This is a repository copy of *Defining and Surveying Wireless Link Virtualization and Wireless Network Virtualization*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/158266/

Defining and Surveying Wireless Link and Network Virtualization

Jonathan van de Belt*, Hamed Ahmadi†, and Linda E. Doyle*

*The Centre for Future Networks and Communications - CONNECT, Trinity College Dublin

Email: vandebej@tcd.ie

†School of Electrical and Electronic Engineering, University College Dublin, Ireland

Abstract—Virtualization is a topic of great interest in the area of mobile and wireless communication systems. However the term virtualization is used in an inexact manner which makes it difficult to compare and contrast work that has been carried out to date. The purpose of this paper is twofold. In the first place, the paper develops a formal theory for defining virtualization. In the second instance, this theory is used as a way of surveying a body of work in the field of wireless link virtualization, a subspace of wireless network virtualization. The formal theory provides a means for distinguishing work that should be classed as resource allocation as distinct from virtualization. It also facilitates a further classification of the representation level at which the virtualization occurs, which makes comparison of work more meaningful. The paper provides a comprehensive survey and highlights gaps in the research that make for fruitful future work.
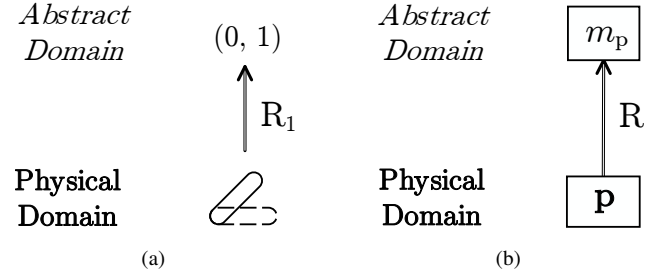
Figure 1. Objects in the physical domain can be *represented* using objects in the abstract domain, such as (a) a switch with two settings represented as a bit through R_1, or (b) more generally as an object **p** represented as $m_\mathbf{p}$ through R.

I. INTRODUCTION

Network Virtualization (NV) allows network services to view network resources, such as servers, routers, links, and data, in a manner that is independent from the underlying physical infrastructure, and to use these resources according to service requirements, rather than based on physical granularities [1]. New network functionality can be achieved using virtualization, such as providing heterogeneous networks with customizable specifications on-demand, the flexible and dynamic management of resources, new types of services, and better security and protection against equipment failure. In addition, network virtualization has the potential to enable new networking technologies and protocols to be developed much faster than they are currently, since these technologies can be tested through isolated virtual networks on existing infrastructure, while ensuring that existing services are unaffected. Lastly, network virtualization can provide cost savings and new business opportunities, through increased efficiencies and through new services and functionality.

Wireless Network Virtualization (WNV) has been proposed as an extension of (wired) network virtualization to the wireless domain, with the main difference being the wireless links. Thus most work to date has focussed on Wireless Link Virtualization (WLV). Initially, the purpose of this paper is to perform a survey of WNV, and to identify open research problems. However, the term 'wireless network virtualization' carries multiple connotations. It has become an umbrella term for several differing concepts, applied at different layers of the network stack, and also to different types of network resources. In the existing literature, works such as [2]–[7] provide a variety of definitions for WNV, but a lack of consistency

persists. A common theme of these definitions is that they regard virtualization as the abstraction and sharing/slicing of resources. As this work will emphasise, virtualization and abstraction are very different concepts, and virtualization is not necessarily limited to the sharing of resources. It is interesting to note that several authors ([8] and [4]) have pointed out a similar vagueness and lack of clarity in the field of network virtualization.

Although several surveys on wireless network virtualization exist such as [5], [6], [9]–[11], this paper brings additional and alternative perspectives (for a more detailed description see Section VI-G). We develop a formal method for describing virtualization as a response to the many definitions that currently exist. We then use this formal method as a means of classifying and analysing the papers we survey which allows for a more systematic approach to the survey process.

More specifically we make the following contributions:

1) We clarify the concepts of abstraction and representation, which are key to understanding virtualization, by drawing on a theory known as abstraction/representation theory and extending it.

2) We propose a formal method for describing virtualization, which we call virtualization theory.

3) We develop a test for virtualization to distinguish virtualization techniques from resource allocation techniques.

4) We survey the existing work on wireless network virtualization, and classify this work in a coherent and meaningful manner, using virtualization theory.

5) We identify several research gaps in wireless network virtualization which have not yet been addressed and propose next steps forward.

The paper is structured around these contributions. Section II introduces key concepts such as abstraction, representation and instantiation, which are prerequisite to virtualization and important for the rest of the paper. Section III introduces the theory on which the paper is grounded. In Section IV we examine the constituent components of networks, and how these components can be virtualized. We give an overview of network and wireless network virtualization in Section V, to introduce the survey. Since the main focus of this paper is on wireless link virtualization, in Section VI we perform a survey of existing literature on wireless link virtualization. This survey allows us to identify open research directions in Section VII, before concluding in Section VIII.

II. Prerequisites to Virtualization

There are a number of concepts, which are key to developing a formal theory of virtualization and in the opening section of the paper we carefully define these concepts. The first two of these are the concepts of abstraction and representation. Table I shows an explanation of these and other terms used in this paper.

In this paper, the term '*abstraction*' means the act of ignoring or hiding details to consider general characteristics, rather than concrete realities. Thus abstraction manages the way in which systems interact, and the complexity of the interaction, by hiding details that are not relevant to the interaction. Increased abstraction allows systems to be used more easily for specific applications, but this comes at the cost of decreased flexibility and customization. Although abstraction is an important concept in computing, as it governs the interaction between humans and computers, it is not necessarily of importance to virtualization. However, it is important to note the difference between abstraction and the adjective 'abstract'. The term 'abstract' refers to ideas and concepts that do not have physical existence.

The term '*representation*' means to describe or symbolize something in a particular way. The term '*instantiation*' means the implementation or realization of a concept or idea. Representation and instantiation are very important to computing as they describe the relationship between abstract entities and the physical world. As this paper will show later, representation and instantiation are of great importance to virtualization, since virtualization can only be done in the abstract domain, while network resources exist in the physical domain.

We now turn to the recently developed Abstraction/Representation (AR) theory, to provide us with a formal framework of abstraction and representation [12], [13]. Because this theory is very new, we explain the most important aspects here, borrowed from [12], before we go on to extend the ideas for the purposes of this paper.

A. Abstraction/Representation Theory [1]

Abstraction/Representation theory [2] is concerned with the physical domain and the abstract domain (also known as the

---

[1]Based on [12]

[2]AR Theory might be better named as Representation Theory, since it deals mostly with representation, but that name has already been taken.

Table I
EXPLANATION OF TERMS USED IN THIS PAPER

Term	Explanation
Abstraction	The act of ignoring details to consider general characteristics
Abstract	Existing as a thought, idea, or concept, but without physical existence
Representation	The act of symbolising or portraying something in a particular way
Instantiation	Creating a concrete realization of a concept or idea

logical domain), and the relationship between these domains. The physical domain, $\mathbf{P}$, is defined as consisting of all physical objects, $\mathbf{p} \in \mathbf{P}$. The abstract domain, M, consists of all abstract objects, $m \in M$. For instance, a *computer* is an object in the physical domain, which can be in different physical states, while a *computation* is a set of objects and relations in the abstract domain. Bold font is used to indicate an object in the physical domain; italic font for an object in the abstract domain.

1) Representation: A physical object can be represented in the abstract domain, through a representational relationship, R. For example, a physical on-off switch can be represented in the abstract domain by a binary digit, shown in Figure 1. The general representation relation between a physical object, $\mathbf{p}$, and an abstract object, $m_{\mathbf{p}}$, is through a directed map $R : \mathbf{p} \rightarrow m_{\mathbf{p}}$. The abstract object, $m_{\mathbf{p}}$ is said to be an abstract *representation* of the physical object $\mathbf{p}$. It it very important to keep in mind that the representation relation is not a mathematical function or a logical relation, but rather a modelling relation that bridges the divide between the physical and the abstract spaces.

2) Physical and Abstract Evolution: In the abstract domain, there can be an evolution or process, $C : m_{\boldsymbol{p}} \rightarrow m'_{\boldsymbol{p}}$, that changes an abstract object $m_{\mathbf{p}}$ to another abstract object $m'_{\mathbf{p}}$. Similarly in the physical domain, a corresponding evolution $\mathbf{H} : \mathbf{p} \rightarrow \mathbf{p}'$, changes the physical state $\mathbf{p}$ to physical state $\mathbf{p}'$. This physical state, $\mathbf{p}'$ can then be represented as $m_{\mathbf{p}'}$, through the representation relation, R. These concepts are shown in Figure 2 (a).

The two abstract objects, $m_{\mathbf{p}'}$ and $m'_{\mathbf{p}}$, lead us to a key concept in AR theory. If $\left| m_{\mathbf{p}'} - m'_{\mathbf{p}} \right| \leq \epsilon$, for some error ϵ and norm $||$, then we can say that the abstract evolution, *C*, and the corresponding physical evolution, $\mathbf{H}$, *commute*. Under the above condition, the two representation relationships, R: $\mathbf{p} \rightarrow m_{\mathbf{p}}$ and R: $\mathbf{p}' \rightarrow m_{\mathbf{p}'}$, and the pair of abstract and physical evolutions $C : m_{\boldsymbol{p}} \rightarrow m'_{\boldsymbol{p}}$ and $\mathbf{H} : \mathbf{p} \rightarrow \mathbf{p}'$, can be said to form a *commuting diagram*. When a set of abstract and physical objects form a commuting diagram using the representation, R, then $m_{\mathbf{p}}$ is a *faithful abstract representation* of physical system $\mathbf{p}$ for the evolutions $C(m_{\boldsymbol{p}})$ and $\mathbf{H}(\mathbf{p})$. This means we can be confident that the evolution *C* in the abstract domain corresponds to the evolution $\mathbf{H}$ in the physical domain.

The implication of commuting diagrams is that the abstract representation of the final state of a physical object (i.e. $m'_{\mathbf{p}}/m_{\mathbf{p}'}$) can be found either by following the physical evolution and then representing the output abstractly, or by
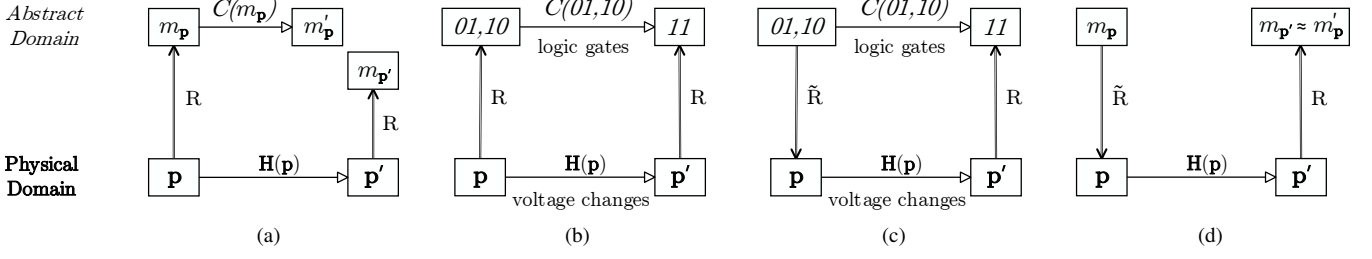
Figure 2. (a) A physical system $\mathbf{p}$ and its representation $m_{\mathbf{p}}$ can undergo abstract evolution, $C(m_p)$, or physical evolution, $\mathbf{H}(\mathbf{p})$. (b) A commuting diagram for a binary addition, showing that an abstract evolution through logic gates, C, is commutative with a physical evolution in voltage states, $\mathbf{H}$. (c) A computer can evaluate an abstract mapping by instantiating the abstract object in the physical domain, performing the physical mapping, and representing it as an abstract object. (d) The most interesting use of a computer is when the abstract mapping is unknown.

theoretically evolving the representation of the physical state. As an example, consider a commuting diagram in which physical voltages are represented by binary numbers, shown in Figure 2 (b). Assume that we want to perform an abstract evolution (binary addition). Then this evolution can be performed either in the abstract domain using logic gates, or through physical manipulation of voltages in the physical domain and representing the result abstractly.

3) Instantiation: The instantiation relationship, $\tilde{\mathrm{R}}$, can be thought of as the inverse to the representation relation. Just as a physical object can be represented in the abstract domain by the representation relation, the instantiation relation, $\tilde{\mathrm{R}}$: $m_{\mathbf{p}} \rightarrow \mathbf{p}$, instantiates an abstract object in the physical domain.

However, unlike representation, the instantiation relation can only exist under specific conditions, as there are many abstract objects that have no physical instantiation. It is necessary for a commuting diagram to exist for a given representation relation, before any attempts can be made to find the inverse instantiation relation (see [12]). Finding an instantiation relation is not straightforward and can be thought of as finding a physical system, that when represented abstractly gives the abstract object that we desire. This often requires trial and error, if such an instantiation is even possible.

Figure 2 (c) shows the binary addition example, but this time the voltages instantiate binary numbers. The instantiation relation, $\tilde{\mathrm{R}}$, can be used to change the physical state to $\mathbf{p}$, so that it instantiates the numbers we wish to add. In the abstract domain, the abstract mapping (i.e. mathematical and logical operations) C, performs the addition to arrive at the result. Meanwhile, the physical mapping, $\mathbf{H}$, manipulates the voltages to produce the physical result. Using the representation relation, R, the abstract representation of $\mathbf{p}'$ is found. If we have confidence that the representation is a faithful abstract representation and also that the instantiation relation is correct, then the outcome of the abstract and the physical evolutions should be the same.

4) Compute Cycle: The previous example describes a computer performing a parallel operation in the abstract and physical domains. However, the most interesting use of a computer is when the abstract mapping C is unknown and we can use the computer to solve an abstract problem, shown in Figure 2 (d). Provided that we are confident in the capabilities of the computer, we can use the computer to find the solution.

The full compute cycle is as follows:
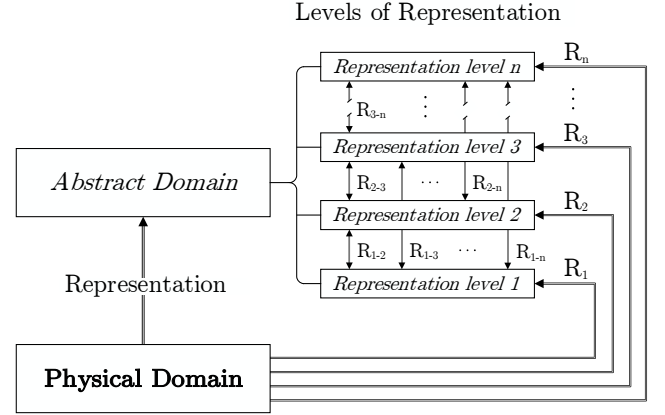
Levels of Representation

Figure 3. The abstract domain can be divided into many levels of representation, since abstract objects can represent other abstract objects. In this case there are n representation levels, ordered arbitrarily, but there can be an infinite number of levels. Each representation level can represent the physical domain directly, through unidirectional representations R_1, R_2, etc., or can represent another representation level, through the bidirectional representations $R_{1\text{-}2}$, $R_{2\text{-}3}$, etc. These representations are bidirectional to show that any representation level can represent another level. This image shows the theoretical representation relations, which does not mean that these relations will all exist in practice.

$$m_{\mathbf{p}} \xrightarrow{\tilde{R}} \mathbf{p} \xrightarrow{\mathbf{H}} \mathbf{p}' \xrightarrow{R} [m_{\mathbf{p}'} \approx m_{\mathbf{p}}']$$

Thus representation and instantiation enable physical computing resources to implement abstract objects and operations, which can be called abstract resources.

B. Levels of Representation

We extend AR theory presented in [12] by focusing on the abstract domain and examining abstract objects in more detail. Most importantly, we observe that objects in the abstract domain, which represent physical objects, can in turn be represented by *different* abstract objects. The same is true for instantiation; objects in the abstract domain, which are instantiated in the physical domain, can instantiate *different* abstract objects. For the purpose of brevity, from here on we only discuss representation and imply that the same is true for instantiation.

