

Graph-based quantitative description of networks' slices isolation

Zbigniew Kotulski, Tomasz Wojciech Nowak, Mariusz Sepczuk, and Marcin Alan Tunia
Faculty of Electronics and Information Technology of the Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665, Warsaw

Email: z.kotulski@tele.pw.edu.pl, T.Nowak@tele.pw.edu.pl, msepczuk@tele.pw.edu.pl, M.Tunia@tele.pw.edu.pl

Abstract—5G networks are expected to be a set of slices which are virtual subnets designed for specific applications. A crucial requirement for providing proper functioning of the network and its security is proper isolation of slices. The aim of this paper is to propose a new method of determination of the isolation level of a slice. We propose a Graph-based model of the sliced network, which has a layered structure. In each layer, the appropriate network elements have their own isolation level. The lowest layer of the Graph-based model represents virtual network elements with isolation traits used for calculating their isolation level. Climbing to the top of the stack of layers one can calculate, successively, isolation level for a network's physical element, a link, subnetworks and, the End-to-End slice's isolation level. We present numerical examples, where suitable traits are specified and the isolation level in each layer is calculated.

I. INTRODUCTION

NETWORK slicing is a key technology for 5G network [1]–[3]. In 5G, transmission quality, network performance and services' reliability are expected to be on extremely high-level (e.g., the bandwidth over 300 Mbps, very small latency of few milliseconds and support up to 200,000 devices/km² with 99.999% reliability level, see [4]–[6]). Unfortunately, some quality parameters are impossible to be satisfied simultaneously. Therefore, the network is divided into slices, where each slice is designed for services with required values of network parameters. In such a case, a crucial problem is secure isolation of slices to prevent inter-slice harmful interaction or even attacks and to provide sufficient Quality of Service in each slice, see [7], [8]. Providing proper slices' isolation is now extensively studied, both, from experimental and theoretical points of view.

Even before the concept of 5G network have been rigorously formulated, experimental research related to future networks caused a need of isolation of functions and processes. Different kinds of testbeds have taken into the isolation property. For instance, the COMCON (COntrol and Management of COexisting Networks) project [9] has been created to design novel control and management mechanisms for supporting the coexistence of networks in the Future Internet. It has considered several use cases to evaluate a reference architecture providing some isolation of specialized networks with certain functionalities to provide their dependable and predictable work. Another testbed, described in [10], was specialized for network experiments with disconnected mobile nodes. Here, the isolation property has been required for precise mea-

surements of network's properties. The paper [11] presents a scheme (possibly inside OMF, the wireless testbeds managerial framework) that exploits wireless testbeds functionality by introducing spectrum slicing of the testbed resources. Since in wireless testbeds slicing there are inter-dependencies among the resources, the isolation of experiments is there a hard task. The paper [12] gives an approach allowing virtualization of testbeds to realize several services like environment control, virtual radio control, slice feedback, and a virtual radio isolation. At least two of them provide the isolation of slices: the environment control is responsible for maintaining control and performance isolation across different environments while virtual radio isolation service is required for isolating the radio resources used by each slice due to the inherent nature of the wireless medium.

Practical experiments concerning isolation are presented in paper [13]. The authors consider OpenVZ and User Mode Linux (UML) for virtualization of the ORBIT wireless testbed and evaluate their relative merit. Their results show that the operating system level virtualization mechanism outperforms UML in terms of system overheads and performance isolation. To measure isolation, they propose two performance measurement metrics: transient response and cross coupling between experiments. The transient response is the instantaneous change in throughput of an experiment running on one slice caused due to time varying change in offered load on another slice, while the cross coupling is the difference in throughput with virtualization as a percentage of the throughput without virtualization. Both measures are estimated in experiments. Another experimental testing of isolation can be found in [14], where the authors compare the container-based approach and general virtual machines (Xen). They show that both approaches give comparable isolation features (with respect to fault isolation, resource isolation, and security isolation), while the container-based approach gives better efficiency expressed in terms of overall performance (throughput, latency, etc.) and/or scalability (measured in number of concurrent VMs), what reflects better performance isolation.

Some approach to numerical evaluation of isolation loss using, both, experimental and theoretical results, has been proposed in the paper [15]. Usually in VM environments, the performance isolation is calculated based on performance loss ratio. For containers, we can consider misbehavior and orchestration and management, so the measurements that only

take performance loss into consideration are not sufficient. In the paper [15] the authors propose a performance isolation measurement model that combines the performance loss and resource shrinkage of containers. They also validate their model experimentally using the open-source container project Docker.

In contradiction to widely applied container-based virtualization [16] as a solution for isolating resources of users or slices, the authors of the paper [17] propose an alternative to enable the isolation, based on commodity OS, utilizing existing features in commodity OS. Assigning every user-id in the OS a dedicated and isolated network the address and the routing table, this method enhances the commodity OS with the property of network name-space isolation.

Except of nodes-located information processing, isolation is also required in network processing. For instance, paper [18] proposes a method to share the host's global IP address for all the guest slivers on a node and isolate their network usage in port-space. In programmable networks all VMs can be configured with the same IP and MAC addresses as the host so that any Ethernet frames from outside can be received by a VM or host. To isolate the packets of different VMs, each VM is assigned with a range of port numbers. The port range of each VM can be got from the database of PLC node. As a result, the corresponding flow entries (forwarding rules) are installed after a VM is launched. Each VM can only listen on the ports that assigned to it. The OVS switches packet based on the destination port number. without address translation. The paper [19] combines the programmable switch OpenFlow with network virtualization and design the INP platform OFIAS, i.e., OpenFlow In A Slice. With the flexibility of OpenFlow and the scalable multiplexing of virtualization, OFIAS can smoothly support multi-party INP with good isolation performance.

Next to purely experimental investigations made in testbeds, more theoretical approaches to isolation can be found in the literature. For instance, in the paper [20] the authors present a framework that improves current infrastructure by extending link virtualization with a new component which they call Multi-Hop Virtual Link. In their proposal this component may be implemented as a tunnel which traverses multi-hop physical nodes. For more virtual links, a Link Switch Engine is applied for strong isolation of switch ability offered to different virtual links. The authors propose to allocate separate hardware to each virtual link providing strong isolation among multiple virtual links.

A next step in making a concept of isolation very practical tool for networks is proposing not only qualitative description and experimental validation of isolation, but also its formal modeling and quantitative representation of its level (estimated or calculated from the model). Such a need has been observed, both, in practical investigations and more theoretical research. For instance, the CONFINE IP Project, Community Networks Testbed for the Future Internet [21], has considered the sliver isolation, covering two different aspects for all resources (nodes and a network): the resources

isolation and performance isolation. The resource isolation means that a slice does not interfere with the operation of other slices and is completely separated from the others: it cannot access the data of other slices, cannot kill their processes, and cannot access the core management system (it is secure). Performance isolation means providing mechanisms to guarantee performance on a predictable level with sufficient (up to a certain point) amount of available resources. To make such a concept functional one should describe these intuitive properties and expectation with some measurable parameters.

