End-to-end Virtual Resource Management across Heterogeneous Networks and Services

(EVANS)

FP7- PIRSES-GA-2010-269323

Start date of the project: 1 May 2011

Duration: 36 months

DELIVERABLE D3.1

Intermediate Report on Vertical Management of Virtualised Resources

Date of preparation: 28 September 2012

Deliverable Leading Partner: BUPT

Contributing Author(s): Zheng Hu(ed.) & Ping Zhang & Hui Tian & Ahmad Umair Mian & Wang Zhibo (BUPT), Ke Xu (THU), Kun Yang (UEssex), Juan-Luis Gorricho & Joan Serrat (UPC), Ebrahim Ghazisaeedi & Ning Wang (UniS), Yan Zhang (SRL).

Consortium:

Partner name	Partner short name	Country
University of Essex	UEssex	UK
Universitat Politècnica de Catalunya	UPC	Spain
Simula Research Laboratory	SRL	Norway
University of Surrey	UniS	UK
Tsinghua University	THU	China
Beijing University of Post and Telecommunications	BUPT	China

Document location: http://www.fp7-evans.eu/evans_wiki

Project web site: http://www.fp7-evans.eu

Table of Contents

Executive Summary 5

1. Introduction 6

2. Virtualized Resource Management in Wired Networks 7

2.1. Research Issues in Wired Network Virtualization 7

2.2. Resource Allocation and Scheduling 8

2.2.1. Problem statement 10

2.2.2. Proposed solution 11

2.2.3. Work in progress 12

2.3. Energy Efficient Splitting 12

2.3.1. Overview 12

2.3.2. Problem formulation 14

2.3.3. Proposed Algorithm 15

3. Virtualized Resource Management in Wireless Networks 18

3.1. Research Challenges in Wireless Network Virtualization 18

3.2. Resource Allocation for Wireless Network Virtualization 19

3.2.1. Introduction 19

3.2.2. Problem Clarification 21

3.2.3. Proposed Elastic Wireless ResourceVirtualization Algorithm 22

3.2.4. Performance Evaluation and Analysis 27

3.2.5. Summary 33

3.3. Radio Resource Management for Wireless Network Virtualization 33

4. Conclusions 35

References 35

This document has been produced in the context of the EVANS Project. The EVANS Project is part of the European Community's Seventh Framework Program and is as such funded by the European Commission.

All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Executive Summary

The aim of this document is to provide an intermediate report about the research activities that have taken place in WP3 of the EVANS project. This WP is concerned about the vertical management of the virtualized network resources, which is more a concern of physical network providers or network infrastructure providers. This is about resource management of individual network domains or about intra-domain resource management issues. Therefore, two types of networks are considered in this deliverable: wired core networks and wireless access networks. In each type of networks both static and dynamic resource allocation and management, which represents the two tasks identified in this WP respectively, have been researched into.

The virtualized resource management issues related to wired and wireless access networks have been discussed in this document. In this regard, the resource scheduling and allocation problem in virtualized wired access networks has been formulated and possible solutions have been presented. Another important problem related to sleeping link optimizations in wired virtualized environment is discussed. The objective is to minimize the power consumption in the physical network, while still satisfying bandwidth demands from individual virtual networks (VNs) on the top. The solution for this problem is presented as the energy efficient splitting of physical links during peak and off peak hours.

The resource scheduling problem in virtualized wireless networks is differentiated from the virtualized wired networks and an algorithm for joint network virtualization and resource allocation of IEEE 802.16 wireless networks is formulated, which not only provides network virtualization (isolation) but also achieves network resource efficiency. Finally the virtualizing radio resources for LTE networks are discussed and random access model for virtualized radio resources is presented.

1. Introduction

There are two types of important stakeholders in the Internet business market: infrastructure providers (InP) that own and manage the physical network infrastructure, and service providers (SP) that provide end-to-end services to end users without necessarily owning any physical infrastructure. Instead, SPs may "rent" network resources from the underlying InPs according to their specific business and service plans. In virtualised networks, an SP typically creates its own virtual networks by "concatenating" the rented (virtual) resources from multiple InPs in order to offer Internet-wide services. On the other hand, an InP needs to concern how to optimally slice its resources, for instance bandwidth, CPU time, memory etc, to various requesting SPs, such that the overall infrastructure resources can be efficiently allocated for maximising its own profits. Therefore, two orthogonal dimensions of management tasks in a virtualised network environment can be envisioned, as depicted in Figure 1.