In essence there are many different representations that can be used. We use the term 'levels of representation' to capture
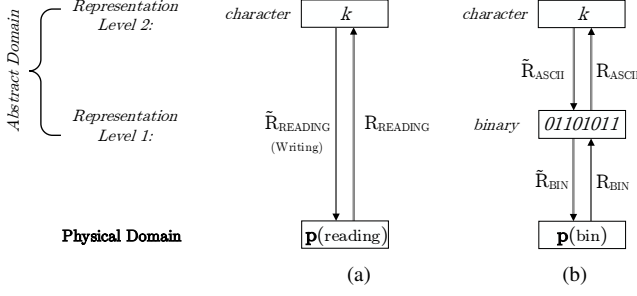
Figure 4. The number of representation levels is a design choice. For example, the character 'k' can (a) directly represent the physical domain, which we commonly know as reading, or (b) represent a binary number, which represents the physical domain. We can speak of one representation level in the first case, the character level, and two representation levels in the second case, the character level and the binary level.
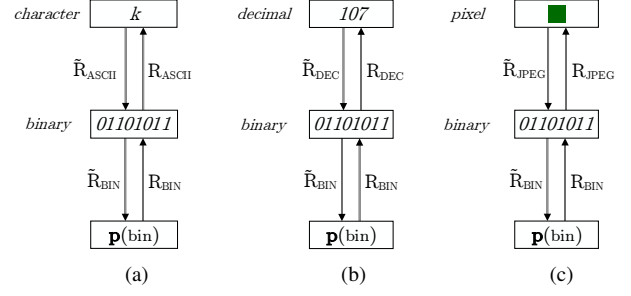

Figure 5. Multiple levels of representation enable a single physical instantiation to be reused for several different abstract objects. In this example, depending on the representation/instantiation, a binary number could instantiate (a) a character, (b) a decimal number, or (c) a pixel in an image.

this idea and this is depicted in Figure 3. In this figure there are n different levels of representation in the abstract domain. We number these representation levels from 1 to n, but the numbering is arbitrary and is only an identifier.

It is important to note that the representation used is an arbitrary design choice, and that it is possible to represent the physical domain at any representation level. The unidirectional arrows, R_1, R_2, etc., show the one-way representation relation from the physical to the abstract domain as seen previously.

Similarly, any representation level can represent another level, shown in the figure through the two-way representation relations R_{1-2}, R_{2-3}, etc.

An important point to observe is that there can be an infinite number of representation levels, as any abstract object can represent another abstract object. However, in practise not every representation level will have a physical representation relation.

A practical example is shown in Figure 4 (a). In this example, a physical object can be represented by the abstract letter 'k' directly, and one representation level exists. However, Figure 4 (b) has two representation levels. In this case the letter 'k' represents a binary number, which in turn represents the physical domain.

C. Choosing a Hierarchy of Representation Levels

In theory, there can be an infinite number of representation levels in the abstract domain, with an arbitrary ordering of representation levels. However, in practise it is more useful to computing (and other abstract domain applications) if an ordered hierarchy of representation levels exists. Then it is possible to think of lower representation levels that are more concrete, and higher representation levels that are more abstract. When levels of representation are used in this manner, then we can number the representation levels in order of increasing abstraction.

One advantage of having an ordered hierarchy of representation levels is that existing physical instantiations for abstract objects can be reused, since it is not an easy task to design a physical instantiation of an abstract system [12]. Rather than finding a physical instantiation for an abstract system,

an instantiation can be found in terms of an abstract system which already has a physical instantiation [3].

Figure 5 shows the advantage of multiple representation levels. In this case, the existing physical instantiation of binary numbers can be used to instantiate additional abstract objects. Many types of data such as integers, characters, volume levels, image brightness, and instructions, can be instantiated in computing using bits [14]. Thus, using multiple levels of representation provides flexibility and easier instantiation. In this example, the binary representation level can be considered more concrete, and the character/decimal/pixel representation level can be considered more abstract.

D. Hardware and Software through Representation

The concept of levels of representation fits in very well with the idea of "hardware" and "software". We define hardware as a compute cycle in which the representation/instantiation occurs between the physical and the abstract domain. Software, in contrast, is defined as a compute cycle in which the representation/instantiation is completely in the abstract domain.

We observe that the concept of representing the physical domain at any representation level (for example in Figure 4.) is consistent with the principle of equivalence of hardware and software, which states:

> Hardware and software are logically equivalent. Any operation performed by software can also be built directly into the hardware and any instruction executed by the hardware can also be simulated in software. [15]

The need for this distinction between hardware and software will be useful later in this paper (for example in section IV-A).

E. How Representation applies to Virtualization

We have examined abstraction and representation, firstly to clarify the distinction between these concepts, and secondly because representation plays a role in virtualization in the following ways:

1) Virtualization must always be done in the abstract domain. The reason for this is that physical resources cannot

[3]Which is not an easy task either, but easier than physical instantiation

be shared or combined without modifying their physical properties in some way. For example, it is not possible to split one processor physically to create multiple virtual processors, however it is possible to split a representation of a processor in the abstract domain. Virtualization allows abstract resources to be reorganised in a manner that is is not limited by the underlying physical resources.

2) Since there can be many different representation levels in the abstract domain, virtualization can be performed at any of these representation levels and virtualization happens *within* a representation level. However, virtualization does not happen across a representation level - that is the act of virtualization does not change abstract resources from one representation level to another representation level. As we will see later, it is important to know the representation level when virtualizing abstract resources.

III. VIRTUALIZATION THEORY

Having gained a better understanding of abstraction and representation, in this section we identify universal concepts of virtualization, and propose a theory of virtualization. Although some of the concepts have been mentioned before in the literature, to the best of our knowledge this is the first time that these concepts are brought together into a unified theory.

A. *Virtualization as Resource Mapping*

As we saw earlier, resources in the abstract domain are representations of the physical domain. To be consistent with the language that is typically used in virtualization, we refer to these resources as *real resources* (RR). Although the term "real" is used, these resources are in the abstract domain, and are not physical resources. Also note that the act of representation is not virtualization, rather representation is a prerequisite for virtualization.

Real resources in the abstract domain can subsequently be virtualized. Virtualization is always performed in the abstract domain and at a specific representation level. Virtualization is a resource mapping which can alter the quantity of resources in some dimension(s). The resources after the virtualization process has occurred are referred to as *virtual resources* (VR), as shown in Figure 6. Virtual resources appear to be the same type of resources as real resources, but can be altered in quantity in some way. This altering allows abstract resources to be used in a flexible manner, not limited by the underlying representation of the physical domain. However, when virtual resources are mapped to real resources, the real resources cannot be used for any other purpose.

Virtual resources are used as if they were real resources, and it should not be possible to perceive any difference between virtual resources and real resources. In this paper we consider that virtual resources are offered to one or multiple users. The term 'users' in this case refers to independent agents, that make decisions on resource use independently. Virtual resources can be offered to different users, and each user has the illusion of full ownership of the resources, meaning that virtual resources can be used for differing purposes.
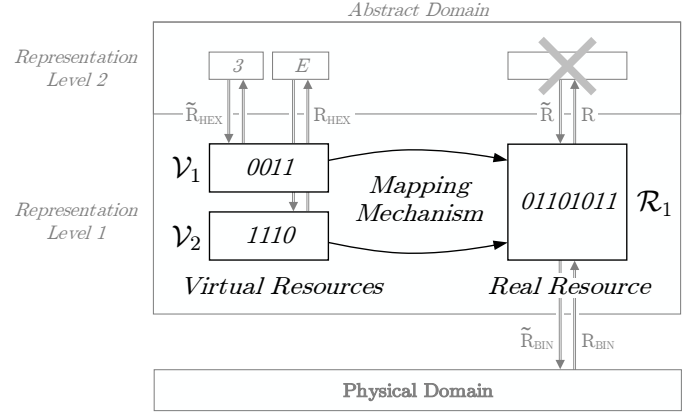

Figure 6. Representation allows physical resources to implement abstract resources, while virtualization allows abstract resources to be owned and used at tailor-made quantities. In this example the 8-bit abstract resource $\mathcal{R}_1$ is not an efficient resource to instantiate a hexadecimal number, as hexadecimal numbers only require 4-bits. Mapping two 4-bit virtual resources, $\mathcal{V}_1$, and $\mathcal{V}_2$, to $\mathcal{R}_1$ using a mapping mechanism is a more efficient use of this resource. The virtual resources $\mathcal{V}_1$, and $\mathcal{V}_2$ can then be used to instantiate hexadecimal numbers using the instantiation relation $\tilde{R}_{HEX}$. The real resource cannot be used to instantiate any higher representation level, shown by the cross in the figure.

In Figure 6, the real resource, $\mathcal{R}_1$, is an 8-bit number. This 8-bit number could be used to instantiate a hexadecimal number, or any other instantiation that uses 8 or less bits. However, a hexadecimal number only requires 4 bits to be instantiated, and using $\mathcal{R}_1$ would not be an efficient use of resources. Mapping two 4-bit virtual objects, $\mathcal{V}_1$, and $\mathcal{V}_2$ to $\mathcal{R}_1$, would be more efficient, and allows two hexadecimal (or any other 4-bit instantiation) to be instantiated, using the same resource, $\mathcal{R}_1$.

Virtualization is always achieved through the use of a *mapping mechanism* (MM) that maps virtual resources to real resources. The mapping mechanism is responsible for presenting the virtual resources as if they were real resources, and for maintaining the isolation between different virtual resources. We call this the *isolation* problem. The mapping mechanism also decides how resources are allocated; in other words deciding how to divide up or combine real resources to create virtual resources. This problem is known as the *embedding* problem. The embedding problem depends greatly on the isolation problem, since the method of isolation determines how the resources can be used. These two problems will be discussed in further detail later.

According to the authors of [16], and [17], the mapping mechanism is simply a function f that maps the set of virtual resources $\mathcal{V}$, to the set of real resources $\mathcal{R}$. The virtual resources can be thought of as the domain of f and the real resources as the codomain of f. The mapping function f maps each element in $\mathcal{V}$ to an element in $\mathcal{R}$.

$$f : \mathcal{V} \longrightarrow \mathcal{R} \cup \{t\}$$

such that if $y \; \varepsilon \; \mathcal{V}$ and $z \; \varepsilon \; R$ then

$$f(y) = \begin{cases} z \text{ if } z \text{ is the real resource for virtual resource } y \\ t \text{ if } y \text{ does not have a corresponding real resource} \end{cases}$$

The value $f(y) = t$ causes a trap or fault handling procedure to occur by the mapping mechanism.

In all figures and examples until now, only one physical resource has been considered, represented as one real resource in the abstract domain. However, virtualization can apply to *multiple* real resources, that are representations of *multiple* physical resources. Similarly, there are several ways in which the mapping of virtual to real resources can be done - we identify four types of virtualization. In Figure 7 we show the general types of virtualization, where there can be multiple physical resources, multiple real and virtual resources, and multiple types of mapping.

The four types of mapping virtual resources, $\mathcal{V}_n$, to real resources, $\mathcal{R}_m$ are:

1) One-to-one: Mapping a single virtual resource to a single real resource, to allow for easier management of resources;
2) Many-to-one: Mapping multiple virtual resources to a single real resource such as partitioning a single resource into a number of smaller and more easily accessible resources of same type;
3) One-to-many: Mapping one aggregated virtual resource to several real resources. Used to aggregate many individual components into larger resource pool; and lastly
4) Many-to-many: Mapping multiple virtual resources to multiple real resources. The combination of aggregating and partitioning resources to create completely customizable resources that can be tailored exactly to requirements.

Many-to-many can be considered the ideal case, since it enables resources to be used in the most flexible manner.

B. Recursion

When virtual resources are presented in such a way that they are indistinguishable from real resources, recursion is possible [18]. Users could choose to virtualize their virtual resources, since they perceive them as real resources. The mapping of resources is now done twice, the first mapping maps the virtual resources received by the user to the real resources, and the second mapping maps virtual resources to virtual resources.

The terms 'real' and 'virtual' can be confusing when recursion is taken into account. For this reason we always refer to the codomain of the mapping function, i.e. the resources that are being virtualized, as 'real' resources, even though these resources might already have been virtualized by a previous virtualization instance.

The resource mapping function, f, described above can be extended directly for recursion by applying the mapping function, f, multiple times and interpreting $\mathcal{V}$ and $\mathcal{R}$ as different instances of virtualization. Mapping function f_1 maps virtual resources $\mathcal{V}_1$ to real resources $\mathcal{R}$. Now, in a second virtualization instance, f_2 maps virtual resources $\mathcal{V}_2$ to 'real' resources $\mathcal{V}_1$.
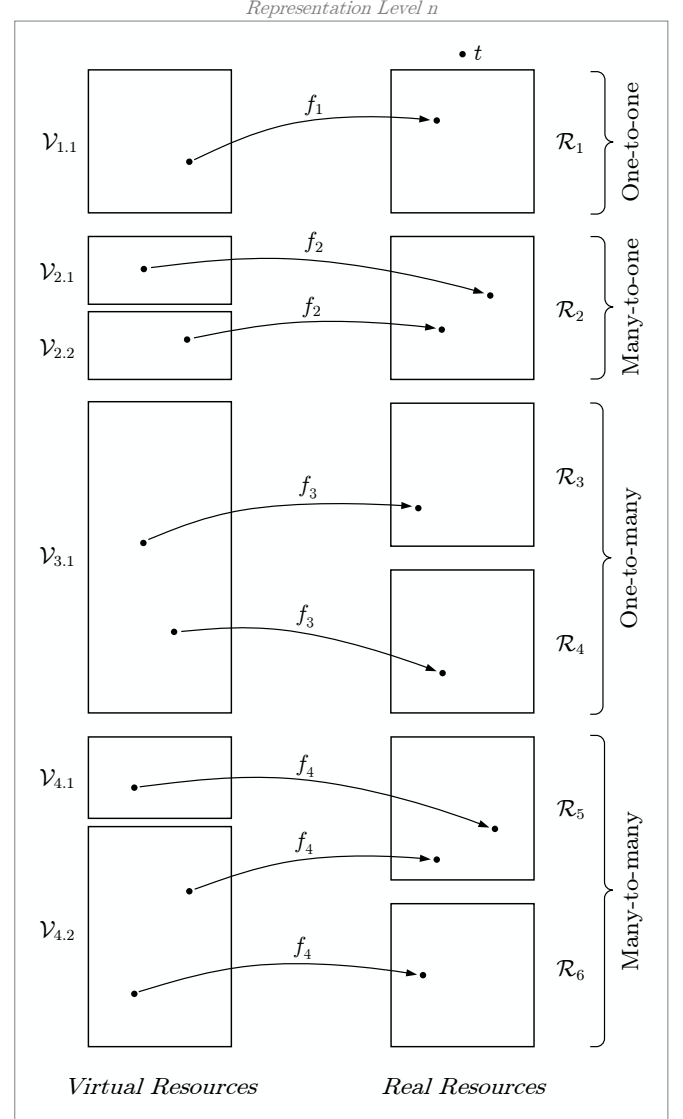
Figure 7. In this example, six physical resources are represented as real resources, $\mathcal{R}_{1-6}$, at some representation level n. Multiple virtual resources are mapped to the multiple real resources in different ways, showing the four types of virtualization. Many-to-many mapping is the ideal case as it enables abstract resources to be used in the most flexible manner.

$$f_1 : \mathcal{V}_1 \longrightarrow \mathcal{R} \cup \{t_1\}$$
$$f_2 : \mathcal{V}_2 \longrightarrow \mathcal{V}_1 \cup \{t_2\}$$

The real resources for f_1 are $\mathcal{R}$, and for f_2 the real resources are $\mathcal{V}_1$. Figure 8 illustrates several virtualization instances that recursively map virtual resources to real resources.