A milestone of formal modeling of slices is the paper [22], where the authors have presented an abstraction that supports programming isolated slices of the network. The proposed semantics of slices ensures that the processing of packets on a slice is independent of all other slices. Further, in the paper they formally define slices and propose algorithms for compiling slices. Finally, the authors describe a tool for automatic verification of formal isolation properties on a level of network packets processing.

Among experimental papers, a more complete pattern of slices isolation gives the paper [23]. The authors consider several resultant parameters to estimate slices isolation (all for several container-based virtualization implementations), which are: Computing Performance, Memory Performance, Disk Performance, Network Performance, Performance Overhead in HPC Applications, all of them measured according to their own methodology. Finally, they use an Isolation Benchmark Suite (IBS) [24], [25], which includes six different stress tests: CPU intensive test, memory intensive test, a fork bomb, disk intensive test and two network intensive tests (send and receive). Such parameters are suitable for estimating performance of isolated systems in different virtualization environments.

The recent trend of isolation modeling is calculation of overall parameters of the sliced network, like its performance properties, e.g., end-to-end (E2E) delay for a slice [26], using detailed transmission or nodes' parameters. In this paper we propose a new Graph-based model suitable for isolation modeling of slices. It uses several isolation parameters (proposed in our earlier paper [27] and makes possible to establish a common level of isolation for an E2E slice in the 5G network.

The rest of the paper is organized as follows. In Section 2 we introduce a Graph-based model of slices and isolation. It uses hierarchical graphs and makes possible to calculate isolation level on a given level of abstraction. Section 3 is an overview of parameters and properties suitable to model isolation of network's elements. In Section 4 we give a mathematical background for calculating isolation level presenting suitable methods and formulas. In Section 5 we illustrate the theoretical results of previous Sections with two numerical examples, while Section 6 concludes the paper and outlines future work.

II. GRAPH-BASED MODEL OF SLICES AND ISOLATION

All communication networks can be considered as a set of interconnected layers, including their different elements and roles. The same approach can be also applied to 5G networks

(see Figure 1). The highest layer represents a high-level view on 5G network which includes two main subnetworks: RAN and CN and the gateway between them. The lower layer refers to all resources located in a selected subnet. The next layer contains a physical resource which can be described by some properties and parameters that define it. The required virtual resources can be located in the last layer. The virtualization was made based on the mentioned earlier properties.

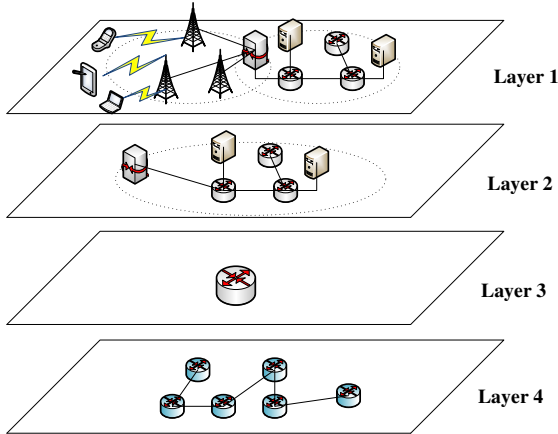


Fig. 1. The 5G network layer decomposition.

Of course, such a general decomposition must be appropriately adapted and transformed to represent a slice-based 5G network with an isolation assurance. For that reason, a new Graph-based layered model of the network has been created.

A. Model assumptions

The described model has been created based on a few assumption, such as:

- RAN, CN and every resource (e.g., router, link, switch, server, etc.) are represented as hypergraphs;
- properties of resources are represented as graph vertices;
- one property can be divided into several virtual properties used in a slice's structure and a value can be assigned to each of them; based on these values it is possible to create requirements how a slice can be created or validate if it is possible to create a slice with a defined set of properties;
- the stratification enables to consider isolation on many levels (e.g., in Layer 5 isolation exists between properties), so it is easy to show a slice as a path with vertices which represent virtual properties.

B. Graph-based model

Our proposed Graph-based architecture is shown in Figure 2. The architecture includes 5 layers. Each of them represents different aspect of slicing:

- Layer 1: the layer consists of hypergraphs with all resources of RAN and CN. Moreover, both areas have

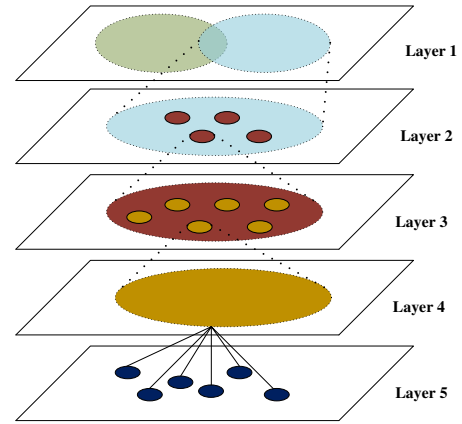


Fig. 2. The 5G Graph-based model.

common part (common resources which match RAN slice and CN slice, e.g. gateway connecting RAN and CN);

- Layer 2: the layer is represented by selected hypergraph of one of area: RAN or CN with sub-hypergraphs (or a second level hypergraph) The sub-hypergraphs apply to all physical resources in this area;
- Layer 3: the sub-hypergraph in this layer refers to dedicated resource which has several properties (in isolation sense), the graph vertices;
- Layer 4: in this layer a property is a graph vertex which can be virtualized;
- Layer 5: this layer includes all virtual properties, created from a vertex in Layer 4, which are the foundations of slices.

III. PROPERTIES AND PARAMETERS OF ISOLATION

In the paper [27] network has been described in terms of *properties* and *parameters*, called in general the *traits*. Both should be normalized; in the paper [27] has been proposed an example method for this operation. The method assumes that each trait's value is preprocessed with the *normalization function* $g : \Lambda \rightarrow \Omega$, where Λ is the trait's domain and Ω is a continuous subsection of the real line \mathbb{R} ; further in this paper we will assume that $\Omega = [0, 1]$. Different types of traits could have different normalization functions, however the Ω should be common for all the traits. The normalization function for the one trait could change between vertices as well, see [27]. The value $\alpha = \inf_{x \in \Omega} x$ will be assigned to the *worst value* of the trait (from the isolation point of view). The value $\omega = \sup_{x \in \Omega} x$ will be assigned to the *best value* of the trait (from the isolation point of view). In the paper [27] there were defined the following trait families:

- *raising trait*: higher trait's value is better (e.g., available link's throughput in *Mbit/s*);
- *falling trait*: lower trait's value is better (e.g., link's BER);

- *Gaussian trait*: trait is the raising trait for $x < \gamma$ and the falling trait for $x > \gamma$, where $\gamma = const$ (e.g., jitter in the packets stream).