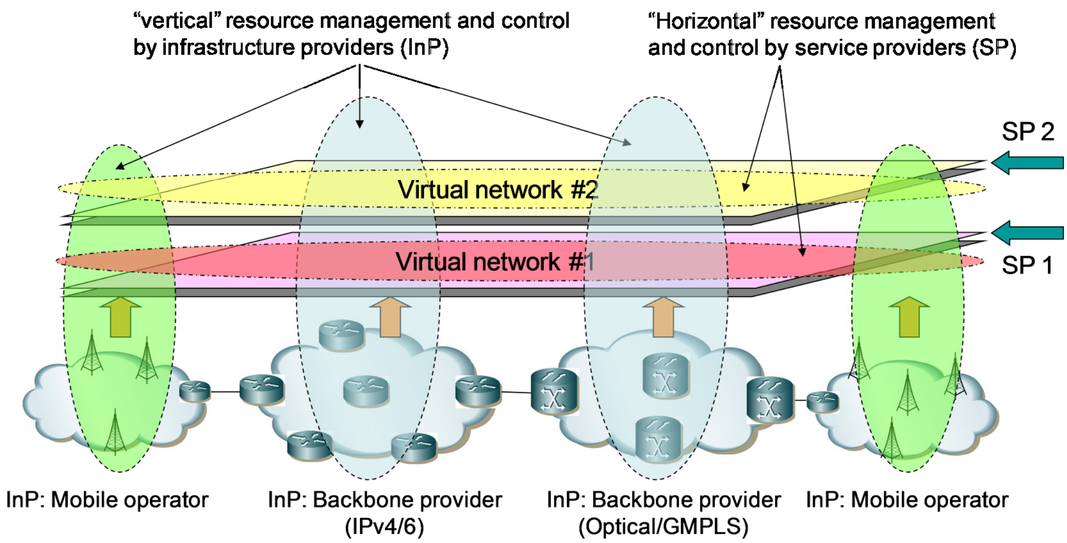


Figure 1. Two Dimensions of Management of Virtualised Networks

Firstly, an InP needs to manage its own physical resources, which involves tasks such as how to describe the physical resources, how to slice them, how to handle incoming resource requests from heterogeneous SPs and allocate virtual resources in a cost-efficient way, etc. This project names this type of management as *vertical* resource management for easy reference. Another dimension of network management is how an SP manages and controls its virtual network resources which are rented from multiple heterogeneous InPs in order to offer its specific services across the corresponding geographical area. This type of management is called *horizontal* resource management, in this project. This deliverable D3.1, which falls into WP3, deals with vertical resource management within individual InPs. Its counterpart in WP4, namely, D4.1, deals with horizontal resource management.

This document records the project's progress regarding investigations into management issues that need to be carried out on the InP side in order to provide network infrastructure support to various services, for instance computation-intensive services such as grid/cloud computing applications and bandwidth-hungry services such as content delivery, as identified in Deliverable D2.1. This WP addresses intra-domain issues with each domain owned by one InP. A domain, or an InP, can take be a wired network provider or a wireless mobile network provider. The former is necessary for the core networks that constitute the Internet backbone, whereas the latter is for various broadband wireless network technologies for mobile access in addition to the conventional fixed access. In order to provide an end-to-end service solution to end users, wireless mobile access network technologies have to be considered as an increasing number of users use mobile devices to get access to the Internet. The EVANS project aims to create a network virtualisation environment over a fully heterogeneous network infrastructure and to provide an integrated network management system across different types of network platforms.

Therefore, the document is organized according to network types: wired networks and wireless networks, as presented in Section 2 and Section 3 of this document respectively. Within each type of network, both static (e.g., at the subscription level of virtual resource requesting) and dynamic (after the invocation of resource leasing)aspects of the management system are described as are represented by Task 3.1 and Task 3.2 respectively in the DoW (Description of Work).

2. Virtualized Resource Management in Wired Networks

2.1. Research Issues in Wired Network Virtualization

In the network virtualization environment, both virtual nodes and links in the VN should be assigned to a specific set of physical nodes and physical paths in the substrate network, with certain constraints on virtual and physical components, which is known as the VN embedding problem. Here virtual network embedding problem across multiple physical network domains (i.e., InPs) has been considered, which will address interactions among multiple domains and facilitate provisioning of the end-to-end virtual networks.

In our research, we consider three parties in the business model, including virtual network user, virtual network provider (VNP) and infrastructure provider (InP). For each virtual network, the VNP is interested in how to get VN's requirements satisfied while minimizing the embedding expenditure. To this end, VNP is desirable to get more detailed resource information (e.g., available node/link capacity) in each substrate network domain. For instance, since the price of intra-domain links is usually much cheaper than that of inter-domain ones, therefore, VNP prefers to embed as many virtual nodes as possible within the same domain that can be interconnected by intra-domain links.