We can say that recursion is a requirement for virtualization, since non-recursive virtualization is simply multiplexing [19]. In the case of perfect virtualization, i.e. that virtual resources can be used exactly as real resources and no overhead exists, infinite recursion is possible [19].
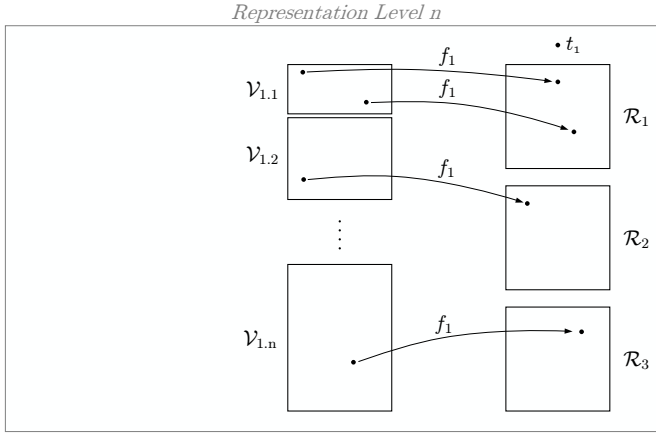
(a) In the first instance of virtualization, many-to-many mapping f_1 maps virtual resources $\mathcal{V}_{1.1-1.n}$ to real resources $\mathcal{R}_{1-3}$.
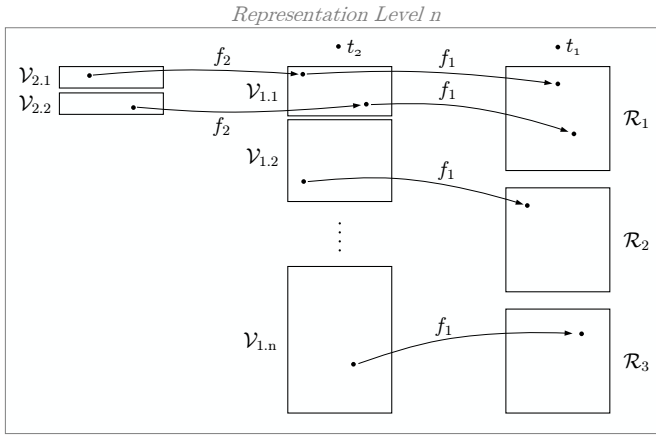
(b) Many-to-one mapping function f_2 is an example of recursive virtualization, since it maps virtual resources $\mathcal{V}_{2.1}$ and $\mathcal{V}_{2.2}$ to 'real' resource $\mathcal{V}_{1.1}$.

(c) The one-to-many function f_3 shows that it could be possible to map virtual resources to a combination of real resources, whether these have been previously virtualized or not. In this case $\mathcal{V}_{3.1}$ is mapped to $\mathcal{R}_4$ and $\mathcal{V}_{1.n}$.

Figure 8. Example of recursive virtualization. f_1 is the first instance of virtualization, f_2 is the second instance, and f_3 is the third. f_2 and f_3 are recursive virtualization, since they depend on the first instance of virtualization, f_1.

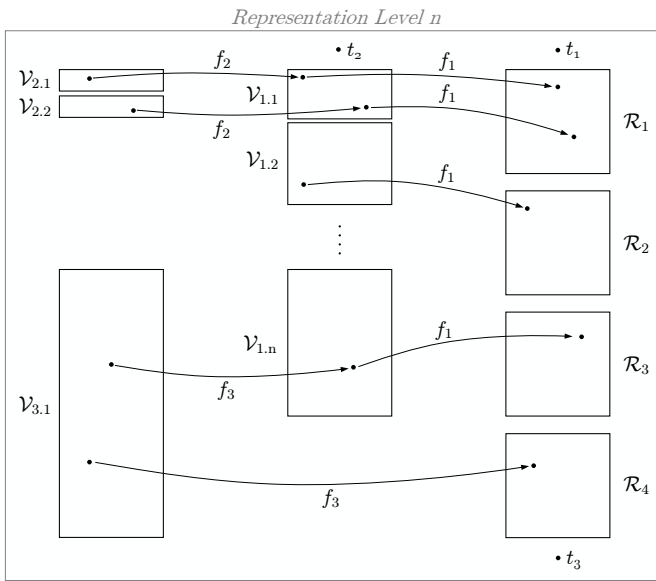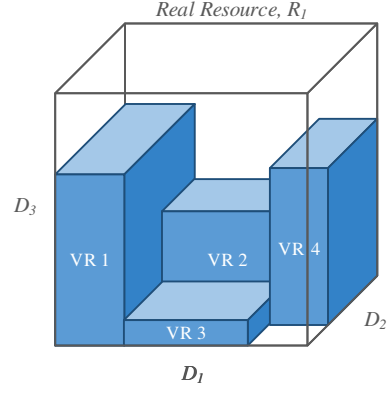Figure 9. The real resource, R_1, can be isolated along three dimensions, D_1, D_2, and D_3. The users of virtual resources, V_1, V_2, V_3, and V_4, are only aware of and can only access each of their individual resources, which appear to them as real resources.

C. The Isolation Problem

The isolation problem is the problem of choosing how to create virtual resources, and how to maintain independence between them. It should be impossible for VRs to interact with other VRs in any manner. By isolating along one or multiple dimensions of the real resources, each virtual resource user is only aware of its own virtual resources and can only use those resources, and thus it cannot interfere with other virtual resource users [20] [2]. This is illustrated in Figure 9. The term 'dimension' refers to a measurable feature of a resource.

Thus we add the isolation dimension(s) to the mapping function:

$$f_1 : \mathcal{V}_1 \xrightarrow{T} \mathcal{R} \cup \{t_1\}$$

for example using the time dimension. An example of how this applies is shown in Figure 10.
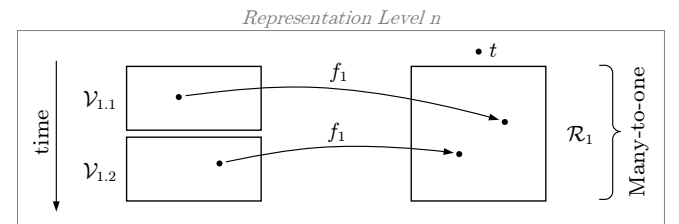
Figure 10. A many-to-one mapping using one dimension (time) for isolation.

However, the ability to isolate using a particular dimension depends on the technical capability of the mapping mechanism. The granularity used by the isolation process is very important as the user of the virtual resources must not be able to perceive any difference between the virtual resource and the real resource. For example, in processing virtualization, processing resources can be isolated in the time dimension. However, the timescale used by the isolation process is so small (smaller than human reaction time of approximately 0.2 seconds) that the user can use the virtual resources as if they

were real resources. If the timescale used by the isolation process was too large, say one hour, then the users of the virtual resources would realise that the resources they are using are not real resources.

Although the isolation problem is a prerequisite to the embedding problem, and influences the embedding problem greatly, in the literature the isolation problem has received significantly less attention compared to the embedding problem.

D. The Embedding Problem

The problem of deciding how to map virtual resources to real resources is also known as the embedding problem. Essentially this is a resource allocation problem, which is the distribution of scarce resources to competing users. There are several problems that can be considered:

1) The first case is that there might not be enough resources to satisfy all of the users' requests;
2) The second case is that the users can request different quantities of resources; and
3) The third case is that each of the resources can be of unequal value.

These three problems are not mutually exclusive and often occur simultaneously. Depending on the objective(s) that the resource owner wants to achieve, different metrics can be used to determine the optimal resource allocation. Even when there are enough resources to satisfy all of the users' requests and a potential solution exists, resource allocation can be a complex problem to solve.

In the context of virtualization, users can make requests for virtual resources and an embedding algorithm determines which requests are successful. Figure 11 shows the resource allocation problems that can occur when users make requests for two-dimensional sets of resources.

E. Definition of Virtualization

Now that we have a better understanding of virtualization, we propose the following definition for virtualization:

> Virtualization is a resource mapping that occurs in the abstract domain. Virtualization takes places within any one representation level in the abstract domain. The representation level used is a design choice. Abstract resources before virtualization, known as real resources, are limited by the granularity of the underlying physical resources. Abstract resources after virtualization, known as virtual resources, are not, and can be larger or smaller than real resources. Virtual resources are independent and can be allocated simultaneously to multiple users, each with the illusion of full ownership of their resources. Virtualization can always be done recursively, since users perceive no difference between the virtual resources and the real resources.

This definition is satisfactory since it addresses the concepts of 1) abstract resources as representations of physical resources. 2) the splitting and combining of resources, 3) ownership and isolation, 4) allocation, and 5) recursion.

(a) The first type of problem occurs when there are more user requests than resources. At least one user will not receive resources.

(b) Another type of problem can happen when users have different request sizes.

(c) A similar problem can occur if the resources are of different sizes.

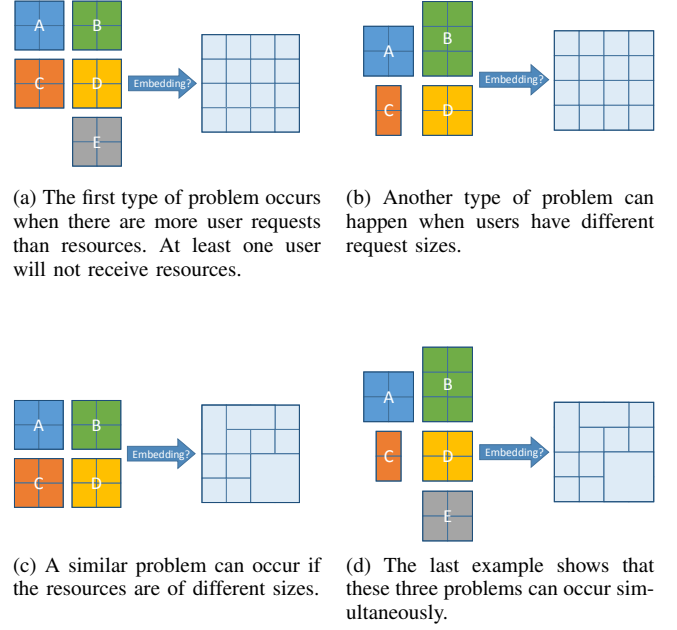(d) The last example shows that these three problems can occur simultaneously.

Figure 11. This figure shows resource allocation problems that can happen when users A, B, C, D, and E request two-dimensional resources from a resource owner. There are three different types of problems that can occur, shown in a), b) and c). The three can occur simultaneously as shown in d). Resources can be allocated request by request in sequence, or by considering multiple requests together, which could provide more efficient embedding.

F. Validity of the Theory

Although we have attempted to validate virtualization theory as much as possible by referring to previous works, some aspects are new and need verification. One method of verifying the theory is to examine virtualization technologies and to see if the theory holds. As an example we examine virtual memory, one of the first virtualization techniques developed.

Virtual memory, or one-level storage as it is also known, was developed in 1961 by the Atlas group to overcome the storage allocation problem of distributing information between main memory and auxiliary memory levels in computers [21]. In a one-level storage system, a distinction is made between the address space, which is the set of identifiers used to refer to information, and the memory space, which is the set of physical memory locations used to store information [22]. Instead of offering computer programs direct access to the memory space, programs can only access the address space, and a supervisor maps the address space to the memory space. By decoupling the address space in this way from the physical memory, it is possible to *combine* both main memory and auxiliary memory into a single address space, thus offering the illusion of one-level storage, shown in Figure 12.

Let us now examine how virtualization theory applies to virtual memory, summarized in Table II. Firstly, it is important to identify the abstract resources that are being virtualized. In this case the abstract resources are information storage in the form of bits. Next, it is possible to see that virtual memory is a one-to-many mapping, since it maps one virtual resource (the address space) to many real resources (the main
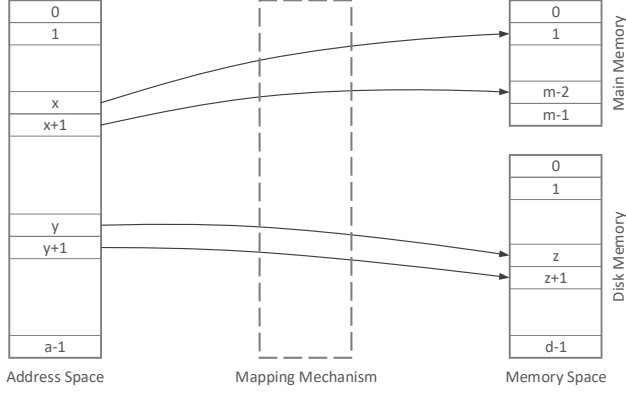
Figure 12. The separation of address and memory spaces allows the address mapping mechanism to combine main memory and disk memory into a one-level storage space. The address mapping mechanism can also give users the perception of having a unique address space for multitasking
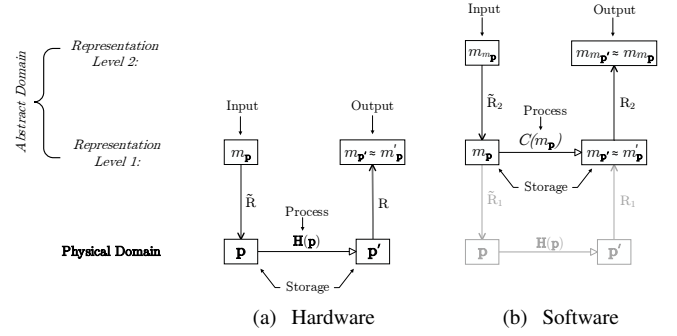

(a) Hardware   (b) Software

Figure 13. (a) Model of computer hardware based on AR theory. Computer hardware can be thought of as having four functions, corresponding to the mappings of AR theory and storage of the physical states. (b) Software must be used to enable all four functions to be virtualized, i.e. performing the compute cycle in the abstract domain, as virtualization can only be done in the abstract domain.

and auxiliary memory spaces) [4]. The isolation dimension is the address space, which is to say the index of locations to store information. In theory recursion is possible in virtual memory, because the resources offered are locations to store information, exactly the same type of resources as non-virtualized memory. In fact, the authors of [23] develop such a recursive virtual memory system. As for the embedding problem, some examples of different approaches to embedding are the different paging and segmentation algorithms that have been developed [17].

must first be able to virtualize nodes and links when creating virtual wireless networks, and thus as a first step in the virtualization process, we examine what constitutes nodes and links. We also analyse how node and link functionality can be virtualized.

A. Nodes

In this paper we consider a node or computer hardware as a physical device that can determine the outcome of abstract operations through physical manipulations [12]. Based on the commuting diagram of computer hardware shown in Figure 13 (a), we can identify four specific functions: *input* (I), i.e. the instantiation of abstract objects in a physical state, *storage* (S) of that physical state in some way, *processing* (P) of the physical state in a way that is commutative to some form of abstract operations, and *output* (O) of result of the physical process in the form of an abstract representation. All functions are required within a node, since, for example, it would be pointless to have processing available but no input, as there would be no information to process. These four functions are consistent with the IPO+S model of computing [24].

Each of these resource types can be represented in the abstract domain and virtualized, which allows resources to be used more efficiently, and also can provide new functionality, such as machine (node) virtualization. However, as mentioned previously, virtualization can only be done in the abstract domain. This means that if all four functions are to be virtualized simultaneously, then it is necessary to introduce an additional level of representation, that is to perform the virtualization in software. We show the show the general compute cycle in the abstract domain (software) in Figure 13 (b).

We refer to resources according to their function type, for example we refer to storage resources rather than resources that instantiate storage functionality. Although technically incorrect, this simplification makes it easier to follow. The types of node resource virtualization are discussed briefly below.

1) Process Virtualization: The idea of process virtualization can be traced back to the concept of compatible time-sharing, first developed at M.I.T. in the early 1960's [25].

Table II
VIRTUALIZATION THEORY APPLICATION: VIRTUAL MEMORY

Concept	Use in Virtual Memory
Abstract resources at a representation level	Location to store bits
Mapping	One-to-many / Many-to-many
Isolation	Address space - index of storage locations
Recursion	Possible but not very useful in practise, see [23]
Embedding	Paging algorithms, segmentation algorithms

The example of virtual memory shows how virtualization theory can apply in practise. We see that each aspect of the theory has a practical counterpart in virtual memory. This offers some validation of the theory. In the next section we examine several other virtualization technologies in the context of wireless networks, but for each virtualization technology the same analysis could be performed.

IV. NETWORK RESOURCE VIRTUALIZATION

Having developed a theory of virtualization, we now consider how virtualization applies to wireless networks. We define a wireless network as 'a set of nodes that can transfer information through links, where some of the links may be wireless in nature'. From this definition, we can deduce that wireless networks consist of two parts: nodes and links. We

---

[4]With the development of multitasking, in reality virtual memory is now many-to-many, because different programs each have their own address space.
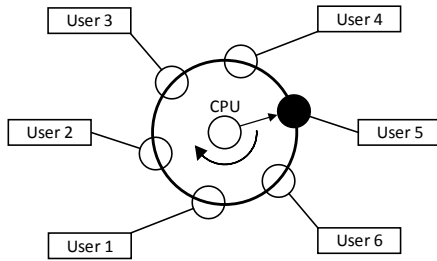
Figure 14. Time-sharing gives several users the perception of having exclusive use of a processor, by giving each user a short burst of computing time. In this example, the CPU is processing user five's burst of time, before it moves on to the next user.

