogeneous environment of available resources. This allows reducing costs by 13% compared to the randomly selected monocloud and by 47% compared with the traditional approach to deploying the network.

In addition, we have developed the method for determining the size of computing resources constant configuration time interval, which involves its changing and the consideration of both the cost of reconfiguration and the use of resources, as well as provides a flexible use of resources in the virtualized environment, which reduces the percentage of free resources by 42% compared to the dedicated equipment and by 9% compared to existing analogs and reducing the workload on the network.

Furthermore, we have improved the distributed method of local reconfiguration of the virtual network computing resources in the case of a failure or overload, which uses decentralized management and considers migration costs, that redistributes virtual network functions in normal and emergency modes while providing rational resource usage and reducing costs on average by 21%.

Author details

Larysa Globa[1], Svitlana Sulima[1]*, Mariia Skulysh[1], Stanislav Dovgyi[2]
and Oleksandr Stryzhak[2]

1 National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

2 National Academy of Sciences in Ukraine, Kyiv, Ukraine

*Address all correspondence to: itssulima@gmail.com

References

[1] Globa L, Kurdecha V, Ishchenko I, Zakharchuk A, Kunieva N. The intellectual IoT-system for monitoring the base station quality of service. In: Proceedings of the IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom); 4-7 June 2018; Batumi, Georgia: IEEE. 2018. pp. 1-5

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper [Internet]. 2019. Available from: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html [Accessed: 31 March 2019]

[3] Shimojo T, Takano Y, Khan A, Kaptchouang S, Tamura M, Iwashina S. Future mobile core network for efficient service operation. In: Proceedings of the IEEE Conference on Network Softwarization (NetSoft); 13-17 April 2015; London, UK: IEEE. 2015. pp. 1-6

[4] Cisco CEO at CES 2014: Internet of Things is a $19 Trillion Opportunity [Internet]. 2014. Available from: https://www.washingtonpost.com/business/on-it/cisco-ceo-at-ces-2014-internet-of-things-is-a-19-trillion-opportunity/2014/01/08/8d456fba-789b-11e3-8963-b4b654bcc9b2_story.html [Accessed: 31 March 2019]

[5] Emmerson B. M2M: The internet of 50 billion devices. WinWin Magazine. January 2010. pp. 19-22

[6] Signaling is Growing 50% Faster than Data Traffic [Internet]. 2012. Available from: http://docplayer.net/6278117-Signaling-is-growing-50-faster-than-data-traffic.html [Accessed: 31 March 2019]

[7] NSN to Push Cloud Computing to Telco Gear Market [Internet]. 2011. Available from: https://www.reuters.com/article/us-nokiasiemens-gear/nsn-to-push-cloud-computing-to-telco-gear-market-idUSTRE78I6LK20110919 [Accessed: 31 March 2019]

[8] Network Functions Virtualisation [Internet]. 2012. Available from: https://portal.etsi.org/NFV/NFV_White_Paper.pdf [Accessed: 31 March 2019]

[9] Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV [Internet]. 2014. Available from: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_nfv003v010201p.pdf [Accessed: 31 March 2019]

[10] Mijumbi R, Serrat J, Gorricho J, Bouten N, De Turck F, Boutaba R. Network function virtualization: State-of-the-art and research challenges. IEEE Communication Surveys and Tutorials. 2015;**18**:236-262. DOI: 10.1109/COMST.2015.2477041

[11] Baumgartner A, Reddy V, Bauschert T. Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization. In: Proceedings of the IEEE Conference on Network Softwarization (NetSoft); 13-17 April 2015; London, UK: IEEE. 2015. pp. 1-9

[12] Abid H, Samaan N. A novel scheme for node failure recovery in virtualized networks. In: Proceedings of the IEEE International Symposium on Integrated Network Management (IM 2013); 27-31 May 2013; Ghent. Belgium: IEEE; 2013. pp. 1154-1160