However, InPs are traditionally reluctant to have their private network topologies exposed, especially to their competitors. Each InP is willing to take virtual network requests and then make independent embedding decisions with as little information shared as possible.

To address this tussle among VNP and InPs, we define a reasonable information sharing scheme, from which each party involved can benefit while maintaining the core commercial secrets of InPs. Then an end-to-end virtual network can be embedded by the following three steps:

1) VN Request Creation: in order to deliver end-to-end services, a VN user needs to design a virtual topology that will cover areas where their potential customers are located or connect their geographically distributed private networks together. Therefore, an end-to-end virtual path usually starts from and terminates at the access network. A virtual network request is generated by indicating the topology attributes, including node requirements (i.e., expected location and capacity) and link requirements (i.e., capacity).

2) VN Request Decomposition: upon receiving a VN request from VN user, VNP first conducts node pre-mapping that aims at finding a set of candidate substrate nodes for each virtual node. Then VNP should identify a path for each virtual link, which will minimize the provisioning cost of the whole VN request. Finally, a VN request can be decomposed into several sub-VN requests, each of which corresponds to a specific substrate domain.

3) VN Embedding: after receiving the sub-VN requests from VNP, each InP involved in will independently get its portion embedded according to its own strategies, such as minimizing the embedding cost, performing load balancing across its substrate, and so on.Since the general embedding problem is computationally intractable, we will resort to heuristic solutions to derive a practical virtual network embedding algorithm.

2.2. Resource Allocation and Scheduling

The resource allocation problem in virtual networks is concerned with an efficient embedding of virtual nodes and links to substrate nodes and links. In the figure below, we represent two virtual networks sharing resources of a substrate network. In this scenario, we consider two resource types: CPU Capacity for the nodes and Link Bandwidth for the links. The figure is a representation of the network at a given time, when VN 1 has a ring topology and VN 2 has a star topology. In the figure, the resources are represented in terms of units. 100 CPU Units could represent 3.5GHz, while 100 Link bandwidth units could represent 1.0Gbps. We require that each substrate node or link has a unique identifier.

The virtual networks determine their own topologies based on their demands as well as availability of substrate resources. We can note that for example while VN 1 actually has a topology with three nodes, it actually also uses three other “hidden nodes”, which should also have capacity to forward its traffic. In the above case, we use a case where the number of CPU units for forwarding is equal to the number of Link units being forwarded (this is based on assuming packets of size 1400Bytes and a requirement of 40,000 cycles to process each packet).

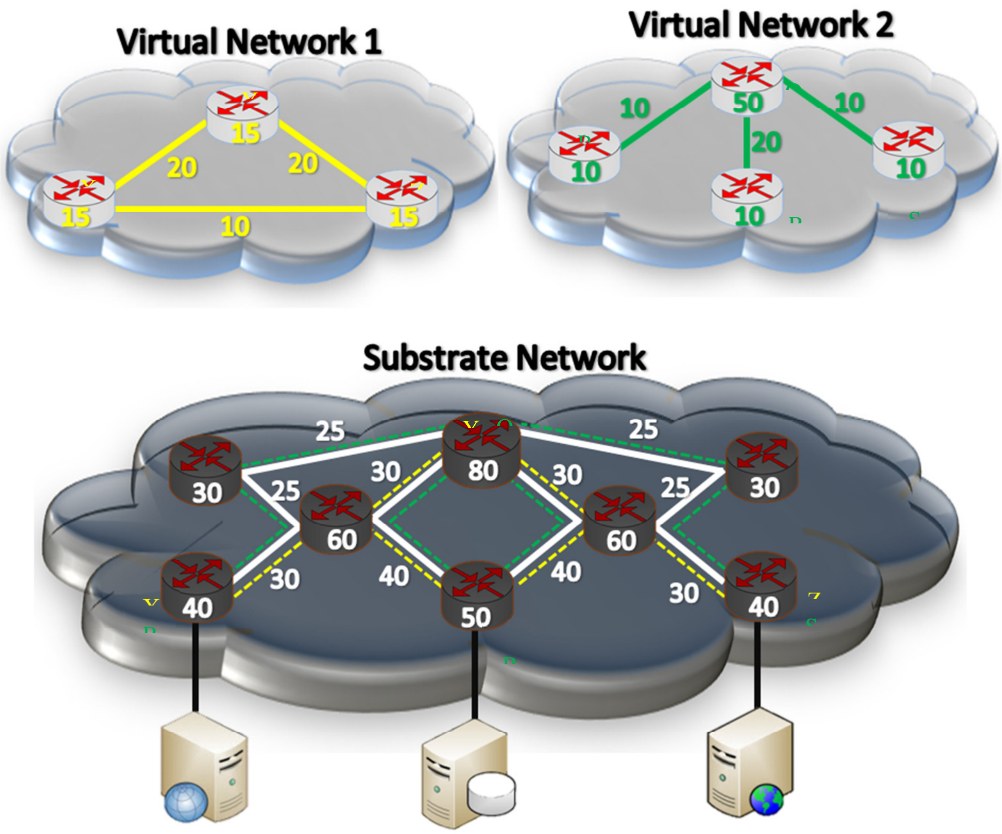


Figure 2. Resource allocation in wired virtualized networks

The virtual networks continuously receive traffic from users. The user requests specify BW requirements and destination. Based on this information and the substrate network status, the following situations may arise:

- Using the same network topology, the VN sends a request for resources to the SN,
- If this request cannot be supported by the substrate network considering current topology, the VN may change topology to use other available links and/or nodes,
- If the change in topology cannot be supported by available substrate resources, the VN through admission control decide to reject the user request. (When we start looking at cooperation and negotiation, this will be a point to consider negotiating with other VNs that are occupying resources that are of interest so they can free up these resources).

We also can also consider possibilities for path splitting, in which case a given traffic flow can be routed over more than one parallel substrate link.

We can therefore represent a virtual network as a graph $G_V = (N_V, L_V)$, where N_v is the set of virtual nodes and L_v is the set of virtual links. The k^{th} virtual node $n_V^K \in N_V$, has an associated CPU demand $D(n_V^K)$, while the virtual link that connects nodes *a* and *b*, $l_V^{(a,b)} \in L_V$ has an associated link bandwidth demand $D(l_V^{(a,b)})$. In the same way, a substrate network can be represented as a graph $G_S = (N_S, L_S)$, where N_s is the set of substrate nodes and L_s is the set of substrate links. The k^{th} substrate node $n_S^K \in N_S$, has an associated CPU capacity $C(n_S^K)$ which represents the available (free) node capacity, while the substrate link that connects nodes *a* and *b*, $l_S^{(a,b)} \in L_S$ has an associated (available) link bandwidth $C(l_S^{(a,b)})$. Therefore, the virtual network mapping problem is in effect a mapping of the graph G_V to G_S. Ideally, we can split the mapping problem into two stages: first mapping the nodes, and then mapping the links (or may be in the opposite order? In any case, the mapping of anyone of them could be highly dependent on the other).

2.2.1. Problem statement

Network Virtualization – which enables the building of multiple virtual networks over a shared physical network – has received a lot of attention from both academia and industry. One of the challenges to Virtualization is efficient resource allocation. Due to the problem complexity, solutions are usually static and based on multiple simplifying assumptions such as availability of unbounded resources in the substrate network and others. In this paper, we propose to apply techniques from Artificial Intelligence to the resource allocation problem. Our objective is to derive a solution that is dynamic and online (including topology optimization); to allow for cooperation and negotiation between substrate and virtual networks; and to achieve an Autonomic Management solution, implying that the networks are self-configuring, self-optimizing, self-healing and user context aware.

The problem we face in this research work is the assignment of physical resources to given virtual network topologies with constraints. The problem statement and the challenges of its solution are clearly exposed in [1-3]. In summary, given a number of VN requests (the request specifies the VNs topology) that can be known in advance or appearing randomly in time, the target is to assign physical resources (nodes and links) of a given substrate network topology to satisfy the VNs requirements and at the same time fulfilling some goals or constraints. Such goals are referred either to the virtual or to the physical network and in the earliest works consisted for instance in the number of virtual nodes assigned to a physical node, number of virtual links assigned to a physical link [1], node CPU usage [2], virtual links capacity [3] and others. The problem has been treated analytically transforming the goals to be achieved in a maximization/minimization of a cost function. But this generally leads to a NP-complete problem that the only way to solve it is by heuristics.

2.2.2. Proposed solution

To achieve the objectives of our work, we propose to apply Reinforcement Learning (RL) techniques, in a context of Multi-Agent-based Systems, to the resource allocation problem. RL is a feedback based approach where an agent receives an immediate reward for its previous action, and from there on, it will try to learn a better policy for the long run, in order to maximize a given utility function. Therefore, the agent's goal – roughly speaking – is to maximize the total amount of reward it receives in the long run [4]. The rationale behind the use of that approach is because RL based approaches have been successfully used in solutions to problems that have similar requirements as the resource allocation in virtual networks [5-7]. Nevertheless, to the best of our knowledge RL has never been used before in that specific problem domain. Specifically, we represent each of the networks by a RL enabled agent. Each agent that represents a virtual network is responsible for customizing its resources to the needs of its users while at the same time minimizing the costs incurred in using the substrate resources. Each of these agents has independent objectives. Similarly, each of the substrate physical networks is represented by an agent whose main objective is to ensure that the overall resources are efficiently used. In the end, our proposal is for the different agents to not only learn from the decisions they make, but also to cooperate and negotiate with each other so as to achieve both individual level as well as system level objectives.