Time-sharing was developed to overcome the limited man-machine interaction of batch processing, which had led to an increase in programming errors and debugging time, as larger and more complex programs were being set [26].

Time-sharing enables several people to make use of a computer at the same time, shown in Figure 14 [27]. Rather than offering users direct access to computing resources, which can lead to serious crashes and memory problems, a supervisor buffers user input, and sequentially runs user programs for small bursts of time. The full sequence of user programs occurs frequently enough (ideally in less than ~ 0.2 seconds) that a computer appears to be fully responsive to all users.

By mapping user processing time to burst of machine processor time in this way, and by maintaining strict isolation, users have the perception of exclusive use of dedicated processors. Thus the illusion of multiple 'virtual' processors is created. Thus virtualization both increases resource efficiency, and also offers users new or improved functionality. The surveys [28]–[30] provide further information about processing virtualization techniques, and a very interesting and informative documentary on the compatible time-sharing system can be found at [31].

2) Storage Virtualization: One method of storage virtualization is virtual memory, which we discussed in section III-F. The use of virtual memory not only automates the storage allocation problem efficiently [32], but also enables machine independence, program modularity, convenient memory addressing, and the capability of handling structured data [22], [17]. For a more detailed view on storage virtualization techniques, see the surveys [33], and [34].

3) Machine (Node) Virtualization: The development of process and storage virtualization led to new functionality, and computing became thought of as a large system of components serving a community of users, where each of the users could run different programs with different processing, memory, and I/O interaction requirements [35]. In such a system, software is commonly split into two classes to avoid system integrity issues: a privileged supervisor (or Operating System) which is presumed to be correct, and a second non-privileged class which is denied any functionality that can cause interference between processes [36].

However, this arrangement only allows one privileged supervisor to be run at a time, and incompatible non-privileged programs cannot be run easily [37]. Machine virtualization

overcomes this problem by constructing simulated copies of the machine, known as virtual machines, and each virtual machine can run a different privileged supervisor [38]. A virtual machine monitor (VMM), also known as a hypervisor, isolates process and memory operations for each virtual machine, and maps them to the host machine using time-sharing and virtual memory techniques [39]. Until recently I/O operations had to be trapped and executed by the VMM.

Advances in machine virtualization, especially in server virtualization have reduced the cost of servers hugely and has led to the widespread adoption of moving computing tasks to the 'cloud' [40]. The recent development of I/O virtualization (see next part) has allowed full computer node virtualization, consisting of storage, process and I/O virtualization. There are many contexts in which the use of node virtualization is growing such as desktop, application, and user virtualization. More information can be found in the works [39], [41], [42], [43], and [44].

4) Input and Output Virtualization: One of the problems encountered by early virtual machines was the mapping of input and output (I/O) paths from virtual device addresses to real device addresses, since absolute addressing is required for I/O paths [37]. Early Virtual Machine Monitors (VMMs) trapped I/O instructions used by the virtual machines, copied instructions, and 'absolutized' them by mapping the virtual addresses to the correct real I/O addresses. While this solution of emulating I/O devices in software enabled virtual machines to use I/O operations, it was a work-around and not I/O virtualization.

The generalized I/O memory management unit (IOMMU), developed by Intel, is a hardware device that maps virtual device addresses to real ones across isolated partitions, supervised by system software [45]. The VMM controls these partitions and thus full virtualization of I/O operations is possible, since virtual I/O operations are mapped directly to the devices. One example of an I/O peripheral that has been virtualized is the Network Interface Card. A survey of I/O virtualization techniques is given in [46].

B. Links

The function of links is to transfer information between nodes in a reliable manner. Similarly to nodes, links can also be thought of as consisting of abstract objects (i.e. information) and physical resources which instantiate that information and physically send it between nodes. Again, similar to nodes, links can only be virtualized in the abstract domain.

Links can be wired or wireless. However, there are significant differences between wired and wireless links, due to the nature of the physical resources used. Both wired and wireless links represent information using the electromagnetic spectrum, but in the case of wired links, the electromagnetic spectrum is isolated from other links through the use of physical cables. In wireless networks, since all links are broadcast, additional measures must be taken to provide isolation and reliability.

1) Wired Link Virtualization: Time-shared systems allowed users to work from remote terminals connected to a mainframe computer, with the perception of working at a personal

computer. Connections were implemented either using dial-up lines over the Public Switched Telephone Network (PSTN), or through private lines leased from the PSTN operators. The dial-up lines were much cheaper to use, but suffered from lack of security and functionality, while private lines offered good security and functionality but were expensive.

Virtual Private Network (VPN) services offer the security and functionality of private lines at much cheaper costs by exploiting the fact that typically communication between nodes in a network only occurs for a small percentage of time. Thus physical links can be time-shared to provide the illusion of private links, known as virtual circuits [47]. VPNs can be scaled much easier than physical private links, and can also be tailored to suit users' preferences [48].

However, although VPNs can isolate different logical networks over a shared infrastructure, they are prone to some limitations. Among these are that the coexistence of different networking solutions is not possible, and also that virtual networks are not fully independent [49]. Another limitation is that broadcasting is not supported in the same way as on native networks. A final limitation is that additional security measures are needed for VPNs which can add overheads, and these and other virtualization overheads can decrease the network reliability, throughput and latency.

2) Wireless Link Virtualization: Although wired link virtualization has existed for many decades, wireless link virtualization (WLV) is an active research area. As mentioned before, wireless link virtualization is the main focus of this paper. Therefore we briefly introduce wireless link virtualization here, whereas a detailed survey of WLV follows in Sections VI and VII.

Wireless link virtualization is the process of virtualizing wireless links, creating virtual resources that are isolated and which can use differing technologies and/or configurations independently. Although appearing to be similar to wireless resource sharing, wireless resource sharing is fundamentally different from virtualization, since wireless resource sharing does not create independent resources. In addition, virtualization allows the combination of resources to occur, which resource sharing does not.

There are several complications which exist in wireless link virtualization which do not exist in wired link virtualization due to the difference between the wired medium and the wireless medium. The first difference is that because of the inherent variation of the wireless channel over time, it is not possible to predict the information throughput of wireless links in advance. A second difference is that wireless links are broadcast, and thus have the potential to interfere with any other wireless link, whereas in wired links this does not occur [20]. Another complication for wireless links is that wireless nodes tend to be highly mobile, which means it is harder to predict and provision for the information transfer between nodes.

Because of the above reasons, WLV is a difficult problem and although there has been a significant amount of work on WLV in recent years, there are still many unresolved issues.

V. Network Virtualization and Wireless Network Virtualization

Having examined the virtualization of the resources that make up networks, and also armed with a better understanding of virtualization, we can now examine network and wireless network virtualization.

A. Network Virtualization

The idea of Network Virtualization (NV) has arisen as a solution to several problems with the Internet today. As a result of the Internet's success and ubiquity in many areas of life, it has become subject to ossification, since the competing interests of existing stakeholders and the large capital investment needed have led to high resistance to disruptive technologies [50], [51]. The need for global agreement between competing providers has limited innovation to simple incremental updates, which do not satisfy demand for new services and functionality, or to ad-hoc workarounds, which do satisfy legitimate needs, but violate core Internet design principles, and as such have impaired flexibility, security, reliability, and manageability [52], [49].

Network virtualization has evolved from the concept of using overlay networks to address these ossification issues. Initially, it was envisaged that overlay networks would be highly programmable platforms for innovation, while simultaneously allowing the existing network infrastructure to be maintained [53], [50]. However, later it was realized that overlay networks could not provide the innovation and flexibility desired because of the limitations of the underlying infrastructure, and that it would be necessary to virtualize the underlying network infrastructure [52].

Network virtualization can be considered as using both node and link virtualization to create complete virtual networks (sometimes called Meta-Networks) [54], [51]. Figure 15 shows the distinction between virtual nodes, virtual links and virtual networks. In addition to offering a potential solution to the Internet impasse, network virtualization also enables the separation of existing Internet providers into the roles of Infrastructure Provider (IP) and Service Provider (SP) [55]. As argued in [51] and [55], there are several reasons why this is beneficial:

1) Existing network providers have very few opportunities to distinguish themselves from their competitors and thus they equally have little incentive to develop and deploy new solutions. Decoupling IPs and SPs enables diversity and innovation, leading to better infrastructure and new and improved services for end users.
2) Separating the role of infrastructure and service providers lowers the barriers of entry significantly, since SPs do not need to invest in their own equipment. This leads to increased competition and innovation.
3) A separation of the two roles enables the sharing and aggregation of network infrastructure, leading to increased efficiency and cost savings.
4) In the current network model, new software and network protocols cannot easily be tested, and often not under real traffic conditions. By splitting the roles of SP and IP,
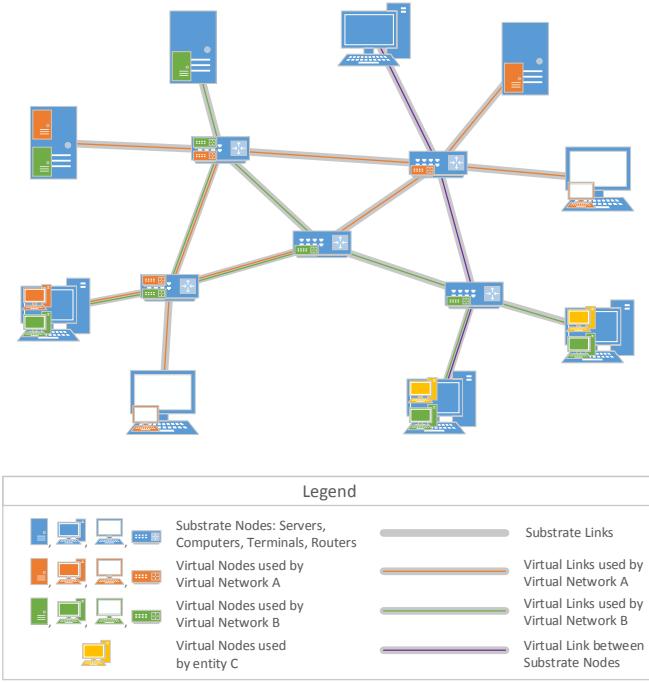
Figure 15. Network Virtualization: Virtual networks A (orange) and B (green) can coexist simultaneously on the underlying substrate. These networks can offer customized services to their users in a secure and isolated manner. Another entity C requires several computing nodes, but no network connectivity and so node virtualization provides entity C with virtual nodes (yellow). Meanwhile, virtual links or VPNs (purple), can be used to connect several substrate nodes.
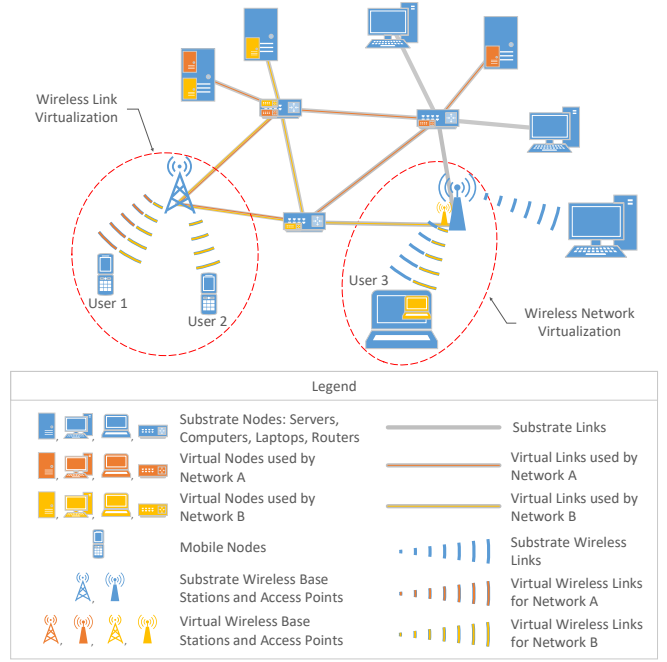

Figure 16. Wireless Link and Wireless Network Virtualization: Networks A (orange) and B (yellow) use network virtualization to coexist simultaneously on the wired network substrate. Wireless link virtualization is used to offer secure and customised services to users 1 and 2, and these services can focus on different aspects such as high-throughput, low-latency, or low-power consumption, etc. In this example, user 1 has access to two types of services, one from networks A and B each, while user 2 only has access to the service from network B. Wireless network virtualization is used to extend virtual network B to user 3.

new experimental software and protocols can be tested in isolated networks, without affecting existing services. The isolation of networks also enables SPs to customise their networks and achieve better security.

Because of the many potential benefits that can be obtained, network virtualization is an area of ongoing research, and several NV testbeds have been constructed. Initially, testbeds have focused on node virtualization such as PlanetLab [50] and GENI [56], but more recently, testbeds such as CABO [55] and 4WARD [57] are interested in full network virtualization, i.e. both node and link virtualization combined. There is also significant ongoing research on network virtualization, summarized in the surveys [58] and [59].

Two interesting ongoing research topics which are related to network virtualization are Software Defined Networking (SDN) and Network Function Virtualization (NFV). Although these topics are outside the scope of this paper, they should be discussed briefly.

Software defined networking can be seen as "the separation of [network] forwarding hardware from the control logic" [60]. Thus SDN can be viewed as an enabler for network virtualization, since it allows functionality to be implemented in the abstract, rather than the physical domain. Referring back to Figure 13, we can think of SDN as moving functionality, i.e. the compute cycle, from (a) hardware to (b) software. However, it is important to note that the use of SDN does not imply virtualization. There are many benefits to be gained from using SDN in of itself [61].

The idea behind network function virtualization is to decompose a given network service into a set of functions which can be implemented on commodity hardware through software virtualization techniques [62], [63]. NFV can be considered as a form of node virtualization, albeit specifically tailored to networking. For example, parts of the mobile core network such as the mobility management entity, the home subscriber server, and many other functions could be virtualized to enable flexible and dynamic operation. NFV represents a significant step towards network virtualization, as techniques from NFV could be used in network virtualization.

B. Wireless Network Virtualization

Wireless network virtualization has been proposed as an extension of network virtualization to wireless networks, providing similar potential benefits in terms of flexibility and efficiency [64], [10]. Specifically, it is hoped that WNV can increase resource efficiency and flexibility for the problems of 1) the continuing rapid growth of demand for wireless services, 2) the ever-greater demand for diversity of services, and 3) the increasing costs of wireless infrastructure.

Similarly to network virtualization, we can think of WNV as using both node and wireless link virtualization to create virtual wireless networks. Only when it is possible to virtualize both the node and link resources that make up the network, is it possible to consider (wireless) network virtualization [1] [4]. As we have seen previously, node virtualization and wired

link virtualization are long-established techniques. However, wireless link virtualization is still at an early stage of development, and without WLV it is not possible to do wireless network virtualization. In fact, most work on wireless network virtualization to date has actually been concerned with wireless link virtualization, since WLV is a prerequisite of WNV. In this work, we distinguish between wireless link virtualization and wireless network virtualization, shown in Figure 16.

Although we are ultimately interested in wireless network virtualization, in this work we focus on wireless link virtualization for several reasons:

1) WLV is a prerequisite for WNV, and the other prerequisite, node virtualization, can already be implemented through many existing techniques.
2) The main difference between NV and WNV is the nature of the links, and thus once link virtualization has been accomplished, we could reuse NV techniques for other aspects of WNV, such as coordinated node and link mapping [65].
3) Work on WNV so far has almost exclusively focused on WLV, although it has self-identified as WNV.

Now knowing that WLV is the main research aspect of WNV, in the next section we perform a survey of existing WLV techniques (although many of these techniques self-identify as wireless *network* virtualization) in order to identify open research problems. We use the theory of virtualization to aid us with the classification and analysis of these works.

VI. Survey of Wireless Link Virtualization

We now apply the theory of virtualization to the main focus of the paper; the survey on wireless link virtualization to establish open research questions.

A. Test for Virtualization

The first part of performing the survey was to develop a test for virtualization, to decide whether a technique that self-identifies as virtualization actually is virtualization. We return to the definition of virtualization to define such a test.

According to the definition of virtualization, virtualization is always recursive in theory. Therefore, one possible test for virtualization is to check for recursion. We propose a test which determines whether proposed techniques are recursive, at least in theory.