The normalization function should satisfy the following assumptions:

- $g(x) \leq g(y)$ iff y is a better trait's value than x ;
- $g(\beta) = C$, where C is a constant dependent on the Ω set and the β was defined in [27] as the trait's *typical value*. In the paper [27] was proposed $C = \frac{\alpha + \omega}{2} = 0.5$.

The Table I contains the example of normalization functions for typical values' domains. Some functions could be parametrized by additional parameters independent from functions' arguments, e.g. β, γ, q, r parameters from the Table I.

IV. CALCULATING THE ISOLATION LEVEL

At the beginning, we need to define types of vertices. Let us assume that the subset of vertices from the Layer \mathcal{L} is indicated as $\{V_1, V_2, \dots, V_n\}$. These vertices can be classified as:

- 1) *similar vertices*, when all of them are described by a common set of traits $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\}$;
- 2) *non-similar vertices*, otherwise.

In fact, the second type of vertices is generalization of the first vertices' type. This observation will be used in a further part of this Section.

A. Isolation-merging function

1) *Calculation for similar vertices*: For the vertex $V_k, k = 1, 2, \dots, n$ we will define a vector \mathbf{I}_k of isolation traits as:

$$\mathbf{I}_k \stackrel{\text{def}}{=} (p_{k,1}, p_{k,2}, \dots, p_{k,m})^T. \quad (1)$$

Now we can propose a formula for calculating the isolation level \mathbf{I} for the Layer \mathcal{L} as:

$$\mathbf{I}(\{V_1, V_2, \dots, V_n\}) \stackrel{\text{def}}{=} F(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n), \quad (2)$$

where the function $F : (\Omega^m)^n \rightarrow \Omega^m$ is a *general merging function*. In this paper we will assume that traits are independent, so the function F can be defined as:

$$F(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n) = \begin{pmatrix} f_1(p_{1,1}, p_{2,1}, \dots, p_{n,1}) \\ f_2(p_{1,2}, p_{2,2}, \dots, p_{n,2}) \\ \vdots \\ f_m(p_{1,m}, p_{2,m}, \dots, p_{n,m}) \end{pmatrix}, \quad (3)$$

where $f_i : \Omega^n \rightarrow \Omega$, for $i = 1, 2, \dots, m$, is a *merging function*. This model is a first level of an approximation, where each trait could be changed independently and one's trait's value does not affect other trait's value.

Now let consider the merging function $f(x_1, x_2, \dots, x_n)$ and mark as $x_{min} = \min\{x_1, x_2, \dots, x_n\}$ and $x_{max} = \max\{x_1, x_2, \dots, x_n\}$. Let assume that the function satisfies the following assumptions:

$$(\forall x \in \Omega) \quad f(x, x, \dots, x) = x, \quad (4)$$

$$(\forall 1 \leq k \leq n)(\forall x_k \leq y_k) \quad (5)$$

$$f(x_1, x_2, \dots, x_k, \dots, x_n) \leq f(x_1, x_2, \dots, y_k, \dots, x_n).$$

Those assumptions define our view over the isolation - if the system is built from the components with the same trait's value, then the system has the same trait value as those components (4), and the isolation will not decrease when a component of a system has been enhanced (5). From these two assumptions one can deduce the following inequalities:

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &\leq f(x_{max}, x_2, \dots, x_n) \leq \\ &\leq f(x_{max}, x_{max}, \dots, x_n) \leq \dots \leq f(x_{max}, x_{max}, \dots, x_{max}) = \\ &= x_{max}, \end{aligned} \quad (6)$$

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &\geq f(x_{min}, x_2, \dots, x_n) \geq \\ &\geq f(x_{min}, x_{min}, \dots, x_n) \geq \dots \geq f(x_{min}, x_{min}, \dots, x_{min}) = \\ &= x_{min}. \end{aligned} \quad (7)$$

The function which satisfies the inequalities:

$$x_{min} \leq f(x_1, x_2, \dots, x_n) \leq x_{max} \quad (8)$$

will be called *the mean function*, see [28]. An example of a mean function is the generalized weighted mean (the power mean):

$$\omega_q(x_1, x_2, \dots, x_n) = \left(\frac{\sum_{i=1}^n w_i x_i^q}{\sum_{i=1}^n w_i} \right)^{\frac{1}{q}}, \quad (9)$$

where $\sum_{i=1}^n w_i > 0$ and $w_i \geq 0$. The parameter q is a real number; there are the following important border-case formulas:

$$\begin{cases} \omega_{-\infty}(x_1, x_2, \dots, x_n) = \min\{x_1, x_2, \dots, x_n\}, \\ \omega_0(x_1, x_2, \dots, x_n) = \sqrt[q]{x_1 x_2 \dots x_n}, \\ \omega_{+\infty}(x_1, x_2, \dots, x_n) = \max\{x_1, x_2, \dots, x_n\}. \end{cases} \quad (10)$$

2) *Calculations for non-similar vertices*: More common situation is when not all considered vertices are described only by one set of traits. Let us assume that the vertex V_k is described by the set of traits $\Pi_k = \{\pi_{k,1}, \pi_{k,2}, \dots, \pi_{k,m_k}\}$ and let us define $\Pi = \bigcup_{k=1}^n \Pi_k = \{\pi_1, \pi_2, \dots, \pi_{|\Pi|}\}$. The goal of the reasoning is to represent the merged isolation in a space common for all vertices. Now, let us introduce the set $\Omega^* = \Omega \cup \theta$, $\theta \notin \Omega$, which has the following properties:

$$\begin{cases} (\forall x \in \Omega) \quad x \not\asymp \theta, \\ (\forall x \in \Omega) \quad x \not\prec \theta, \\ (\forall x \in \Omega) \quad x \neq \theta. \end{cases} \quad (11)$$

The element θ is a special element, e.g. the imaginary unit i satisfies (11) when $\Omega = [0; 1]$. The aim of this value is to indicate that for considered trait and vertex the trait's value does not exist (it is undefined). Let us define the function $T_k : \Omega^{m_k} \rightarrow (\Omega^*)^{|\Pi|}$:

$$T_k \begin{pmatrix} p_{k,1} \\ p_{k,2} \\ \vdots \\ p_{k,m_k} \end{pmatrix} \stackrel{\text{def}}{=} \begin{pmatrix} r_{k,1} \\ r_{k,2} \\ \vdots \\ r_{k,|\Pi|} \end{pmatrix}, \quad (12)$$

TABLE I
EXAMPLES OF TYPICAL NORMALIZATION FUNCTIONS, BASED ON [27]