By conception, a RL-based approach is an autonomic solution to the allocation of virtual network resources to substrate networks exhibiting the following distinguishing features:

I) Self-Configuring: As users make requests for resources, the virtual network agents continuously evaluate their network topologies searching for possibilities of re-configuration. Whenever they find these possibilities, they can make requests for this to the substrate agents. Virtual network agents can also negotiate and cooperate amongst themselves so as to agree on the usage of substrate resources and achieve the best utility both at individual and at system levels.

II) Self-Optimizing: As the conditions of the substrate network change, substrate agents should exhibit proactive as well as reactive characteristics. For example, if a physical node and/or link are added to the network, or if their capacities are changed, the substrate agent should re-evaluate the network load to establish a potential re-allocation of resources. While in principle this would not require any action on the side of the virtual network agents, in our solution we require that these agents always look out for possibilities of optimizing their resource usage whenever there are changes in the substrate network.

III) Self-Healing: If for any reason a node and/or link become unavailable, the substrate agent – in collaboration with virtual network agents – should make decisions so as to cause the minimal possible disruptions in the customer service. This situation is different from II) in a way that while self-optimization is mainly aimed at optimizing costs, and possible improvements in customer service levels, self-healing is much more urgent as in such cases there are possibilities of violating agreements with customers.

IV) User Context Aware: Based on user context information, the virtual network agent may take decisions about the resources being used by the user. For example, a user whose location is near a Wi-Fi hot spot and this user is stationary and he is occupying a high bandwidth – say for a video on demand service – he could be offloaded to the Wi-Fi and if changes to the nodes for this specific customer are needed, these changes can be effected by substrate agent.

2.2.3. Work in progress

Efficient Resource Allocation is one of the practical challenges of network virtualization. Most of the current approaches make some assumptions that cannot be achieved in practice. While these could give solutions to specific instances of this problem, the possibility to make improvements is a major motivation of this work. Our work involves making the resource allocation task autonomic, which is a vital characteristic especially considering the complexity of current and future networks.

Nevertheless we are facing several challenges because although the concept of RL is itself simple, its application to that problem domain is by far not straightforward. In particular we are working on a model of the process that all the agents will use to learn, a policy that each of agent employ, the specific reward functions for each agent and the learning objective. In addition we will have to adapt this to a multi-agent environment, in which case we need to ensure the stability of the agents' learning dynamics to ensure global convergence, and define how each of the agents adapt to actions of other agents. Last but not least we will have to decide on the cooperation mechanisms between virtual network agents and substrate network agents.

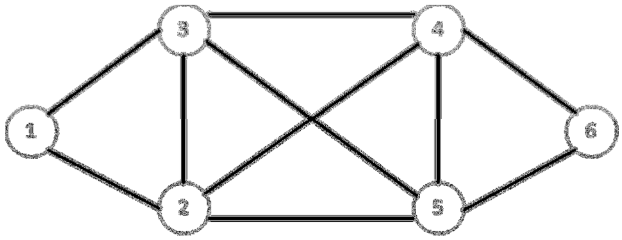
2.3. Energy Efficient Splitting

2.3.1. Overview

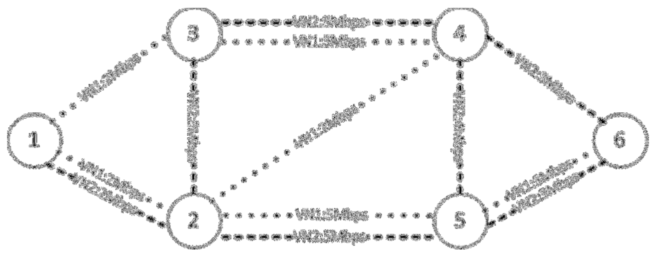
In this section, we specifically consider sleeping link optimizations in the network virtualization environment. Given the physical network topology (operated by the InP) and a set of virtual network topologies (substrate topologies), the objective is to minimize the power consumption in the physical network, while still satisfying bandwidth demands from individual virtual networks (VNs) on the top. In this regard, a heuristic offline algorithm is proposed, which tries to push maximum number of physical links into sleep mode over the off-peak period, while still guaranteeing the off-peak traffic demand of all involved VNs. The algorithm marks a physical link as a capable candidate in case the direct connected nodes are not isolated and there is a replacement path with guaranteed off-peak bandwidth demand, for all of involved VNs, in case of link inactivity. Simulation results over the GÉANT network topology with randomly generated virtual networks show our proposed algorithm is able to turn active notable number of physical links into sleep mode for energy saving over off-peak hours.