All of the papers which we examine define abstract wireless link resources in some manner, and subsequently allocate these resources to different users based on some criteria. The allocation of resources to users is done through the use of an algorithm, whether this is an optimization, a heuristic method, an auction-based approach, or any other resource allocation method. These resource allocation methods all have one thing in common; they have a set of **input** resources and a set of **output** resources. Input resources are the wireless link resource that algorithms allocate, while output resources are the resources that the users receive from algorithms. In general, input resources are relatively straightforward to determine, but output resources can be harder to identify.

Therefore we base our test for recursion on the mapping of input to output resources. We define recursive techniques as techniques in which no difference exists between input and output resources. In other words, the *types* of both resources are the same, and both the input and the output resources can be thought of as at the *same* representation level. Techniques for which this is true can be considered recursive, and hence virtualization, and we can call the input and output resources real and virtual resources respectively.

The recursive test for virtualization is very simple:

> **Recursive Test for Virtualization**
> 1) Check the **units** of the input resources.
> 2) Check the **units** of the output resources.
> 3) If they are the same, then the technique can be considered recursive and is virtualization.

However, techniques for which the type of input and output resources differ cannot be considered recursive. In this case the input and output resources are at different representation levels. Users cannot use the same technique again to split or aggregate resources. These techniques can be considered methods of resources sharing/combination, but not virtualization.

For example, a technique for which the input is spectrum resources (measured in Hertz) and the output given to users is also spectrum resources (also measured in Hertz), is considered a recursive technique, since users could use the same technique again to virtualize their own spectrum resources, and split or aggregate the resources in a new way.

However, a technique which has as input spectrum resources (measured in Hz), and as output information rate (measured in bit/s) can not be considered recursive, since the units of the input and output resources are different. Users could not reuse this technique to split or aggregate their information rate.

B. Survey Overview

The survey analyses over 60 works from 2003 onwards that self-identify as wireless network virtualization and propose algorithms for virtualization. However as discussed already, these works are in fact concerned with wireless link virtualization. Therefore the survey is about wireless link virtualization.

We apply the recursive test to a representative selection of papers which self-identified as virtualization, to determine whether these works were also deemed virtualization according to the formal definition of virtualization developed in this paper. We consider works on any type of wireless link technology, and at any layer of the communication stack. Only papers that propose resource mapping techniques are included, and thus many framework and architecture papers are omitted.

We split the papers into two groups: those that fail the recursive test, described in subsection VI-C, and those which pass the recursive test and are therefore deemed to be relevant, analysed in subsection VI-D. We briefly discuss why some papers did not pass the test; and why they are considered resource allocation techniques according to our definition of virtualization, rather than virtualization techniques.

Table III
NON-RECURSIVE WIRELESS LINK RESOURCE ALLOCATION TECHNIQUES

	First Author	Reference	Input Resource Type (Units)	Output Resource Type (Units)	Recursive?
Resource Allocation Techniques	Lu	[66]	Subchannels (KHz)	Rate (Kbit/s)	No
	Lv	[67]	Channels (MHz)	Bandwidth – i.e. rate (Mbit/s)	No
	Xu	[68], [69]	Subchannels (MHz)	Sum Rate (bit/s/Hz)	No
	G. Liu	[70], [71]	Subchannels (MHz)	Rate (Mbit/s)	No
	Kamel	[72]	Subchannel (MHz)	Rate (Mbit/s)	No
	G. Zhang	[73], [74]	Subcarrier (MHz)	Data Rate (bit/s)	No
	Feng	[75] - Spectrum	Spectrum (MHz)	Traffic (Mbit/s)	No
	Cai	[76]	Radio Resource (KHz)	Utility (Bit/s/$)	No
	Chen	[77]	Channel (MHz)	Utility (Profit/bitrate)?	No
	Liang	[78]	Radio Resource (MHz)	Utility (?)	No
	Khatibi	[79]	Radio Resource Unit (MHz)?	Rate(Mbit/s)	No
	Rahman	[80] - LVN	Spectrum (MHz)	Rate (Mbit/s)	No
	Fu	[81], [82]	Fraction of time of subchannel (ms, KHz)	Information-theoretic Rate (Kbit/s)	No
	Kamel	[83]	Physical Resource Blocks (KHz, ms)	Rate (Mbit/s)	No
	Zaki	[84], [85] - Dynamic	Physical Resource Blocks (KHz, ms)	Load (Kbit/s)	No
	Zhao	[86]	Physical Resource Blocks (KHz, ms)	Rate, Delay (Mbit/s, ms)	No
	Li	[87]	Physical Resource Blocks (KHz, ms)	Date Rate (Kbit/s)	No
	B. Liu	[88]	Physical Resource Blocks (KHz, ms)	Traffic Rate (Kbit/s)	No
	Panchal	[89]	Physical Resource Blocks (KHz, ms)	Load (Kbit/s)	No
	Kalil	[90] [91]	Resource Blocks (KHz, ms)	Throughput (Mbit/s)	No
	Moubayed	[92]	Resource Blocks (KHz, ms)	Sum Rate (Kbit/s)	No
	Kokku	[93], [94]	OFDM Blocks/Slots (MHz, ms)	Rate/QoS (Mbit/s)	No
	Costa-Perez	[95]	LTE Resource Blocks (MHz, ms)	Rate (Kbit/s)	No
	Wei	[96]	Physical Resource Blocks (KHz, ms)	Utility (bit/Joule)	No
	Wang	[97]	Physical Resource Blocks (MHz, ms)	Throughput(Mbit/s)	No
	Di Stasi	[98]	Channels in Space (Hz, Grid Location)	Throughput (Mbit/s)	No
	Q. Zhu	[99], [100]	Subchannels and Power (Hz, Watts)	Utility (Hz/m^3)?	No
	Fan	[101]	Bandwidth and Power (MHz, dBm)	Rate (Mbit/s)	No
	K. Zhu	[102]	Subchannels, Power and Antennas (Hz, Watts, # of antennas)	Rate (bits/s)	No
	Ahmadi	[103]	Spectrum, Antennas (KHz, # of Antennas)	Rate (Mbit/s)	No
	Ahmadi	[104]	Spatial Stream (MHz, time block)	Rate (Mbit/s)	No
	Rahman	[80] - CVN/RVN	Spectrum, Remote Radio Heads (MHz, # of)	Rate (Mbit/s)	No

C. Non-Recursive Techniques

We briefly discuss the papers that did not pass the test, shown in Table III, to examine why they failed the test. Although these works are not virtualization according to our definition, these techniques still represent significant contributions in wireless link resource allocation and allow wireless link resources to be shared in an efficient and/or fair manner.

The works in Table III are organised by the type of resources that they allocate and by date. As can be seen, we found several types of papers which consider spectrum resources as input, whether isolated in frequency, frequency and time, or space and other dimensions.

1) Frequency-based isolation: As an example of the recursive test, we look at the works [66]–[74], [79], [80]. In these works, input spectrum resources are allocated as subchannels (in units of Hz) to users based on the rate (in bit/s) that can be achieved by users. Thus the output users are receiving is a data rate, rather than a number of subchannels. Although these methods advance the field of spectrum resource allocation, because the input and the output resources are not of the same type, these techniques cannot be considered recursive, and subsequently do not fall under our strict definition of virtualization. Similarly, the work [75] proposes a spectrum resource sharing technique that relies on estimating the users' traffic requirements, and thus the output resources are at a higher level of representation than the input resources.

2) Frequency and Time based isolation: There are also techniques which allocate input spectrum resources using frequency and time isolation, either based on users' estimated traffic loads and requirements, or based on the best rates that users can achieve. The works [84]–[90] can be considered in the first category, and take into consideration users' traffic requirements when assigning spectrum blocks. The second category contains the works [81]–[83]. Similarly, the works [93], [94], and [95] implement resource allocation schemes that allocate wireless resources based on the data rate requested by the virtual networks. These implementations are very impressive resource sharing techniques, but again because the techniques are not recursive, they cannot be considered virtualization.

3) Space-based isolation: Several works such as [98]–[104] propose allocating virtual spectrum resources using space-based isolation methods such as power control, and Multiple Input Multiple Output (MIMO) schemes. However, these schemes rely on the Channel State Information (CSI) to allocate virtual spectrum resources. Since CSI is an information-

level resource, these schemes cannot be considered recursive.

Moreover space-based isolation can only be achieved if the resource users do not care which spatial resources they receive - in other words if the location of the users does not matter. The reason for this is that resources that are requested and allocated in a particular direction are different to resources that have no spacial restrictions. For example, allocating bandwidth to a spatial stream means that units of the resource are now Hz in a particular direction of a coordinate system, and we can think of the spectrum resource as changing from a scalar to a vector resource.

In fact, directed space-based isolation schemes enable spatial densification, and MIMO techniques can be seen as analogous to increasing the number of wireless nodes [105]. Taking this perspective clarifies why directed spectrum resources are not the same as virtual spectrum resources.

Although the techniques discussed in this subsection provide clever and efficient methods to share limited spectrum resources efficiently, they take as input spectrum resources, but the output resources that users receive are at the information representation level (see next section). Although this leads to better spectral efficiency and more optimal use of resources, this means that these techniques cannot be considered virtualization.

D. Recursive Techniques

The papers which pass the recursive test are shown in Table IV, where we have combined the input and the output resources types for brevity (since they are the same). We discovered that the recursive test could also be used to determine the representation level of a virtualization technique. Methods which have the same unit types (for example, Hz, KHz, MHz, etc.) can be grouped together at the same representation level. We found that there were two main representation levels for wireless link resources:

  1) Spectrum Level Virtualization, which the virtualization of wireless resources in the form of spectrum, such as [106].
  2) Information Level Virtualization, also known as data path or flow-level virtualization, which refers to the virtualization of the information carried by the spectrum, and allows wireless links to be shared or combined at the information level, such as [64].

Table IV classifies papers firstly according to the representation level, which is in our opinion the most important distinction between virtualization techniques. Subsection VI-E discusses spectrum-level virtualization techniques, while subsection VI-F discusses information-level techniques. We also examined the types of isolation used at each representation level, and in particular the isolation methods that seemed to be more successful. Therefore in the table, the second classification is by isolation dimension(s), ordered by date. The third consideration of the survey is the embedding method used by virtualization techniques. Due to space limitations, the different embedding methods are described in Table V. A final aspect that we examined was the type of mapping that techniques performed, which was very heavily focussed on many-to-one mapping, i.e. partitioning.

In fact, from the survey it became clear that several works have also touched on the concept of 'levels of virtualization' or 'depth of virtualization', such as [4]–[7], [9], [93], [107]. However the majority of these works describe the levels in terms of the types of resources used, rather than different representation levels. From analysis of these works, it became clear that the terms 'levels of virtualization' and 'depth of virtualization' often refer to the concept of virtualization at different levels of representation, but that this is not articulated clearly in these works.

E. Spectrum Level Virtualization

To the best of our knowledge, spectrum-level virtualization was first proposed in [108], in which wireless network virtualization is considered as an extension to the wired network virtualization testbed GENI. The authors of [108] conceptually discuss several methods of isolation at the spectrum level, such as along the frequency, time, space, and code dimensions or a combination of these. The authors of [2] propose a virtualization framework for wireless networks and discuss some of the steps involved in deploying virtual wireless networks.

1) Frequency-based Isolation: One of the first proposals of a spectrum virtualization technique was [3]. This work proposed an auction-based method for mapping virtual spectrum bands to real spectrum bands, and simulated the allocation of resources to maximize the revenue for the auctioneer.

The work [106] implements a Spectrum Virtualization Layer (SVL) that offers virtual spectrum resources to users according to their desired bandwidth. In this way the limitations of real spectrum with fixed bandwidths and non-contiguous segments can be overcome. The virtual spectrum is mapped to real spectrum according to a predefined spectrum map. The wireless link mapping is many-to-many as it can map the virtual baseband to a real spectrum band with the same bandwidth, a narrower bandwidth, or to several non-contiguous bands with a greater combined bandwidth. The authors show that SVL can be used to create a virtual Wifi Network over TV White Space channels, and that SVL could be used to combine the access points of several different technology standards.

The authors of [109] and [110] propose a channel virtualization scheme to share spectrum opportunistically, based on the price that virtual networks are willing to pay for virtual channels. The requirements of virtual networks are split into baseline and fluctuant parts. In this work the virtual channels used for the fluctuant requirements can be shared between different virtual networks and thus there is a probability of collision. The authors propose several algorithms to assign the virtual resources in such a way that maximizes the revenue for the owner of the real spectrum, and perform simulations to compare these and other algorithms from wired network virtualization.

2) Frequency and Time based Isolation: The works [85] and [84] first propose a spectrum virtualization method involving frequency and time isolation. The idea in these works is to virtualize Long Term Evolution (LTE) base stations, also called eNodeBs, using a virtual machine hypervisor that also

Table IV
WIRELESS LINK VIRTUALIZATION: ISOLATION

	First Author	Reference	Input & Output Resource Type (Units)	Recursive?	Representation Level	Isolation	Mapping
Virtualization	Forde	[3]	Bands (MHz)	Yes		Frequency	$m \mapsto 1$
	Tan	[106]	Baseband (MHz)	Yes		Frequency	$m \mapsto m$
	Yang	[109], [110]	Channel (MHz)	Yes		Frequency	$m \mapsto 1$
	Zaki	[84], [85] - Static	Physical Resource Blocks (KHz, ms)	Yes	Spectrum	Frequency and Time	$m \mapsto 1$
	Yang	[111]	Physical Resource Blocks (KHz, ms)	Yes			$m \mapsto 1$
	van de Belt	[112], [113]	Physical Resource Blocks (KHz, ms)	Yes			$m \mapsto 1$
	Hsu	[114]	Bandwidth and Time units (KHz, ms)	Yes			$m \mapsto m$
	Lu	[115]	Physical Resource Blocks (KHz, ms)	Yes			$m \mapsto 1$
	X Zhang	[116]	Timeslot and (x, y) coordinates in Space (ms, (meter, meter) from [117])	Yes		Time and Space - Power	$m \mapsto 1$
	Chandra	[64]	Information Throughput (Kbit/s)	Yes		Time - Switching	$m \mapsto 1$
	Smith	[118]	Throughput (Kbit/s)	Yes			$m \mapsto 1$
	Coskun	[119]	Throughput (Kbit/s)	Yes			$m \mapsto 1$
	AlHazmi	[120]	Packets (Bytes)	Yes			$m \mapsto 1$
	Lv	[121]	Packets (Bytes)	Yes			$m \mapsto 1$
	Mahindra	[122]	Throughput (Mbit/s)	Yes	Information		$m \mapsto 1$
	Perez	[123]	Packets (Bytes)	Yes			$m \mapsto 1$
	Sherwood	[124]	Throughput (MBit/s)	Yes			$m \mapsto 1$
	Bhanage	[125], [126]	Throughput (Mbit/s)	Yes		Time - Addressing	$m \mapsto 1$
	Xia	[127]	Throughput (Kbit/s)	Yes			$m \mapsto 1$
	Aljabari	[128]	Throughput (Kbit/s)	Yes			$m \mapsto 1$
	Nakauchi	[129]	Throughput (Mbit/s)	Yes			$m \mapsto 1$
	Katsalis	[130]	Throughput (Mbit/s)	Yes			$m \mapsto 1$
	Feng	[75] - Flow	Rate(Mbit/s)	Yes			$m \mapsto 1$
	Mahindra	[122]	Throughput (Mbit/s)	Yes		Space - Power	$m \mapsto 1$

has control of the spectrum resources. The virtual eNodeBs can request physical radio resource blocks (PRBs) from the hypervisor, The authors propose two different algorithms - a static algorithm where the virtual PRBs are assigned once, and a dynamic algorithm which depends on the operator's load. However, only the static algorithm can be considered virtualization, because the dynamic algorithm fails the test for recursion since the hypervisor assigns virtual PRBs based on a higher level of representation (information-level metrics such as channel conditions, user loads and QoS requirements).

The authors of [111] propose a virtual resource embedding algorithm for frequency and time domain resources. In this work, virtual network operators can request a number of contiguous frequency and time-domain resources for a duration of time periods. Virtual resource requests can be chosen with several priority levels, corresponding to the likelihood of acceptance of a request. The heuristic algorithm proposed is based on finding the most suitable location for a request using Karnaugh-map theory and the embedding density, a measure of how efficiently resources are used. The authors compare several algorithms through simulations in terms of resource utility and the acceptance ratio of virtual resource requests.