Λ	Parameter family	The worst value	The best value	Typical value	Function $g(x)$
\mathbb{R}	raising/falling	0	$\pm\infty$	β	$g(x) = 1 - 2^{-\frac{x}{\beta}}$
\mathbb{R}	falling/rising	$\pm\infty$	0	β	$g(x) = 2^{-\frac{x}{\beta}}$
\mathbb{R}	raising	$-\infty$	$+\infty$	$\beta; \beta \neq 0$	$g(x) = \left(1 + e^{\frac{\beta-x}{ \beta }}\right)^{-1}$
\mathbb{R}	raising	$-\infty$	$+\infty$	0	$g(x) = (1 + e^{-x})^{-1}$
\mathbb{R}	falling	$+\infty$	$-\infty$	$\beta; \beta \neq 0$	$g(x) = \left(1 + e^{\frac{x-\beta}{ \beta }}\right)^{-1}$
\mathbb{R}	falling	$-\infty$	$+\infty$	0	$g(x) = (1 + e^x)^{-1}$
$[q; r] \in \mathbb{R}_{\geq 0}$	raising	q	r	$\frac{q+r}{2}$	$g(x) = \frac{x-q}{r-q}$
$[q; r] \in \mathbb{R}_{\geq 0}$	falling	r	q	$\frac{q+r}{2}$	$g(x) = \frac{r-x}{r-q}$
\mathbb{R}	Gaussian	$\pm\infty$	γ	β	$g(x) = 2^{-\frac{x-\gamma}{ \beta-\gamma }}$

where

$$r_{k,j} = \begin{cases} \text{value of } \pi_j \text{ for } V_k, & \text{when } \pi_j \in \Pi_k \\ \theta, & \text{otherwise.} \end{cases} \quad (13)$$

Now, one can define a general merging function F^* for non-similar vertices as:

$$F^* : (\Omega^*)^{n|\Pi|} \rightarrow (\Omega^*)^{|\Pi|}. \quad (14)$$

According to equation (3), we can write:

$$F^*(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n) = \begin{pmatrix} f_1^*(r_{1,1}, r_{2,1}, \dots, r_{n,1}) \\ f_2^*(r_{1,2}, r_{2,2}, \dots, r_{n,2}) \\ \vdots \\ f_{|\Pi|}^*(r_{1,|\Pi|}, r_{2,|\Pi|}, \dots, r_{n,|\Pi|}) \end{pmatrix}, \quad (15)$$

where the function $f^* : (\Omega^*)^n \rightarrow \Omega^*$ is a non-similar vertex's version of the merging function f . According to equations (4,5), the following assumptions upon the f^* could be made:

$$(\forall x \in \Omega^*) \quad f^*(x, x, \dots, x) = x, \quad (16)$$

$$(\forall 1 \leq k \leq n)(\forall x_k \leq y_k) \quad (17)$$

$$f^*(x_1, x_2, \dots, x_k, \dots, x_n) \leq f^*(x_1, x_2, \dots, y_k, \dots, x_n),$$

$$f^*(x_1, x_2, \dots, x_n) = \theta \iff x_1 = x_2 = \dots = x_n = \theta. \quad (18)$$

The equation (17) requires that the values x_k and y_k must be comparable. The θ element does not satisfy this condition for any other value from the Ω^* set, according to the (11). Consequently, when one defines trait's value for an already existing vertex, the merged trait's value with isolation merging function could be higher, lower or stay at the same point.

The $f^*(x_1, x_2, \dots, x_n)$ function could be calculated in the following way: let the $Z = (z_1, z_2, \dots, z_{n'})$ be a string of elements from $X = (x_1, x_2, \dots, x_n)$ created by selecting all elements except the elements equal to θ . Then, we can use e.g.

(9) formula for calculations, using only values from Z . The weights should be the same for the element $z_k; 1 \leq k \leq n'$ and the corresponding element from X .

3) *Choosing the merging function:* The following aspects should be considered for choosing an appropriate merging function.

- Interpretation of the merged trait, e.g. available throughout for a path of vertices is upper-bounded by a minimal value, so the merging function could be defined as $f(x_1, x_2, \dots, x_n) = \min\{x_1, x_2, \dots, x_n\}$.
- Implementation constraints: the integer-valued weights and function's parameters could result in more accurate and faster calculations; the operations upon integers are faster than on typical IEEE 754 [29] double precision numbers. The *money-like* types which allow exact operations are slower than hardware supported types.
- Precision constraints: each operation on non-integer number suffers from the finite precision problem, which causes losing the information on the less-important part of a number. Using large number of operations (multiplication, adding, power) leads to very uncertain results.
- The expected value of a merging function: the normalization function defines the central element $C = g(\beta)$ of Ω (i.e., 0.5 for the set $\Omega = [0, 1]$) as a typical value which should be close to the expected value of the merging function. This value could depend on the number of merged properties and the function's internal parameters, like the parameter q for the generalized mean. The Figure 3 shows how the mean value changes for the generalized mean function. Only the arithmetical mean ($q = 1$) from this functions family satisfies, for all $n > 1$, the equation:

$$\mathbb{E}(f_q(x_1, x_2, \dots, x_n)) = g(\beta). \quad (19)$$

- The shape (e.g. convexness) of a merging function: traits are interpreted with some logic, e.g.:

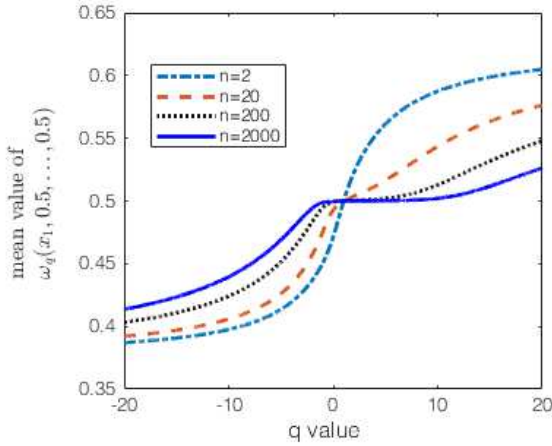


Fig. 3. Mean value of ω_q function for various number of traits n and the parameter q .

- *The OR logic*: the merged trait's value is more strongly dependent on large values than on small ones. This kind of traits should be merged with convex functions. An example trait is *number of CVE vulnerabilities* for a network's node.
- *The AND logic*: the merged trait's value is more strongly dependent on small values than on large ones. This kind of traits should be merged with concave functions. An example trait is *encryption strength* in number of key's bits for a network's node.
- *The neutral logic*: the merged trait's value depends on the small and large values equally. This kind of traits should be merged with function which satisfies the following equation:

$$\frac{\partial^2}{\partial x_k^2} f(x_1, x_2, \dots, x_k, \dots, x_n) = 0. \quad (20)$$

B. Calculating the isolation vector

The isolation vector for a network could be calculated recursively by calculating isolation vectors for subnetworks and a group of vertices from a layer. This *bottom-up* method allows adding special information, which should be included into the calculation process, and which cannot be defined for vertices in lower layers.