We use a simple network topology to illustrate the high-level idea of the algorithm in turning links into sleep mode over the off-peak hours in order to save energy. Figure 3.A shows a small physical network topology based on which multiple VNs can be provisioned. In Figure 3.B the mapped topology of two virtual networks, with their *peak-time* bandwidth demands over the physical network is shown. Each line model denotes one virtual network.

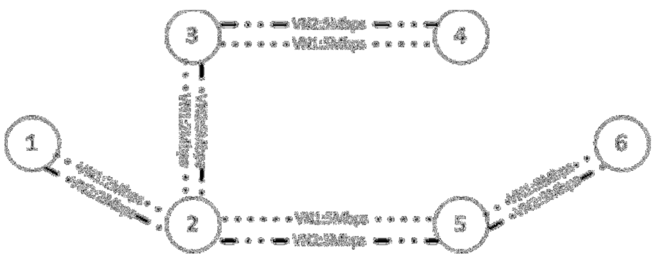
During *off-peak* time the traffic demands over the two VNs become reduced, in which case there is an opportunity to put a subset of physical links to the sleep mode. To do this, customer traffic across the virtual links in the VNs which are mapped onto the sleeping physical links need to be rerouted via alternative virtual path where there are available reserved bandwidth resources. For instance, Figure 3.C shows a resulting reduced topology to be used during off-peak time. As it is shown the physical links (1-3, 2-4, 4-5, and 4-6) are set into sleep mode while all the nodes still have full connectivity for all the previously involved virtual networks and their off-peak bandwidth demands are guaranteed through other active links. As can be observed from Figure 3.C, due to the unavailability of the sleeping link between node 2 and 4 in the reduced topology, off-peak traffic in VN1 between the two nodes needs to be rerouted via alternative paths in the reduced topology. Towards this end, Figure 3.D shows the re-routed traffic for the slept links over the defined replaced path. The replaced paths for links 4-6, 4-5, 1-3 and 2-4 are 1, 2, 3 and 4 respectively.



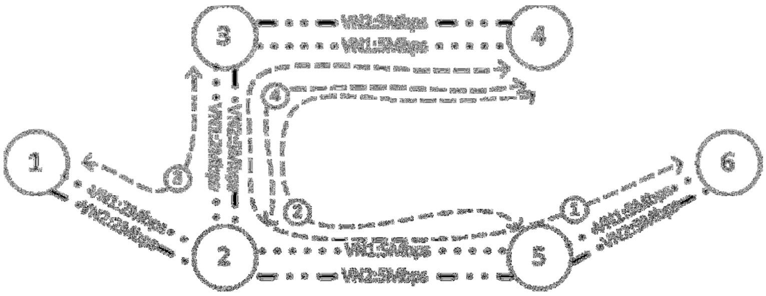
(A). Physical network topology.



(B). Peak time, mapped virtual networks topology.



(C). Reduced topology during off-peak time.



(D). Alternative paths for slept links in VNs.

Figure 3. An Example for link sleeping and traffic rerouting

2.3.2. Problem formulation

The objective is to reduce power consumption in network virtualization environment over off-peak hours, while still offering bandwidth demands. This could be happen by setting maximum number of physical links into sleep mode, over off-peak hours. The algorithm description is the following:

Given:

i) Physical substrate network topology.

ii) Overlay virtual network topologies.

iii) Physical link capacity and allocated bandwidth for each virtual network.

iv) Off-peak virtual network traffic demands.

Find: The set of links which are capable to be set into sleep mode over off-peak time.

Subject-to: Full connectivity and off-peak bandwidth constraints.

The problem can also be defined with Integer Linear Programming (ILP), precisely. The substrate network is modeled as an undirected graph $G_s = (V_s, E_s)$, where V_s is the set of substrate vertices, and E_s is the set of substrate edges.Vertices represents nodes and edges denote links in network. So, $N_s = |V_s|$ represent total number of physical nodes, and $L_s = |E_s|$ shows total number of substrate links.

The virtual network topology is asubset of the actual physical network topology. Similar to substrate network each of the overlay virtual networks is also modeled as an undirected graph $G_v = (V_v, E_v)$, where V_v is the set of virtual nodes, and E_v is the set of virtual links. Each virtual link, between node i and j $(L_{i,j})$, in n_{th} virtual network (VN_n) is associated with a residual bandwidth allocated over substrate network. This allocated bandwidth capacity is represented here with $C_n^{i,j}$. In addition, $P_n^{i,j}$ is set of links over a path between node i and j, in n_{th} virtual network. Besides, VN_T is set of all involved virtual networks.