Our work [112] extends the Karnaugh-map algorithm to the dynamic case, which allows virtual resources to be re-embedded at every time period. Through simulations, we show that this uses resources in a more efficient manner and leads to a higher request acceptance ratio. We also propose a greedy dynamic embedding algorithm which has increased performance compared to the dynamic Karnaugh-map algorithm, and formulate the objective problem for the

virtual resource embedding problem with prioritised requests.

In another work, [113], we demonstrate how virtualization could be applied to a use-case scenario, in the context of a virtual public safety operator. We show that virtualization enables spectrum resources to be used in a flexible manner that guarantees connectivity for public safety users during an emergency, but allows spectrum resources to be used by commercial operators as needed during day-to-day operation.

The work [114] proposes a dynamic virtual resource embedding algorithm that allows virtual resource requests to be fulfilled through multiple virtual resource blocks. In other words the real resource owner can aggregate spectrum blocks to fulfil resource requests. This use of spectrum aggregation means that the embedding used is potentially many-to-many.

In the work [115], the dynamic greedy algorithm is extended by using a genetic algorithm to find the most fit embedding location for a virtual resource request. The authors show that the genetic algorithm offers a small improvement over the dynamic greedy algorithm in terms of resource utility and rejection rate.

3) Time and Space based Isolation: The work [116] proposes a time and space based isolation method to allocate wireless network resources to experiments on a grid. The objective is to fit as many experiments into a limited grid, while maintaining strict isolation. The allocation takes into account the distance between nodes as a means of modelling potential interference. The authors show that using space-based isolation in addition to time-based isolation can provide significant gains in resource utilization. Gains are more significant for small scale experiments than for larger experiments.

One important aspect to note is that space-based isolation is possible in this case because the location of the nodes does not matter. However, in more commercial applications this assumption most likely will not be true, and therefore space-based isolation will not be possible (see section VI-C3).

4) Discussion: Spectrum-level virtualization has so far mainly focused on frequency isolation, plus the combination of isolation by frequency and other dimensions such as time or space. One reason for this might be that frequency isolation is already extensively used in networks as frequency division multiplexing.

There are several reasons that could explain why frequency isolation is so widely used. The first is that due to the nature of spectrum resources, different parts of the spectrum are suitable for different applications and thus it is intuitive to divide up the spectrum into bands for different applications. Another reason why frequency isolation could be preferable to time isolation is that it allows wireless links to be operational at all times, whereas using time isolation this is not possible. Time isolation also requires that nodes in a network be synchronised tightly.

Space isolation is also used widely in wireless network and allows the spectrum to be reused in multiple locations. However space isolation is possibly the most difficult form of ensuring independence between different links, due the unreliable and random nature of propagation of radio waves.

With the exception of [106], to date the work on spectrum virtualization has focussed heavily on simulation and on efficient embedding schemes, rather than on implementing virtualization techniques with strict isolation. One of the reasons for this might be that most of the works deal with cellular technologies, and thus there is an emphasis on efficiency, as it is costly and time-consuming to add additional capacity to these types of networks. Another reason could be the challenges of implementation of virtualization in these networks, since many techniques for cellular networks rely on strict synchronisation.

F. Information Level Virtualization

The concept of wireless network virtualization was first seen as an extension to wired network virtualization [108], [131], and initial wireless link virtualization techniques focused on information-level link virtualization, which had already been accomplished for wired networks (see section IV-B1). The idea of Virtual Local Area Networks (VLANs) had also been established for wired networks, and so some of the first wireless link virtualization papers focused on finding methods to allow multiple virtual links to coexist on the same physical wireless device at the information level.

1) Time-based Isolation using Switching: One of the first papers to describe and implement a method of wireless link virtualization was [64]. In this paper, the authors address the problem of connecting a wireless node to multiple networks simultaneously, which up until this point could only be done using multiple wireless front-ends. To solve the problem of expense and excessive battery consumption by multiple wireless cards amongst other reasons, the MultiNet solution

was developed, which time-switched a wireless card between multiple virtual networks at regular intervals, or at adaptive intervals based on traffic load. Virtual interfaces are presented to the node at the information level (Media Access Control (MAC) layer), and the node can send and receive packets through one or more interfaces simultaneously. The interfaces appear as if they are constantly active, but in practise the device driver buffers packets until the right virtual network is active. This solution reduced the energy consumed dramatically compared to using multiple cards, since two radios consumed about double the energy consumed by MultiNet. However, the process of switching between networks means that the association procedure is initiated every time a switch occurs, and leads to a decrease in performance in terms of delay, throughput and packet loss when compared to solutions involving multiple wireless cards.

The work [118] proposes a round-robin time-switching approach for virtualizing wireless links used by virtual nodes. These virtual networks are co-ordinated centrally to ensure synchronization between nodes. However, because of propagation delay, some overhead is incurred when switching from one virtual network to another and this leads to a relatively high packet loss and increased latency.

To deal with the issues of long handover times when switching between virtual links, the authors of [119] use the Power Saving Mode (PSM) to shorten delays compared to virtualization without PSM enabled. They also take advantage of the Point Coordination Function (PCF) features of wireless cards to avoid repeating the association procedure for every switch. In [120], an extension of [119], the authors investigate the effects of virtualization on traffic properties.

The work [121] examines time-switching in the context of wireless mesh networks, and proposes a round-robin scheme to broadcast packets from differing virtual networks. Because links can be unreliable, and no acknowledgement is used when using the broadcast mode, the algorithm rebroadcasts certain packets based on a successful reception probability.

However some of the disadvantages of the time-switching approach are that a high level of synchronization is required between different nodes, and also that when there are many virtual networks the delay for each network increases, which affects the performance of delay-sensitive applications.

2) Time-based Isolation using Addressing: The concept of Virtual Access Points (VAPs) was first introduced in [132] as a means of allowing multiple access points (and thus multiple networks) to coexist simultaneously on one physical access point. Each VAP is allocated a unique Service Set Identifier (SSID) and capability set. The use of addressing in this way provides time isolation at the information level (MAC layer) and enables VAPs to be indistinguishable from physical access points [132]. VAPs cannot emulate the operation of a physical AP at the radio frequency layer (Spectrum level) unless multiple physical radios are available. This limitation requires all virtual networks to use the same physical radio parameters such as channel. However, VAPs can differentiate at the information level, offering different throughput rates, packet loss, and latency, etc.

One of the first papers to implement Virtual Access Points

Table V
WIRELESS LINK VIRTUALIZATION: EMBEDDING

	First Author	Reference	Isolation	Embedding
Spectrum Level	Forde	[3]	Frequency	Auctioning of paired spectrum bands based on price and distance between bands
	Tan	[106]		Implementation of spectrum virtualization based on predefined spectrum map(s)
	Yang	[109], [110]		Baseline and varying channels with collision probability. Comparison of heuristic, genetic, and optimal allocation algorithms based on price
	Zaki	[84], [85] - Static	Frequency and Time	Predefined spectrum allocation
	Yang	[111]		Heuristic embedding based on Karnaugh-map areas
	van de Belt	[112], [113]		Dynamic greedy embedding and dynamic version of [111]
	Hsu	[114]		Bottom-Left algorithm with splitting of resources
	Lu	[115]		Dynamic genetic algorithm for Karnaugh-map areas
	X Zhang	[116]	Time and Space - Power	Embedding based on distance between nodes and experiment duration
Information Level	Chandra	[64]	Time - Switching	Time-switching at either fixed intervals or adaptive intervals based on traffic load
	Smith	[118]		Synchronised time-switching in a round-robin fashion
	Coskun	[119]		Fast time-switching at regular intervals using PSM and PCF features
	AlHazmi	[120]		Extension of [119] with more detailed analysis of virtualization capabilities
	Lv	[121]		Round-robin time-switching with opportunistic rebroadcasts
	Mahindra	[122]	Time - Addressing	Policy manager limits each experiment to a predefined maximum rate
	Perez	[123]		Weighted round-robin scheduler
	Sherwood	[124]		Priority resource allocation policy configured by network administrator
	Bhanage	[125]		Based on airtime quotas allocated at hardware setup.
	Bhanage	[126]		Weighted sharing based on an airtime fairness metric
	Xia	[127]		Not clear how vMAC interfaces are embedded but possibly through rate control
	Aljabari	[128]		No specific embedding - limited by machine performance
	Nakauchi	[129]		Embedding based on a weighted fraction of airtime
	Katsalis	[130]		Allocated according to predefined throughput share guarantees
	Feng	[75] - Flow		Flow scheduler allocates flows to virtual networks based on traffic load
	Mahindra	[122]	Space - Power	Policy manager limits each experiment to a predefined maximum rate

is [122]. This paper compares space-based and time-based isolation for virtualization on the ORBIT testbed. Space-based isolation will be discussed later. For the time-based isolation the authors prefer the VAP approach rather than the time-switching approach, because of the disadvantages of the time-switching approach already discussed. A policy manager limits the maximum rate that each experiment can achieve. The authors show that using VAPs adds minimal overhead to the throughput performance and the delay compared to the conventional approach.

The work [123] investigates a virtualization approach that stores packets from different virtual operators in separate queues until the scheduler sends packets to the wireless interface using a weighted round-robin scheme. Although this approach does not use VAPs, it also uses packet addresses to isolate at the information level.

Similarly, in [124], addressing-based isolation is used to partition the network into different 'slices', which can have different data rates set by the network administrator. Address-based isolation is also used in [126] to create a virtual WIMAX base-station that can offer slice customization to its users. The resource allocation is based on airtime quotas configured at setup.

A method for controlling the air-time usage of different virtual networks is proposed in [125] and [129]. The virtual networks are isolated using virtual access points and the slice identifiers are used to monitor and control air-time usage of each network.

Virtual WiFi [127] proposes a method of having multiple MAC interfaces for virtual machines on the same physical wireless device. This is achieved through giving a different address to each virtual MAC (vMAC) interface and assigning each virtual machine a MAC address. However, it is not clear whether vMACs have any resource limits (apart from the hard performance limits of the device).

The authors of [128] also use VAPs to deploy multiple wireless networks on a shared physical infrastructure. The focus is on open-source virtualization of access points and wireless network interfaces. However, the embedding used is not described.

In [129] virtual networks are isolated through MAC addressing, and each virtual network has its own queue with different parameters. The algorithm to assign wireless airtime to virtual networks is based on a weighted fraction. The authors show that the desired fractions of airtime can be achieved quite closely using their resource allocation algorithm.

The CONTENT project aims to investigate end-to-end virtualization in heterogeneous wireless and optical networks. In the project, wireless virtualization is done by assigning each flow a virtual identification tag, similar to the above approaches [130]. The embedding is performed according to predefined throughput guarantees.

The work [75] developed two algorithms, one a spectrum-level algorithm and the other a flow/information level technique that uses addressing to allocate virtual packets based on traffic load. However only the flow level technique can be

considered virtualization, since the spectrum level technique allocates spectrum also based on traffic load, and thus is not recursive.

3) Space-based Isolation: As mentioned previously, the authors of [122] also examine space-based isolation for virtual networks. Isolation is achieved between nodes on a shared channel by using sufficient spatial separation to avoid interference and by managing the power levels of transmissions. Crucially, this work is concerned with allocating experiments on a grid, and space-based isolation is appropriate in this scenario, since it does not matter *where* experiments are located in space.

4) Discussion: Although time-switching was initially found to be a potential solution for wireless link virtualization at the information level, it soon became apparent that the delays between switching led to significant performance decreases in terms of throughput and especially latency. Another problem with time-switching is the need for precise synchronization between different nodes. The addressing-based isolation method provides many advantages in this regard compared to time-switching.

However, the addressing-based solution could have some overheads compared to time-switching since every node receives all of the traffic, and packets are only filtered out at the receiving nodes. This could lead to reduced battery life, an important point to consider for wireless nodes.

The space-based approach does not seem to offer many advantages compared to the time-switching or the addressing-based approaches, since isolation between different virtual links is very difficult to achieve reliably, and thus this affects the performance of the virtual links.

In terms of the embedding techniques used, the majority of works employ either predefined resource allocation, or round-robin style allocation. Only [64] and [75] propose embedding schemes based on traffic load. It is clear that the focus of information level techniques is on achieving isolation and actually implementing virtualization. A possible explanation of the emphasis on isolation is that these virtualization schemes are used in scenarios where the resource capacity can easily be increased, such as WiFi.

G. Comparison with Existing Surveys

As previously mentioned, there are several surveys concerning wireless link and network virtualization, amongst which the most important are [5], [6], [9]–[11]. We briefly compare the approaches taken by these works to this paper, and highlight how this paper offers additional and complementary insights to the existing literature. Typically such a review would come before performing the survey; however, in this case, now that the survey is complete, it is easier to show how the approach taken in this paper provides additional understanding.

1) Focus on Purpose / Technology: In general, existing surveys define virtualization in terms of its perceived purpose (e.g. in terms of business models, benefits etc.), and classify virtualization techniques by purpose and technology. For example in [11] wireless network virtualization techniques are grouped by access technology, such as IEEE 802.11 (WiFi), 3GPP LTE, IEEE 802.16 (WiMAX), while in [9], [10], and [6] techniques are grouped by technology and purpose – whether techniques have been developed for commercial or experimental use.

Although the above approaches are useful for giving a detailed description of work that has been done to date, it can be hard to identify universal virtualization trends and potential research topics. By taking a purposely technology-agnostic approach based on a strong definition of virtualization, in this work we are able to classify virtualization techniques theoretically. The advantage of this is that it is easier to identify common trends (described in the next section) and propose problematic research areas in a clear and coherent manner. In addition, the technology-agnostic approach has another benefit; it can still be relevant to future technologies when these are developed.

As a specific example, compare the above Table IV with Table I found in [11]. Both tables cover a similar collection of papers (some papers found in [11] do not pass the recursive test, and are omitted), but in the latter table techniques are grouped by technology, whereas in this work techniques are grouped by representation level. In the technology-oriented table, all of the works on LTE ([84], [85], [111], [112]) can be found at the spectrum-level in the other table, and the works on 802.11 ([118], [122], [127], [129]) can be found at the information-level. By focusing on technology, one might wrongly assume that virtualization in 802.11 can only be done through information-level techniques such as traffic shaping, or that virtualization in LTE can only be achieved by spectrum virtualization. However, virtualization can always be done at any representation level; for example, 802.11 virtualization occurs at the spectrum-level in [106]. The *observation* that to date most work in LTE has been at spectrum-level, and most work in WiFi has been at information-level is only possible because of the technology-agnostic approach.

Thus the focus on the purpose/technology of virtualization techniques offers certain insights and enables the detailed description of virtualization techniques, but it can mean that general trends and new possibilities are missed.

2) Lack of Specificity: The theoretical approach underpinning this survey provides a second advantage over existing surveys; it offers a specific language that is clear and succinct, which can lead to new insights about virtualization.

Several of the concepts described in this paper are briefly mentioned in the current literature. For example, the need for isolation in virtualization is mentioned in almost all surveys listed above. However, although isolation is recognised as important, it is not explicitly described. In [6] and [11] isolation is used for the survey tables, although a vast range of different isolation measures are given, among which are: MAC layer, time division/slot, PRBs, packet, spatial, flow, traffic shaping, rate control, slice, sub-channel/sub-carrier. The large number of isolation types does not provide clarity, but instead means that isolation is left vague. By using a theoretical approach in this survey, a clear understanding of isolation is reached, and we can identify the specific isolation types such as frequency, time, etc., even when different terms are used.

Similarly, the concept of 'virtualization levels' can be found in [6], [9], [11] and many other works on virtualization. Again, this concept is insufficiently described and is unclear. Often 'levels' of virtualization are described as spectrum-, infrastructure-, and network-level virtualization. In fact, the first two terms are simply *link* virtualization and *node* virtualization respectively, and the third is the combination of both, i.e. network virtualization. These are not "levels" in any meaningful sense, just different applications of virtualization.

The term 'level' ([6], [11]) is also used to refer to different levels of the protocol stack, or alternatively 'depth of virtualization'/'perspective' ([9], [5]) is used. Some of the levels described are 'flow-level virtualization', 'protocol-perspective', and 'front-end- and spectrum-perspective'. However, there is no clear distinction between any of these levels or method to recognise the virtualization level. In order to be specific and to avoid introducing ambiguity when discussing different levels of representation (i.e. virtualization levels in other works), we started this paper with a theoretical overview of AR theory, which ultimately allows us to be specific when surveying the literature.

Finally, by providing specificity, in this work we are able to derive a test for virtualization, which allows us to distinguish virtualization and non-virtualization techniques with some validity. Without specificity, the classification of works can seem arbitrary, as one does not know the difference between one class and the next.