C. The comparison problem for vectors

From practical point of view, it is very important to compare calculated isolation vectors which are defined over a common set of traits. Two vectors, which have only one different trait's value, are easy to compare. When multiple traits' values are different, the situation is much more complicated. In mathematical terms the aim is to define the linear order for the set $(\Omega)^{|\Omega|}$; the θ value could be omitted, because each trait should be defined for at least one vertex in a graph, so the final vector has all non- θ values. Such an order could be defined by assigning to each of isolation vectors a number, which can be compared. This assignment is provided by the

extracting function $\Phi : (\Omega)^{|\Omega|} \rightarrow \Omega$. This function should satisfy following inequalities for each $x_k \in \Omega$:

$$\frac{\partial}{\partial x_k} \Phi(x_1, x_2, \dots, x_k, \dots, x_n) \geq 0, \quad (21)$$

$$\frac{\partial^2}{\partial x_k^2} \Phi(x_1, x_2, \dots, x_k, \dots, x_n) \leq 0. \quad (22)$$

The inequality (21) means that the extracting function is raising for each of its parameters. The inequality (22) describes the assumption that the function is concave for each parameter.

D. Extracting the single value

The following example function family, which satisfies the assumptions (21, 22) could be used for extracting the single value for an isolation vector:

$$\Phi_q(x_1, x_2, \dots, x_k, \dots, x_n) = 1 - \left(\frac{1}{n} \sum_{i=1}^n (1 - x_i)^q \right)^{\frac{1}{q}}. \quad (23)$$

This function has the following partial derivatives:

$$\begin{aligned} \frac{\partial}{\partial x_k} \Phi_q(x_1, x_2, \dots, x_k, \dots, x_n) &= \\ &= \left(\sum_{i=1}^n (1 - x_i)^q \right)^{\frac{1}{q}-1} \frac{(1 - x_k)^{q-1}}{n} \geq 0, \end{aligned} \quad (24)$$

$$\begin{aligned} \frac{\partial^2}{\partial x_k^2} \Phi_q(x_1, x_2, \dots, x_k, \dots, x_n) &= \\ &= \frac{1-q}{n^2} \left(\sum_{i=1}^n (1 - x_i)^q \right)^{\frac{1}{q}-2} (1 - x_k)^{q-2} \sum_{i=1; i \neq k}^n (1 - x_i)^q. \end{aligned} \quad (25)$$

If $q \geq 1$, then the second derivative calculated in the equation (25) is non-positive, so the assumption defined in equation (22) is satisfied.

V. EXAMPLES

In this section we consider two examples: calculating the isolation of a single node in a network and calculating the isolation over an E2E path for a single slice. The second example contains a list of steps which are included in the isolation assessment process.

A. Example 1. Single node

Let us consider a single network element with traits, traits' values and normalization functions defined in the Table II. The normalization functions are fitted to expected traits' domains, which is the main reason during selecting the normalization function. We assumed for the trait *symmetric encryption algorithm's strength* that at this moment the largest available (and practically used) key size is 256 bits. Theoretically, the key could have any length (e.g. one-time keys for stream ciphers), but very large key size is impractical as well, so in such typical case will not be considered as an option. Since the domain is constrained to the range of integer numbers, the linear normalization function for this trait was chosen.

TABLE II
SET OF NORMALIZATION FUNCTIONS AND NORMALIZED VALUES FOR A SINGLE NODE

Parameter	Value	Normalization function	Normalized value
Symmetric encryption algorithm's strength $\begin{cases} a = 0 \\ b = 256bits \end{cases}$	160 bits	$g(x) = \frac{x-a}{b-a}$	0.625
Average time between vulnerabilities assessments $\beta = 4h$	8h	$g(x) = 2^{-\frac{x}{\beta}}$	0.25
Amount of electromagnetic radiation $\beta = 25dB\mu V/m$	55dB $\mu V/m$	$g(x) = \left(1 + e^{\frac{x-\beta}{ \beta }}\right)^{-1}$	0.2315

The isolation vector for this vertex (which is a representation of this single node scenario) is (0.625, 0.25, 0.2315). We can extract the single value from this vector using the function $\Phi_2(x_1, x_2, x_3)$:

$$I_{final} = 1 - \sqrt{\frac{\sum_{i=1}^3 (1 - x_i)^2}{3}} =$$

$$= 1 - \sqrt{\frac{(1 - 0.625)^2 + (1 - 0.25)^2 + (1 - 0.2315)^2}{3}} =$$

$$= 0.3433. \quad (26)$$

B. Example 2. A simple end-to-end slice

The process of isolation analysis is defined as follows:

- 1) definition of use-case to be modeled;
- 2) definition of all network resources to be modeled (nodes and links);
- 3) definition of each resource affiliation to RAN and/or CN;
- 4) for each resource definition of relevant isolation parameters and properties to be determined or measured;
- 5) for CN, RAN and E2E definition of relevant isolation parameters and properties to be determined or measured;
- 6) choosing a set of functions to normalize isolation parameters and properties;
- 7) choosing a set of functions to calculate isolation from parameters and properties;
- 8) choosing a set of functions to compare two or more isolation tuples;
- 9) definition of a slices spanned across previously defined resources;
- 10) for each layer, calculation of slices' isolation, with a previously chosen set of functions;
- 11) performing comparison, if needed, with a previously chosen set of comparison functions.

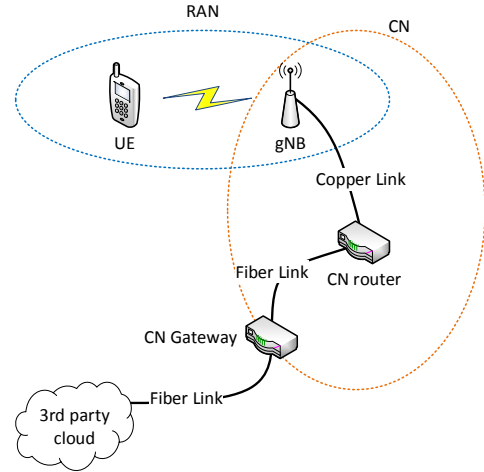


Fig. 4. The example of a network scenario for the purpose of isolation analysis.