Over the off-peak hours, networks are less utilized while full bandwidth capacity is active over all the links. In order toderive an algorithm which sets maximum number of links into sleep mode while guaranteeing required bandwidth, we need off-peak traffic demand. Therefore, off-peak traffic demand matrices or rates, for all the involved virtual networks, are given to our algorithm as an input. Off-peak traffic demand set for n_{th} virtual network is represented by OTD_n. These sets give the off-peak demands per link. In this regard, $\gamma_n^{i,j}$ reprents traffic demand between node i and j in n_{th} virtual network. Over off-peak period, available bandwidth capacityon virtual link between i and j, in n_{th} virtual network, is presented by $ac_n^{i,j}$.

In addition, P_L stands for total power consumption over substrate network links. We use $\delta_{i,j}$ to show the link status between node i and j. It is “1” in case the link is on, or it is “0” when the link is in sleep mode or off. Besides, $pl_{i,j}$ represents power consumed by the physical link between node i and j.

Considering the above definitions, it is possible to formulate the problem mathematically as follow:

Objective:

$$Minimize\left\{P_L = \sum_{i=1}^{N_s}\sum_{j=1}^{N_s}\delta_{i,j} \times pl_{i,j}\right\} \tag{1}$$

or

$$Maximize\left\{|E_s| - \sum_{i=1}^{N_s}\sum_{j=1}^{N_s}\delta_{i,j}\right\} \tag{2}$$

Subject-to:

$$\forall \gamma_n^{i,j} \in OTD_n \quad , \quad \exists P_n^{i,j} \quad \forall n \in VN_T \tag{3}$$

$$ac_n^{i,j} \geq \gamma_n^{i,j} \quad \forall L_{i,j} \in P_n^{i,j} \tag{4}$$

Eq. (1) states the main objective of algorithm which is minimizing the substrate network links power consumption over off-peak hours. This objective also can be represented by Eq. (2) which approaches the same goal by maximizing the number of possible links that are capable to be pushed into sleep mode. Eq. (3) and (4) are discussing the algorithm constraints. Eq. (3) maintains full connectivity between required nodes. It states there is at least one path for all off-peak bandwidth demands. Note that the path could be found with any desired routing protocol. Eq. (4) argues off-peak bandwidth guarantee.

2.3.3. Proposed Algorithm

This heuristic algorithm is trying to decrease power consumption over network virtualization environment while guarantees the off-peak bandwidth demands. The algorithm considers less stressed links status as inactive, and then answers the question for each link that is it necessary to set this link into active mode? This question should be answered precisely in order to guarantee full connectivity and off-peak bandwidth demands for all involved VNs.

The proposed algorithm is shown in. An initial parameter is calculated in first step in order to be used in next steps for sorting and decision making matters. Stress Rate (SR_k) isneededtobecalculatedfor all substrate network links. SR_k presents intensity of involved virtual networks over link k, and it is calculated by Eq. (5). This helps us for link sorting in third step. Besides, we use SR in step 4 in order to make decision to which links should be pushed into sleep mode at initial session.

$$SR_k = \frac{numberofVN\sin volvedinlink_k}{tota\ln umberofVNsoverISP(VN_T)} \tag{5}$$

In the second step the algorithm will calculate available bandwidth capacity for each virtual link over off-peak hours.

This is driven using off-peak traffic demand, which is given to the algorithm as an input, and with Eq. (6).

$$ac_n^{i,j} = C_n^{i,j} - \gamma_n^{i,j} \tag{6}$$

Since, we want to make network power consumption proportional to link utilization, and because the links with high number of involved VNs are more essential for connectivity and bandwidth demands, the algorithm will start setting links back to the active mode, if it is necessary, from a link which had the larger number of VNs involved and higher utilization. This happens in third step in which the algorithm sort the links in descending order, based on SR. For links with equal SR, the algorithm sortsthem based on link utilization. So, the top link in the list, has the largest number of VNs involved, and is the highest utilized link over the network.

In order to maximize the number of capable links to be pushed into sleep mode, the algorithm, in step 4, pushes all the links into sleep mode, except the links which their Stress Rate is equal to 1. This makes the temporary sleep mode topology.