VII. Observations and Research Questions

A few conclusions can be drawn from our examination and survey of the the literature on wireless link virtualization. In this section, several observations are discussed, and we propose key open research questions regarding wireless link virtualization and wireless network virtualization.

A. Comparison and Analysis of Virtualization Techniques

1) Metrics: Although the benefits of wireless link/network virtualization have been discussed in an abstract manner in several papers already ([2], [4], etc.), little work has been done on measuring these benefits quantitatively. It has been very difficult to quantify the benefits of virtualization directly. Currently, to the best of our knowledge, no performance metrics exist that can measure and compare virtualized wireless networks and non-virtualized wireless networks, and show whether virtualization provides benefits in terms of isolation, flexibility, customizability, robustness, scalability and other measures, In the same vein, little work has been done which compare virtualized networks to non-virtualized networks using existing performance metrics, such as cost, revenue, efficiency, coverage, and capacity.

2) Requirements: Several papers have discussed requirements for wireless link/network virtualization such as [49], [90], [9] and [6], but little work has been done in developing methods to compare how virtualization techniques address these requirements. Some of the requirements listed in these works are subjective and can not be measured easily such as the requirements of allowing heterogeneity and mobility and

providing a generic and modular interface that is technology agnostic. To the best of our knowledge, there are currently no available methods of comparing virtualization techniques in terms of aspects discussed earlier such as isolation, flexibility, etc.

3) Aggregation: Another observation is that most work is currently focused on the sharing (or slicing) of link resources. An interesting research topic would be to investigate whether virtualization can provide advantages to networks through the aggregation of resources, and what applications could benefit from this. In a sense, some existing work such as Carrier Aggregation (CA) [133] could be considered to fall under this heading.

4) Representation Level: A very interesting research topic could be to investigate which representation level(s) would be more suitable for virtualization for different types of wireless networks such as nomadic, mobile and ad-hoc networks. Although this question could not possibly be answered for every wireless network application, it might be possible to explore whether some applications are better suited to spectrum-level or information-level virtualization.

One consideration to bear in mind is that due to the stochastic nature of the wireless channel, it is not possible to guarantee a specific data-rate, unlike a wired connection. This could mean that spectrum-level virtualization can offer some advantages over information-level virtualization in that it allows virtual links to have customizable spectrum-level properties, such as modulation and coding scheme, the assignment of frequencies, and the choice of technology, whereas for information-level virtualization every virtual network uses the same spectrum-level conditions and parameters. Spectrum-level virtualization could also allow spectrum resources to be used more efficiently than information-level virtualization, since wireless links can be optimized to the channel conditions and application requirements. However, the isolation required between different virtual links at the spectrum level is not as straightforward, and might require additional processing and overheads in terms of initialization and management.

Some advantages of information-level virtualization compared to spectrum level virtualization are that it might be easier and less complex. it could also be easier to integrate with wired network virtualization, since packets with VLAN tags coming from a wired virtual network can be forwarded directly from the wireless virtual access point to the nodes. Each virtual link can also easily set its own information-level settings such as data rate, latency, security and authentication, etc.

5) Isolation: Another question which has barely been addressed in the literature is the form of isolation that is most appropriate for different applications and scenarios. The only work we are aware that compares forms of isolation is [122]. From the survey, we found that works at the spectrum-level provide isolation primarily through the frequency dimensions, and sometimes combine this with an addition dimension, while at the information-level the isolation is almost exclusively done in the time dimension.

6) Research Questions: Several questions emerge on these issues such as: what wireless network applications would benefit most from virtualization and which virtualization tech-

niques are most suitable? Are there any applications that could benefit from resource aggregation? Is it possible to develop metrics to quantify virtualization techniques in terms of isolation, efficiency, flexibility, customizability, scalability, and robustness? Can we compare virtualization methods at different levels of representation using these metrics? What forms of isolation are most appropriate?

The questions shown above provide a few examples of the many questions relating to this topic, which could be investigated further and could provide many benefits to wireless networks.

B. Implementation at the Spectrum Level

Currently, the virtualization of wireless links has been implemented at the information level, and is being used commercially with success [134]. However, to the best of our knowledge, the only implementation of a link virtualization scheme at the spectrum level is the Spectrum Virtualization Layer [106]. Virtualizing wireless link resources at the spectrum level could provide many advantages over the information level, since it would allow the users of virtual network resources much greater control, flexibility, and customizability. The electromagnetic spectrum is considered a very expensive and valuable resource, but it remains artificially underused in many cases [135]. An implementation at the spectrum-level would allow spectrum to be used more efficiently and tailored to specific applications, while also maintaining strict isolation between different networks.

However, to date, the majority of work on spectrum level virtualization has been theoretical in nature, and has focused on the embedding problem, that is, the issue of mapping virtual resources to real resources. The isolation problem has received very little attention, and it would be interesting to examine the different types of isolation for spectrum level virtualization, and discover the advantages and disadvantages of each.

1) Frameworks: Several architectures and frameworks for spectrum level virtualization have been proposed such as [2], [20], [136], and [4]. Architectures have often been aimed at specific technologies, such as LTE [84], and are often theoretical in nature, such as [107] and [9, Chapter 6]. Although these architectures have made valuable contributions, to make further progress towards implementation, it could be necessary to consider aspects of virtualization that have received little to no attention, such as the different steps involved in virtualization, rather than only the embedding problem.

2) Purpose of virtual networks: Another observation that most works do not consider is the application and purpose of virtual networks and how to know how many resources they require. Only the works [100] and [99] consider the request strategies of virtual network users. This is a significant limitation since the users requesting virtual networks somehow need to decide how many resources to request from an infrastructure provider and which resources to request. Since virtualization allows resources to be used in a much more flexible manner than in traditional networks, conventional prediction strategies might not be suitable, and analysis should be done to investigate how virtual network users can predict how many and

which resources they require, which request strategies to adopt, and also how to discover and request available resources. Different mechanisms for requesting virtual resources and virtual networks could be examined too.

3) Node and network virtualization: In most of the wireless network virtualization work so far, the focus is on virtualizing the link resources and assuming that the node resources can be virtualized easily. However this might not always be the case. The current work on Network Function Virtualization tackles some of the processing issues and shows that this assumption does not apply. To the best of our knowledge the only works that consider both wireless node and link virtualization are [98] and [80], however for both of these works the wireless link resource allocation does not pass the recursive test, and thus the wireless link aspect of these works is not considered virtualization.

Another aspect that is not yet known to what extent is the virtualization of transmission resources possible, for example recent work on Multiple Input Multiple Output (MIMO) transmission could be considered as the aggregation of several antennas to create a virtual single directed antenna (one-to-many virtualization). It is not known yet whether it is physically possible to share one antenna between multiple virtual entities (many-to-one virtualization).

Building on an implementation of spectrum level link virtualization, it would be very interesting to apply some of the concepts developed in network virtualization and to create wireless virtual *networks*, rather than virtual links only. It might be possible to create wireless network virtualization implementations that allow users to dynamically request wireless virtual networks with specific, customizable requirements.

4) Research Questions: A set of questions on these issues can be formed: What are the differences between isolation methods for spectrum-level virtualization and is it possible to show the advantages and disadvantages of each isolation approach? Is it possible to develop a virtual wireless network implementation at the spectrum-level, that allows users to dynamically request and pay for virtual networks with specific node and link requirements such as locations, processing power, storage, interfacing, latency and capacity requirements? What issues arise when combining node and link virtualization for wireless network virtualization and can we apply methods from network virtualization to overcome these issues? How do virtual resource users know 1) how many and which virtual resources they need, 2) which resources are available, 3) how to request these available resources, and 4) what are the best request mechanisms and strategies to adopt?

C. Discussion

It is clear that there are still many research challenges and open questions on the topics of wireless link virtualization and wireless network virtualization. We have divided these questions into two broad categories above, questions concerning the analysis of wireless network virtualization, and questions regarding the implementation of wireless network virtualization at the spectrum level. Some of these questions might be easier to investigate, while other issues might remain unresolved.

Although developing new metrics for comparing virtualization techniques would be a very interesting challenge, in our opinion it is more important to develop implementations at the spectrum level first. The reasons for this are that 1) Metrics for comparing virtualization techniques, either to examine which technique is most appropriate, or to see whether virtualization offers benefits over traditional networks, are only useful once these techniques exist and are implementable, 2) Implementations at the spectrum level could change the way in which spectrum resources are used, and potentially allow for new functionality and business models, 3) Implementations could aid in answering other questions, for example, issues about scalability, robustness, and flexibility, while additional problems might appear that cannot be known in advance, and finally 4) Implementation at the spectrum level would enable comparison between spectrum and information level wireless link virtualization, and analysis of suitable virtualization techniques for different applications.

VIII. CONCLUSION

In this work our purpose was to provide an overview of the challenges in wireless link virtualization and to consider possible next steps forward. However, before this could be accomplished, it was necessary to clarify several aspects about wireless network virtualization. We first revisited several key concepts and clarified the difference between abstraction and representation. We examined AR theory, and extended it to include the concept of levels of representation. We also showed how representation and instantiation apply to virtualization.

We developed a theory of virtualization to discuss virtualization in a coherent and structured manner. We showed that recursion is a key property of virtualization. We demonstrated how virtualization can be broken down into the isolation and the embedding problems, and provided a new definition of virtualization.

We examined network resource virtualization, which is necessary for network virtualization, starting from processing virtualization and continuing through to link virtualization. Next we described network and wireless network virtualization and showed that before WNV can be done, it is necessary to first develop wireless link virtualization. Thus we performed a survey of existing wireless link virtualization techniques by representation level and isolation method, and described several interesting observations. Finally, we observed several open problems in the area of wireless link virtualization and wireless network virtualization and formed research questions based on these problems.

In future work we hope to examine the steps involved in virtualization and work on a spectrum-level implementation, and possibly develop new metrics for comparing virtualization techniques.

ACKNOWLEDGEMENT

REFERENCES

[1] N. Chowdhury and R. Boutaba, "Network Virtualization: State of the Art and Research Challenges," *Communications Magazine, IEEE*, vol. 47, no. 7, pp. 20–26, Jul. 2009.

[2] J. Sachs and S. Baucke, "Virtual Radio: a Framework for Configurable Radio Networks," *Proceedings of the 4th International ICST Conference on Wireless Internet*, pp. 61–68, 2008.

[3] T. K. Forde, I. Macaluso, and L. E. Doyle, "Exclusive Sharing & Virtualization of the Cellular Network," *IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN 2011*, pp. 337–348, 2011.

[4] A. Wang *et al.*, "Network Virtualization: Technologies, Perspectives, and Frontiers," *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 523–537, 2013.

[5] H. Wen, P. K. Tiwary, and T. Le-Ngoc, "Current Trends and Perspectives in Wireless Virtualization," *International Conference on Selected Topics in Mobile and Wireless Networking, MoWNeT 2013*, pp. 62–67, 2013.

[6] C. Liang and F. R. Yu, "Wireless Network Virtualization: A Survey, Some Research Issues and Challenges," *Communications Surveys & Tutorials, IEEE*, vol. 17, no. 1, pp. 358–380, 2014.

[7] Liang, Chengchao and Yu, F Richard, "Wireless Virtualization for Next Generation Mobile Cellular Networks," *IEEE Wireless Communications*, no. 1, pp. 61–69, Feb. 2015.

[8] A. Nakao, "Network Virtualization as Foundation for Enabling New Network Architectures and Applications," *IEICE transactions on communications*, vol. 93, no. 3, pp. 454–457, 2010.

[9] H. Wen, P. K. Tiwary, and T. Le-Ngoc, *Wireless Virtualization*. Springer, 2014.

[10] M. Yang *et al.*, "Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 4–18, 2015.

[11] M. Richart *et al.*, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, 2016.

[12] D. C. Horsman, "Abstraction/Representation Theory for Heterotic Physical Computing," *Philosophical Transactions of the Royal Society of London A*, vol. 373, no. 2046, 2015.

[13] C. Horsman *et al.*, "When does a physical system compute?" in *Proceedings of the Royal Society of London A*, vol. 470, no. 2169, 2014.

[14] Data In The Computer. [Online]. Available: http://homepage.cs.uri.edu/book/binary_data/binary_data.htm (last accesed on 07/15/2016)

[15] A. S. Tanenbaum and J. R. Goodman, *Structured Computer Organization*, 4th ed. Prentice Hall PTR, 1998.

[16] R. P. Goldberg, "Architecture of Virtual Machines," in *Proceedings of the Workshop on Virtual Computer Systems*, 1973, pp. 74–112.

[17] P. J. Denning, "Virtual Memory," *ACM Comput. Surv.*, vol. 2, no. 3, pp. 153–189, 1970.

[18] G. J. Popek and R. P. Goldberg, "Formal Requirements for Virtualizable Third Generation Architectures," *Communications of the ACM*, vol. 17, no. 7, pp. 412–421, Jul. 1974.

[19] G. Belpaire and N.-T. Hsu, "Formal Properties of Recursive Virtual Machine Architectures," in *Proceedings of the Fifth ACM Symposium on Operating Systems Principles*, 1975, pp. 89–96.

[20] K.-m. Park and C.-k. Kim, "A Framework for Virtual Network Embedding in Wireless Networks," in *Proceedings of the 4th International Conference on Future Internet Technologies*, 2009, pp. 5–7.

[21] T. Kilburn *et al.*, "One-Level Storage System," *Electronic Computers, IRE Transactions on*, vol. 11, no. 2, pp. 223–235, 1962.

[22] J. Fotheringham, "Dynamic Storage Allocation in the Atlas Computer, Including an Automatic Use of a Backing Store," *Communications of the ACM*, vol. 4, no. 10, pp. 435–436, 1961.

[23] H. C. Lauer and D. Wyeth, "A Recursive Virtual Machine Architecture," in *Proceedings of the Workshop on Virtual Computer Systems*. New York, NY, USA: ACM, 1973, pp. 113–116.

[24] P. Checkland, *Systems Thinking, Systems Practice*. John Wiley and Sons, 1981.

[25] "An Experimental Time-Sharing System, author=Corbató, Fernando J and Merwin-Daggett, Marjorie and Daley, Robert C, booktitle=Proceedings of the May 1-3, 1962, spring joint computer conference, pages=335–344, year=1962, organization=ACM."

[26] R. J. Creasy, "The Origin of the VM/370 Time-sharing System," *IBM Journal of Research and Development*, vol. 25, no. 5, pp. 483–790, 1981.

[27] Bob Bremer. Origins of TimeSharing. [Online]. Available: http://www.bobbemer.com/TIMESHAR.HTM (last accesed on 07/15/2016)

[28] J. M. McKinney, "A Survey of Analytical Time-Sharing Models," *ACM Comput. Surv.*, vol. 1, no. 2, pp. 105–116, Jun. 1969.

[29] L. Kleinrock, "A Continuum of Time-sharing Scheduling Algorithms," in *Spring Joint Computer Conference, Proceedings of the*, 1970, pp. 453–458.

[30] S. F. Yashkov and A. S. Yashkova, "Processor Sharing: A Survey of the Mathematical Theory," *Automation and Remote Control*, vol. 68, no. 9, pp. 1662–1731, 2007.

[31] Timesharing: A Solution to Computer Bottlenecks. [Online]. Available: https://www.youtube.com/watch?v=Q07PhW5sCEk (last accesed on 07/15/2016)

[32] D. Sayre, "Is Automatic "Folding" of Programs Efficient Enough to Displace Manual?" *Communications of the ACM*, vol. 12, no. 12, pp. 656–660, 1969.

[33] S. Raina, "Virtual Shared Memory: A Survey of Techniques and Systems," *University of Bristol, Bristol, UK*, 1992.

[34] J. Protic, M. Tomasevic, and V. Milutinović, *Distributed Shared Memory: Concepts and Systems*. John Wiley & Sons, 1998, vol. 21.

[35] F. J. Corbató and V. A. Vyssotsky, "Introduction and Overview of the Multics System," in *Proceedings of the 1965 Fall Joint Computer Conference, Part I*, 1965, pp. 185–196.

[36] A. J. Critcklow, "Generalized Multiprocessing and Multiprogramming Systems," in *Proceedings of the November 12-14, 1963, Fall Joint Computer Conference*, 1963, pp. 107–126.

[37] J. P. Buzen and U. O. Gagliardi, "The Evolution of Virtual Machine Architecture," *Proceedings of the June 48 1973 national computer conference and exposition on AFIPS 73*, pp. 291–299, 1973.