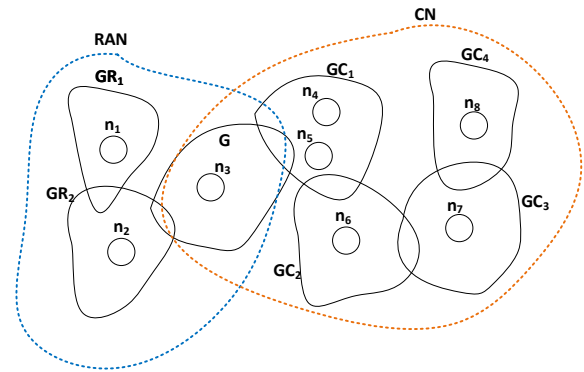


Fig. 5. The representation of the selected piece of network in the Graph model.

As an example, the simple network is analyzed in context of E2E calculation of isolation (see Figure 4). RAN network consists of a single User Equipment (UE), which connects with gNB node using Radio Link. CN consists of three types of equipment: gNB (which is common for RAN and CN), CN Router and CN Gateway. gNB is connected with CN router using Copper Link and CN Router is connected with CN Gateway using Fiber Link. CN Gateway is connected with a 3rd-party-vendor cloud using Fiber Link, which is considered outside of RAN and CN.

The described piece of network can be represented as a graph structure using the Graph model (see Figure 5)

RAN and CN are the hypergraphs which include sub-hypergraphs: GR_x , GC_x and G (GR_x - RAN sub-hypergraph of the element x , GC_x - CN sub-hypergraph of the element x , G sub-hypergraph of the gateway). Each sub-hypergraph has several nodes (n_i) which present the properties. Every property can be virtualized: v_1n_1 , v_2n_1 , v_3n_1 , etc. (v_in_j means a virtual property "i" of a physical property "j").

According to the process of E2E isolation analysis, each step may be described as follows:

- 1) **Use case definition:** in a network depicted in the Figure 4 a slice is being configured from UE to CN gateway. There is a need to assess isolation level of the slice based on the equipment and media used to serve this slice. There are two types of routers in CN which can be chosen in the network's implementation (it is assumed that each of them has assessed the same chosen parameters and properties). There is a need to assess the isolation of the slice while the first and the second types of routers are chosen. The isolation in these two cases should be compared to choose a solution with better isolation for a slice. The analysis should be performed only on Layers 1, 2 and 3 of the model.
- 2) **Network resources to be modeled:** in the analysis there should be modeled the following resources:
 - User Equipment in RAN;
 - Radio Link in RAN;
 - gNB as a gateway between RAN and CN (affiliated to RAN and CN);
 - Copper Link in CN;
 - CN Router in CN;
 - Fiber Link in CN;
 - CN Gateway as a gateway between CN and 3rd-party-cloud (affiliated to CN).

Figure 5 represents the network for the isolation analysis in a graph form.

- 3) **RAN-CN affiliation:** the affiliation was described in the previous point.
- 4) **Parameters and properties for elements:** the following isolation parameters and properties were identified for each resource:
 - User Equipment in RAN (GR_1):
 - slicing application: programming language used (e.g., Java), enumeration (n_1).
 - Radio Link in RAN (GR_2):
 - symmetric encryption algorithm strength: the number of bits, a nonnegative integer (n_2).
 - gNB as a gateway between RAN and CN (G):
 - average time between vulnerabilities assessments: hours, a nonnegative real number (n_3).
 - Copper Link in CN (GC_1):
 - symmetric encryption algorithm strength: the number of bits, a nonnegative integer (n_4);
 - amount of electromagnetic radiation: i.e., dBV/m, a real number (n_5).
 - CN Router in CN (GC_2):
 - average time between vulnerabilities assessments: hours, a nonnegative real number (n_6).
 - Fiber Link in CN (GC_3):
 - symmetric encryption algorithm strength: the number of bits, a nonnegative integer (n_7).
 - CN Gateway as a gateway between CN and 3rd-party-cloud (GC_4):

- average time between vulnerabilities assessments: hours, a nonnegative real number (n_8).
- 5) **Parameters and properties for CN, RAN and E2E:** the following isolation parameters and properties were identified:
 - Core Network (CN):
 - symmetric encryption algorithm strength;
 - amount of electromagnetic radiation;
 - average time between vulnerabilities assessment.
 - Radio Access Network (RAN):
 - symmetric encryption algorithm strength;
 - programming language used;
 - average time between vulnerabilities assessment.
 - End-to-End (E2E):
 - symmetric encryption algorithm strength;
 - programming language used;
 - amount of electromagnetic radiation;
 - average time between vulnerabilities assessment;
 - (produced by extracting the single value from other traits) isolation level.
 - 6) **Choose a set of functions to normalize isolation parameters and properties:** These functions are defined in the Table III.
 - 7) **Choose a set of functions to calculate isolation from parameters and properties:** We will use as *the isolation-merging function* the following formula:

$$\omega_{-1}(x_1, x_2, \dots, x_n). \quad (27)$$

for calculating the isolation inside the RAN or CN part of the network and the weighted version of this mean for calculating isolation E2E for these two parts of the network. The value $q = -1$ is used, because it is very fast to implement, it is a merging function with the *AND logic*. The Figure 3 shows that this function has the mean value very close to 0.5 (the central element of the Ω set), which is advisable.

We use the weighted function for merging isolation between CN and RAN for flattening an impact of each graph's vertex in results. The weights are defined in the column *Weights* in the Table V. Since the gNB node (and in consequence its vertex in the Graph model) belongs to RAN and CN as well, its impact is doubled. To avoid this excessive influence, the subnetworks on the path should be separated.

- 8) **Choose a set of functions to compare two or more isolation tuples:** We will use the proposed method for calculating the final isolation and the calculated final isolations (the I_{final} values) will be used for comparisons of the slices' isolation levels.
- 9) **Define a slice spanned across previously defined resources:** We assume in this scenario that the slice is from UE to the 3rd-party-cloud and contains all devices and links between these nodes.
- 10) **For each layer, calculate slices isolation with a previously chosen set of functions:** The results of

TABLE III
SET OF NORMALIZATION FUNCTIONS AND NORMALIZED VALUES

Part	Element	Parameter	Value	Normalization function	Normalized value	Typical values / margin values
RAN	UE in RAN	Programming language used	C++ (0.75)	N/A	0.75	N/A
	Radio Link in RAN	Symmetric encryption algorithm strength	160 bits	$g(x) = \frac{x-a}{b-a}$	0.625	$\begin{cases} a = 0 \\ b = 256bits \end{cases}$
gNB	gNB as a gateway between RAN and CN	Average time between vulnerabilities assessments	8h	$g(x) = 2^{-\frac{x}{\beta}}$	0.25	$\beta = 4h$
CN	Copper Link in CN	Symmetric encryption algorithm strength	128 bits	$g(x) = \frac{x-a}{b-a}$	0.5	$\begin{cases} a = 0 \\ b = 256bits \end{cases}$
		Amount of electromagnetic radiation	55dBμV/m	$g(x) = \left(1 + e^{\frac{x-\beta}{ \beta }}\right)^{-1}$	0.2315	$\beta = 25dB\mu V/m$
	CN Router in CN	Average time between vulnerabilities assessments	12h	$g(x) = 2^{-\frac{x}{\beta}}$	0.125	$\beta = 4h$
	Fiber Link in CN	Symmetric encryption algorithm strength	256 bits	$g(x) = \frac{x-a}{b-a}$	1	$\begin{cases} a = 0 \\ b = 256bits \end{cases}$
	CN Gateway as a gateway between CN and 3rd party cloud	Average time between vulnerabilities assessments	4h	$g(x) = 2^{-\frac{x}{\beta}}$	0.5	$\beta = 4h$

TABLE IV
ISOLATION INSIDE RAN AND CN

Part	Parameter	Values	Merged values
RAN	Average time between vulnerabilities assessments	0.25	0.25
	Encryption algorithm strength	0.625	0.625
	Programming language used	0.75	0.75
CN	Encryption algorithm strength	0.5, 1	$\frac{2}{\frac{1}{0.5} + \frac{1}{1}} = 0.6667$
	Amount of electromagnetic radiation	0.2315	0.2315
	Average time between vulnerabilities assessments	0.25, 0.125, 0.5	$\frac{3}{\frac{1}{0.25} + \frac{1}{0.125} + \frac{1}{0.5}} = 0.2143$

calculations are in Tables IV and V. From the obtained results presented in the Table V, we can build the *isolation vector* (0.75, 0.6522, 0.2222, 0.2315) for this example network. One can *extract the single value* from this vector using the function $\Phi_2(x_1, x_2, x_3, x_4)$:

$$I_{final} = 1 - \sqrt{\frac{\sum_{i=1}^4 (1 - x_i)^2}{4}} = 0.4128. \quad (28)$$

- 11) **Perform comparison, if needed, with a previously chosen set of comparison functions:** The defined slice has I_{final} in a medium level and it could be improved. From the Table V we can choose the worst isolation trait (by its value): **Average time between vulnerabilities assessments.** In this scenario this trait is merged from

TABLE V
ISOLATION VECTOR VALUES

Parameter	Values	Weights	Isolation vector's values
Programming language used	0.75	1	0.75
Encryption algorithm strength	0.625, 0.6667	1/3, 2/3	$\frac{1}{\frac{1}{0.625} + \frac{2}{0.6667}} = 0.6522$
Average time between vulnerabilities assessments	0.25, 0.2143	1/4, 3/4	$\frac{1}{\frac{1}{0.25} + \frac{3}{0.2143}} = 0.2222$
Amount of electromagnetic radiation	0.2315	1	0.2315

RAN and CN with different weights and with different values. The vertex to improve could be determined by exhaustive search where all traits, except current vertex's traits, has origin values and the current vertex's trait's value is set to 1 (the ω value). The result of this search is summarized in the Table VI. The *CN Router in CN*'s trait's value should be improved, because enhancement of this element could make the biggest effort on the I_{final} value. Let us assume now, that we improved this trait's value to 4h (0.5 after normalization). After this operation, the value of this trait in the isolation vector is 0.3333 and the $I_{final} = 0.4481$.

C. Discussion of the results

The I_{final} values are below the 0.5, which should be expected, because the traits' values in both examples are generally low or medium. In this scenario the trait's values are higher than 0, so the ω_{-1} function could be used for merging isolation. If zero values for the traits are expected, the parameter q should be greater than 0. Such a situation

TABLE VI

SEARCHING FOR THE BEST VERTEX TO IMPROVE THE WORST TRAIT: THE RESULTS

Vertex	Origin trait's value	Max. available merged value in RAN	Max. available merged value in CN	Max. trait's value in the isolation vector
G - gNB as a gateway between RAN and CN	0.25	1	0.2727	0.3333
GC ₂ - CN Router in CN	0.125	0.25	0.4286	0.3636
GC ₄ - CN Gateway as a gateway between CN and 3rd party cloud	0.5	0.25	0.2308	0.2353

could happen if linear functions are used for normalization or one of the properties' values is zero.

A change in the Step 11 for the Example 2 is small, because in the isolation vector there exists a parameter with the value 0.2315, which has a significant impact on the I_{final} and which should be improved.

The procedure for finding the best vertex and its trait to improve the isolation is defined in a heuristic way. There is a space for further improvements and research. In this case we assumed that each trait has the same merging function, but it could be trait-dependent, and it should be considered by the algorithm executed in the Step 11. This algorithm also could consider the cost of each trait's improvement in a vertex.

VI. CONCLUSIONS AND FUTURE WORK

In this paper the Graph-based quantitative description of networks' slices isolation has been proposed. We have presented a model which can be used to represent in a transparent way a slice in 5G Network, allowing its detailed analysis and supporting additional calculations. As an example of such calculations we have considered the problem of estimating the isolation level of the end-to-end slice and also isolation level of each network's element, both, physical and virtual. Moreover, we have proposed a general framework and the mathematical rules defining how the isolation of that slice can be calculated. Finally, in the paper we have included examples of isolation calculation for a single node and for the end-to-end scenario of a single slice. The numerical results proved to be promising, indicating possibilities of application of our approach in slices management and optimization.

The paper presents a research on its initial state. In our opinion the future work on the presented topic should be continued. Among others, it should include the following issues:

- Validation and verification of the presented Graph model by modeling different scenarios;
- Development of methods for comparing different types of slices;

- Proof of concept application development;
- Development of a set of parameters and properties for devices and links;
- Development of better algorithm for selecting a trait and graph's vertex to improve the isolation;
- Research of integration of the presented model with 5G MANO systems.