[38] R. P. Goldberg, "Architectural Principles for Virtual Computer Systems," DTIC Document, Tech. Rep., 1973.

[39] ——, "Survey of Virtual Machine Research," *Computer*, vol. 7, no. 6, pp. 34–45, Jun. 1974.

[40] I. Ahmad *et al.*, "An Analysis of Disk Performance in VMware ESX Server Virtual Machines," in *Workload Characterization, 2003. WWC-6. 2003 IEEE International Workshop on*, Oct. 2003, pp. 65–76.

[41] R. Rose, "Survey of System Virtualization Techniques," Oregon State University, Tech. Rep., 2004.

[42] M. F. Bari *et al.*, "Data Center Network Virtualization: A Survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 909–928, Feb. 2013.

[43] S. N. T.-c. Chiueh and S. Brook, "A Survey on Virtualization Technologies," *RPE Report*, pp. 1–42, 2005.

[44] A. Blenk *et al.*, "Survey on Network Virtualization Hypervisors for Software Defined Networking," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 655–685, First Quarter 2016.

[45] D. Abramson, "Intel Virtualization Technology For Directed I/O," *Intel Technology Journal*, vol. 10, no. 3, pp. 179–192, 2006.

[46] B. Zhang *et al.*, "A Survey on I/O Virtualization and Optimization," in *ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual*, Jul. 2010, pp. 117–123.

[47] A. Rybczynski, "X.25 Interface and End-to-End Virtual Circuit Service Characteristics," *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 500–510, Apr. 1980.

[48] C. Metz, "The Latest in Virtual Private Networks: Part I," *IEEE Internet Computing*, vol. 7, no. 1, pp. 87–91, Jan. 2003.

[49] J. Carapinha, J. Jiménez, and C. E. Vargas, "Network Virtualization - a View from the Bottom," *Business*, pp. 73–80, 2009.

[50] L. Peterson *et al.*, "A Blueprint for Introducing Disruptive Technology into the Internet," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 59–64, 2003.

[51] J. S. Turner and D. E. Taylor, "Diversifying the Internet," in *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, vol. 2, Dec. 2005, pp. 6 pp.–760.

[52] T. Anderson *et al.*, "Overcoming the Internet Impasse through Virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[53] D. Andersen *et al.*, "Resilient Overlay Networks," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, 2001, pp. 131–145.

[54] J. Touch *et al.*, "A Virtual Internet Architecture," *ISI Technical Report ISI-TR-2003-570*, pp. 73–80, 2003.

[55] N. Feamster, L. Gao, and J. Rexford, "How to lease the Internet in your spare time," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 61–64, 2007.

[56] GENI Concepts. [Online]. Available: http://groups.geni.net/geni/wiki/GENIConcepts (last accesed on 07/15/2016)

[57] Virtualisation Approach: Evaluation and Integration. [Online]. Available: http://www.4ward-project.eu/index.php?s=file_download&id=95 (last accesed on 07/15/2016)

[58] N. M. K. Chowdhury and R. Boutaba, "A Survey of Network Virtualization ," *Computer Networks*, vol. 54, no. 5, pp. 862 – 876, 2010.

[59] A. Fischer *et al.*, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[60] B. A. A. Nunes *et al.*, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, Third 2014.

[61] D. Kreutz *et al.*, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[62] R. Mijumbi *et al.*, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[63] B. Han *et al.*, "Network Function Virtualization: Challenges and Opportunities for Innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, Feb 2015.

[64] R. Chandra and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, 2004, pp. 882–893.

[65] N. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual Network Embedding with Coordinated Node and Link mapping," in *INFOCOM 2009, IEEE*, 2009, pp. 783–791.

[66] X. Lu *et al.*, "An Elastic Resource Allocation Algorithm Enabling Wireless Network Virtualization," *Wireless Communications and Mobile Computing*, vol. 15, no. 2, pp. 295–308, 2015.

[67] P. Lv, X. Wang, and M. Xu, "Virtual Access Network Embedding in Wireless Mesh Networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1362 – 1378, 2012.

[68] X. Xu *et al.*, "Resource Pooling for Frameless Network Architecture with Adaptive Resource Allocation," *Science China Information Sciences*, vol. 56, no. 2, pp. 1–12, 2013.

[69] ——, "SDN Based Next Generation Mobile Network with Service Slicing and Trials," *China Communications*, vol. 11, no. 2, pp. 65–77, Feb. 2014.

[70] G. Liu *et al.*, "Distributed Resource Allocation in Full-Duplex Relaying Networks with Wireless Virtualization," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec. 2014, pp. 4959–4964.

[71] ——, "In-Band Full-Duplex Relaying: A Survey, Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 500–524, 2015.

[72] M. Kamel, L. Bao Le, and A. Girarg, "LTE Multi-Cell Dynamic Resource Allocation for Wireless Network Virtualization," in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*, Mar. 2015, pp. 966–971.

[73] G. Zhang *et al.*, "Virtual Resource Allocation for Wireless Virtualization Networks Using Market Equilibrium Theory," in *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, Apr. 2015, pp. 366–371.

[74] ——, "A Bargaining Game Theoretic Method for Virtual Resource Allocation in LTE-Based Cellular Networks," *Science China Information Sciences*, vol. 58, no. 12, pp. 1–9, 2015.

[75] Z. Feng *et al.*, "An Effective Approach to 5G: Wireless Network Virtualization," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 53–59, Dec. 2015.

[76] Y. Cai *et al.*, "Software Defined Device-to-Device (D2D) Communications in Virtual Wireless Networks with Imperfect Network State Information (NSI)," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[77] L. Chen *et al.*, "Distributed Virtual Resource Allocation in Small Cell Networks with Full Duplex Self-backhauls and Virtualization," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[78] C. Liang and F. R. Yu, "Distributed Resource Allocation in Virtualized Wireless Cellular Networks Based on ADMM," in *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, Apr. 2015, pp. 360–365.

[79] S. Khatibi and L. M. Correia, "A Model for Virtual Radio Resource Management in Virtual RANs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 68, 2015.

[80] M. M. Rahman, C. Despins, and S. Affes, "Design Optimization of Wireless Access Virtualization Based on Cost QoS Trade-Off Utility Maximization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 9, pp. 6146–6162, Sept 2016.

[81] F. Fu and U. C. Kozat, "Wireless Network Virtualization as a Sequential Auction Game," *Proceedings - IEEE INFOCOM*, pp. 1–9, 2010.

[82] F. Fu and U. Kozat, "Stochastic Game for Wireless Network Virtualization," *Networking, IEEE/ACM Transactions on*, vol. 21, no. 1, pp. 84–97, 2012.

[83] M. I. Kamel, L. B. Le, and A. Girard, "LTE Wireless Network Virtualization: Dynamic Slicing via Flexible Scheduling," in *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, Sep. 2014, pp. 1–5.

[84] Y. Zaki *et al.*, "LTE Mobile Network Virtualization Exploiting Multiplexing and Multi-User Diversity Gain," *Mobile Networks and Applications*, vol. 16, no. 4, pp. 424–432, 2011.

[85] ——, *Mobile Networks and Management: Second International ICST Conference, MONAMI 2010, Santander, Spain, September 22-24, 2010, Revised Selected Papers*. Springer, 2011, ch. A Novel LTE Wireless Virtualization Framework, pp. 245–257.

[86] L. Zhao *et al.*, "LTE Virtualization: From Theoretical Gain to Practical Solution," in *Teletraffic Congress (ITC), 2011 23rd International*, Sep. 2011, pp. 71–78.

[87] M. Li *et al.*, "Investigation of Network Virtualization and Load Balancing Techniques in LTE Networks," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, May 2012, pp. 1–5.

[88] B. Liu and H. Tian, "A Bankruptcy Game-Based Resource Allocation Approach among Virtual Mobile Operators," *Communications Letters, IEEE*, vol. 17, no. 7, pp. 1420–1423, 2013.

[89] J. S. Panchal, R. D. Yates, and M. M. Buddhikot, "Mobile Network Resource Sharing Options: Performance Comparisons," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4470–4482, Sep. 2013.

[90] M. Kalil, A. Shami, and Y. Ye, "Wireless Resources Virtualization in LTE Systems," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, 2014, pp. 363–368.

[91] M. Kalil *et al.*, "Efficient Low-Complexity Scheduler for Wireless Resource Virtualization," *IEEE Wireless Communications Letters*, vol. 5, no. 1, pp. 56–59, Feb 2016.

[92] A. Moubayed, A. Shami, and H. Lutfiyya, "Wireless Resource Virtualization With Device-to-Device Communication Underlaying LTE Network," *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 734–740, Dec 2015.

[93] R. Kokku *et al.*, "NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 5, pp. 1333–1346, 2012.

[94] ——, "CellSlice: Cellular Wireless Resource Slicing for Active RAN Sharing," in *Communication Systems and Networks (COMSNETS), 2013 Fifth International Conference on*, 2013, pp. 1–10.

[95] X. Costa-Perez *et al.*, "Radio Access Network Virtualization for Future Mobile Carrier Networks," *Communications Magazine, IEEE*, vol. 51, no. 7, pp. 27–35, 2013.

[96] J. Wei *et al.*, "Pricing-Based Power Allocation in Wireless Network Virtualization: A Game Approach," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, Aug. 2015, pp. 188–193.

[97] X. Wang, P. Krishnamurthy, and D. Tipper, "A Collaborative Spectrum-Sharing Framework for LTE Virtualization," in *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, Oct 2015, pp. 260–269.

[98] G. D. Stasi, S. Avallone, and R. Canonico, "Virtual network embedding in wireless mesh networks through reconfiguration of channels," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2013, pp. 537–544.

[99] Q. Zhu and X. Zhang, "Game-Theory Based Power and Spectrum Virtualization for Maximizing Spectrum Efficiency over Mobile Cloud-Computing Wireless Networks," in *Information Sciences and Systems (CISS), 2015 49th Annual Conference on*, Mar. 2015, pp. 1–6.

[100] ——, "Bayesian-Game Based Power and Spectrum Virtualization for Maximizing Spectrum Efficiency over Mobile Cloud-Computing Wireless Networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, Apr. 2015, pp. 378–383.

[101] B. Fan, H. Tian, and B. Liu, "Game theory based power allocation in LTE air interface virtualization," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, March 2015, pp. 972–976.

[102] K. Zhu and E. Hossain, "Virtualization of 5G Cellular Networks as a Hierarchical Combinatorial Auction," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2640–2654, Oct 2016.

[103] H. Ahmadi *et al.*, "Substitutability of Spectrum and Cloud-based Antennas in Virtualised Wireless Networks," *IEEE Wireless Communications*, 2016.

[104] ——, "Virtualization of Spatial Streams for Enhanced Spectrum Sharing," in *Global Telecommunications Conference (GLOBECOM 2016), 2016 IEEE*, 2016.

[105] N. Bhushan *et al.*, "Network densification: the dominant theme for wireless evolution into 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, February 2014.

[106] K. Tan *et al.*, "Enable Flexible Spectrum Access with Spectrum Virtualization," in *Dynamic Spectrum Access Networks (DYSPAN), 2012 IEEE International Symposium on*, 2012, pp. 47–58.

[107] M. Yang *et al.*, "OpenRAN: A Software-Defined Ran Architecture via Virtualization," *ACM SIGCOMM computer communication review*, vol. 43, no. 4, pp. 549–550, Aug. 2013.

[108] S. Paul and S. Seshan, "Technical Document on Wireless Virtualization," *GENI: Global Environment for Network Innovations,, Tech. Rep*, Sep. 2006.

[109] M. Yang *et al.*, "Opportunistic Spectrum Sharing Based Resource Allocation for Wireless Virtualization," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*. IEEE, Jul. 2013, pp. 51–58.

[110] ——, "Opportunistic Sharing Scheme for Spectrum Allocation in Wireless Virtualization," *Soft Computing"*, vol. 18, no. 9, pp. 1685–1696, 2014.

[111] ——, "Karnaugh-Map like Online Embedding Algorithm of Wireless Virtualization," in *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, 2012, pp. 594–598.

[112] J. van de Belt, H. Ahmadi, and L. Doyle, "A Dynamic Embedding Algorithm for Wireless Network Virtualization," in *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, Sep. 2014, pp. 1–6.

[113] J. van de Belt *et al.*, "A Prioritised Traffic Embedding Mechanism Enabling a Public Safety Virtual Operator," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*, Sep. 2015, pp. 1–5.

[114] F.-T. Hsu and C.-H. Gan, "Resource Allocation with Spectrum Aggregation for Wireless Virtual Network Embedding," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*, Sep. 2015, pp. 1–5.

[115] G. Lu *et al.*, "A Dynamic Allocation Algorithm for Physical Carrier Resource in BBU Pool of Virtualized Wireless Network," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on*, Sep. 2015, pp. 434–441.

[116] X. Zhang *et al.*, "Efficient Resource Allocation for Wireless Virtualization using Time-Space Division," in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2012, pp. 59–64.

[117] D. Raychaudhuri *et al.*, "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3, Mar. 2005, pp. 1664–1669.

[118] G. Smith *et al.*, "Wireless Virtualization on Commodity 802.11 Hardware," in *Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, 2007, pp. 75–82.

[119] H. Coskun, I. Schieferdecker, and Y. Al-Hazmi, "Virtual WLAN: Going Beyond Virtual Access Points," *Electronic Communications of the EASST*, vol. 17, 2009.

[120] Y. Al-Hazmi and H. de Meer, "Virtualization of 802.11 Interfaces for Wireless Mesh Networks," in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, Jan. 2011, pp. 44–51.

[121] P. Lv *et al.*, "Multicast Service-Oriented Virtual Network Embedding in Wireless Mesh Networks," *IEEE Communications Letters*, vol. 16, no. 3, pp. 375–377, March 2012.

[122] R. Mahindra *et al.*, "Space Versus Time Separation for Wireless Virtualization on an Indoor Grid," in *Next Generation Internet Networks, 2008*, Apr. 2008, pp. 215–222.

[123] S. Perez, J. M. Cabero, and E. Miguel, "Virtualization of the Wireless Medium: A Simulation-Based Study," in *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, Apr. 2009, pp. 1–5.

[124] R. Sherwood *et al.*, "Flowvisor: A Network Virtualization Layer," *OpenFlow Switch Consortium, Tech. Rep*, pp. 1–13, 2009.

[125] G. Bhanage *et al.*, "SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec. 2010, pp. 1–6.

[126] ——, "Virtual Basestation: Architecture for an Open Shared WiMAX Framework," in *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized iIfrastructure Systems and Architectures*, 2010, pp. 1–8.

[127] L. Xia *et al.*, "Virtual WiFi: Bring Virtualization from Wired to Wireless," *ACM SIGPLAN Notices*, vol. 46, no. 7, pp. 181–192, 2011.

[128] G. Aljabari and E. Eren, "Virtualization of Wireless LAN Infrastructures," in *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on*, vol. 2, Sep. 2011, pp. 837–841.

[129] K. Nakauchi, Y. Shoji, and N. Nishinaga, "Airtime-Based Resource Control in Wireless LANs for Wireless Network Virtualization," in *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*, Jul. 2012, pp. 166–169.

[130] K. Katsalis *et al.*, "Wireless Network Virtualization: The CONTENT Project Approach," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on*, Dec. 2014, pp. 90–94.

[131] T. Spyropoulos, S. Fdida, and S. Kirkpatrick, "Future Internet: Fundamentals and Measurement," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 2, pp. 101–106, 2007.

[132] B. Aboba, "Virtual Access Points," IEEE, Tech. Rep., Mar. 2003.

[133] H. Lee, S. Vahid, and K. Moessner, "A Survey of Radio Resource Management for Spectrum Aggregation in LTE-Advanced," *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 745–760, Second 2014.

[134] Windows 7 adds Virtual Wifi Technology from Microsoft Research. [Online]. Available: http://www.istartedsomething.com/20090516/windows-7-native-virtual-wifi-technology-microsoft-research// (last accesed on 07/15/2016)

[135] P. J. Weiser, *The Untapped Promise of Wireless Spectrum*. The Brookings Institution, Jul. 2008.

[136] M. Hoffmann and M. Staufer, "Network Virtualization for Future Mobile Networks: General Architecture and Applications," in *Communications Workshops (ICC), 2011 IEEE International Conference on*, Jun. 2011, pp. 1–5.