REFERENCES

- [1] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A blueprint for introducing disruptive technology into the Internet," *ACM SIGCOMM Computer Communication Review*, vol.33, iss.1, pp.59-64, 2003, <https://doi.org/10.1145/774763.774772>.
- [2] C. Chapman, S. Ward, "Description of Network Slicing Concept", NGMN Alliance, January, 2016, <https://www.ngmn.org/publications/all-downloads/article/description-of-network-slicing-concept.html>.
- [3] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A. A. Gebremariam, T. Taleb, and M. Bagaa, "End-to-end Network Slicing for 5G Mobile Networks," *Journal of Information Processing*, vol.25, pp.153-163, Feb. 2017, <https://doi.org/10.2197/ipsjip.25.153>.
- [4] Nokia "Dynamic end-to-end network slicing for 5G," (White Paper), 2016.
- [5] T. Shimojo, Y. Takano, A. Khan, S. Kaptchouang, M. Tamura, and S. Iwashina, "Future mobile core network for efficient service operation," in *Proceedings of the 2015 1st IEEE Conference on Network Softwareization (NetSoft)*, London, 2015, pp.1-6, <https://doi.org/10.1109/NETSOFT.2015.7116190>.
- [6] U. Herzog, A. Georgakopoulos, I.-P. Belikaidis, M. Fitch, K. Briggs, S. Diaz, O. Carrasco, K. Moessner, B. Miscopein, S. Mumtaz, and P. Demestichas, "Quality of service provision and capacity expansion through extended-DSA for 5G," *Transaction of Emerging Telecommunications Technologies*, vol.27, iss.9, pp.1250-1261, September 2016, <https://doi.org/10.1002/ett.3061>.
- [7] Z. Kotulski, T. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, T. Osko, J.-P. Wary, "On end-to-end approach for slice isolation in 5G networks. Fundamental challenges", *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, in: M. Ganzha, L. Maciaszek, M. Paprzycki (Eds.). ACSIS, vol.11, pp.783-792, 2017, <https://doi.org/10.15439/2017F228>.
- [8] Z. Kotulski, T. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, T. Osko, J.-P. Wary, "Towards constructive approach to end-to-end slice isolation in 5G networks", *EURASIP Journal of Information Security*, vol. 2018:2, pp.1-16, 2018, <https://doi.org/10.1186/s13635-018-0072-0>.
- [9] D. Schlosser, M. Hoffmann, T. Hoßfeld, M. Jarschel, A. Kirstaedter, W. Kellerer, S. Kohler, "COMCON: Use Cases for Virtual Future Networks", in: T. Magedanz et al. (Eds.): *TridentCom 2010*, LNICST 46, pp.584-586, 2011, https://doi.org/10.1007/978-3-642-17851-1_48.
- [10] J. White, G. Jourjon, T. Rakatoarivelo, M. Ott, "Measurement Architectures for Network Experiments with Disconnected Mobile Nodes", in: T. Magedanz et al. (Eds.): *TridentCom 2010*, LNICST 46, pp.315-330, 2011, https://doi.org/10.1007/978-3-642-17851-1_26.
- [11] A.-Ch. Anadiotis, A. Apostolaras, D. Syrivelis, T. Korakis, L. Tassioulas, L. Rodriguez, I. Seskar, M. Ott, "Towards Maximizing Wireless Testbed Utilization Using Spectrum Slicing", in: T. Magedanz et al. (Eds.): *TridentCom 2010*, LNICST 46, pp.299-314, 2011, https://doi.org/10.1007/978-3-642-17851-1_25.
- [12] G. Bhanage, I. Seskar, D. Raychaudhuri, "A Service Oriented Experimentation Framework for Virtualized WiMAX Systems", in: T. Korakis et al. (Eds.): *TridentCom 2011*, LNICST 90, pp.152-161, 2012, https://doi.org/10.1007/978-3-642-29273-6_12.
- [13] G. Bhanage, I. Seskar, Y. Zhang, D. Raychaudhuri, S. Jain, "Experimental Evaluation of OpenVZ from a Testbed Deployment Perspective", in: T. Magedanz et al. (Eds.): *TridentCom 2010*, LNICST 46, pp.103-112, 2011, https://doi.org/10.1007/978-3-642-17851-1_7.
- [14] S. Soltész, H. Potzl, M.E. Ficzynski, A. Bavier, L. Peterson, "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors", *Proceeding of EuroSys'07 Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems*, pp.275-287, 2007. <https://doi.org/10.1145/1272996.1273025>.

- [15] C. Zhao, Y. Wu, Z. Ren, W. Shi, Y. Ren, J. Wan, "Quantifying the Isolation Characteristics in Container Environments". in: X. Shi et al. (Eds.): *Network and Parallel Computing. NPC 2017*. Lecture Notes in Computer Science, vol.10578, Springer, https://doi.org/10.1007/978-3-319-68210-5_17.
- [16] Sh. Ma, J. Jiang, B. Li, B. Li, "Maximizing Container-based Network Isolation in Parallel Computing Clusters", *IEEE 24th International Conference on Network Protocols (ICNP)*, 2016, <https://doi.org/10.1109/ICNP.2016.7784434>.
- [17] M. Chen, A. Nakao, "Feather-Weight Network Namespace Isolation Based on User-Specific Addressing and Routing in Commodity OS", in: T. Magedanz et al. (Eds.): *TridentCom 2010*, LNICST 46, pp.53-68, 2011, https://doi.org/10.1007/978-3-642-17851-1_4.
- [18] P. Du, M. Chen, A. Nakao, "Port-Space Isolation for Multiplexing a Single IP Address through Open vSwitch", in: T. Magedanz et al. (Eds.): *TridentCom 2010*, LNICST 46, pp.113-122, 2011, https://doi.org/10.1007/978-3-642-17851-1_8.
- [19] P. Du, M. Chen, A. Nakao, "OFIAS: A Platform for Exploring In-Network Processing", in: T. Korakis et al. (Eds.): *TridentCom 2011*, LNICST 90, pp.142-151, 2012, https://doi.org/10.1007/978-3-642-29273-6_11.
- [20] Sh. Ma, B. Wang, X. Zhang, T. Li, "An Evolving Architecture for Network Virtualization", in: V.C.M. Leung et al. (Eds.): *TridentCom 2014*, LNICST 137, pp.379-386, 2014. , https://doi.org/10.1007/978-3-319-13326-3_36.
- [21] CONFINE Project, Community Networks Testbed for the Future Internet, <http://confine-project.eu/>
- [22] S. Gutz, A. Story, C. Schlesinger, N. Foster, "Splendid Isolation: A Slice Abstraction for Software-Defined Networks", *HotSDN'12*, August 13, 2012, Helsinki, Finland, <https://doi.org/10.1145/2342441.2342458>.
- [23] Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferrero, Timoteo Lange, Cesar A. F. De Rose, "Performance Evaluation of Container-based Virtualization for High Performance Computing Environments", *21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2013, <https://doi.org/10.1109/PDP.2013.41>.
- [24] "Isolation Benchmark Suite", 2012. [Online]. Available: <http://web2.clarkson.edu/class/cs644/isolation>.
- [25] J.N. Matthews, W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, M. McCabe, J. Owens, "Quantifying the Performance Isolation Properties of Virtualization Systems", *ExpCS'07*, 13-14 June, 2007, San Diego, CA, <https://doi.org/10.1145/1281700.1281706>.
- [26] D. Sattar, A. Matrawy, "Optimal Slice Allocation in 5G Core Networks", *arXiv:1802.04655 [cs.NI]*, 2018.
- [27] Z. Kotulski, T. Nowak, M. Sepeżuk, M. Tunia, "5G networks: types of isolation and their parameters in RAN and CN slices", submitted.
- [28] P.P Korovkin, *Inequalities*. Moscow : Mir Publishers, 1975.
- [29] IEEE. *IEEE Standard for Floating-Point* - IEEE Std 754-2008. 2008.