Afterwards, the algorithm tries to check the possibility of setting a link into sleep mode by finding a replaced path, for all involved VNs in the link, which has enough bandwidth capacity for re-routed traffic. This happens in step 5, 6 and 7. At first, the algorithmtries to find a replaced path between two previously connected nodes, considering off-peak traffic demand, using the operator's desired routing protocol, for each involved VN in link. If the algorithm finds a path which supports the required off-peak demand for the link, then it returns the link as a capable for setting into sleep mode. Besides, it updates available bandwidth for all the links involved in the replaced path. In the other hand, if there is a replaced path for the VN, however one or some of the links over the path does not have enough capacity to handle the re-routed traffic, the algorithm turns the link into active mode and undoes all the capacity updates which have been done over the replaced path's links. Nevertheless, if the routing protocol does not return any replaced path between two previously connected nodes over the VN topology, the algorithm, in step 6, tries to find a replaced path over current active physical topology. In case there is a replaced path over the physical topology while one or some of the links over the path do not support the VN, which is the algorithm is working on, and then if there is enough physical capacity over the link, required bandwidth based on $C_n^{a,b}$ will be allocated and checking process of step 5 will be done again. However, if there is not any replaced path even over physical topology the link will be turned on and the bandwidth updates will be undone.

In some cases, over a physical link, one or some of the virtual links are not removable while the others are. Step 6 of the algorithm deals with this issue. It makes sure that the algorithm set a physical link into sleep mode, unless there is an un-removable virtual link over it.

Step1: Calculate Stress Rate (SR_k) using Eq. (5), for all the links over the ISP.

Step2: Calculate the off-peak available capacity ($ac_n^{i,j}$) of the links based on off-peak traffic demand, and Eq. (6).

Step3: Sort the links in descending order, based on SR. For links with equal SR, sort it based on link utilization. So, the top link in the list, has the largest number of VNs involved, and is the highest utilized link over the network.

Step4: Remove all the links from the topology, except the links with a SR equal 1.

Step5: For the top unchecked link in the list, find a replaced path by network default routing algorithm for two previously direct connected nodes (a, b) of link *k*, for all the involved VNs. Considering off-peak traffic demand, update the available capacity of the links for each VN over the path, with the following calculations and conditions:

$ac_n^{i,j} = ac_n^{i,j} - \gamma_n^{a,b} \rightarrow$

- If for all the links over the replaced path: $ac_n^{a,b} \geq 0$, then go to Step 7.
- If there is one or more links over the replaced path that $ac_n^{a,b} < 0$, then put back the link to the topology and undo the updates, then go to Step 8.
- If no replaced path is founded between node "a" and "b", go to the next step.

Step6: Try to find a replaced path, by network default routing protocol, over current active physical links, then:

- If there is a physical path however some of the links over the path does not support our desired VN, and the links have enough physical bandwidth, then allocate the required bandwidth for the VN based on $C_n^{a,b}$, and go back to Step 5 and do the checking process for this link again.
- Otherwise; put the link back to the topology and undo the updates, then go to Step 8.

Step7: Check the capability of link removal for all VNs involved in the link, then:

- If all the VNs are removable go to the next step.
- Otherwise, put back the link into the topology and undo the updates, then go to the next step.

Step8: Repeat Step 5 for the next link, unless all the links are checked; otherwise stop.

Figure 4. Proposed algorithm for link removal

3. Virtualized Resource Management in Wireless Networks

3.1. Research Challenges in Wireless Network Virtualization

The research challenge is to extend the network virtualization from wired network to wireless cellular networks. The difference mainly between the wired and wireless domain is the broadcasting nature of the wireless links. The wireless cellular networks links suffer more interference than the wired networks. Radio resource management and scheduling could be based on different criteria such as bandwidth, data rate, channel conditions and pre-defined contracts among virtual networks. Hence in order to embed virtual links without interference, it is important to divide the communication domain into different dimensions in a slotted way as the wired network virtualization does. These dimensions can be frequency, time and space. Hence it is one scheduling problem of frequency, time, space or code allocation.

The introduction of network virtualization to mobile networks, e.g. LTE networks, is a uniqu e opportunity for mobile operators to deal with the highly increasing traffic loads and cut dow n the infrastructure investment on CAPEX and OPEX costs through sharing the physical infra structures by multiple mobile operators.Radio resource management and scheduling could be based on traditional criteria such as bandwidth, data rate, channel conditions and pre-defined contracts among virtual network. Wireless resources on air interface in the case of cellular network are abstracted into unified resource and form a resource pool. The virtual network needs to provision a fair allocation of dedicated resource slices for users in order to provide more effective solutions for random access.

In wireless network, random access protocols provide a more flexible and efficient way of managing channel for access. Thus, an efficient collision resolution algorithm leads to fast system access and high throughput. The Pseudo Bayesian Broadcast algorithm is found to be exceptionally effective in practice since it makes nearly the "best possible use" of the information available on the network in determining the broadcast probabilities to use. Hence, we will try to introduce the Pseudo Bayesian Broadcast algorithm in the random access protocols to provide a dynamic back off algorithm with fast retransmission and access priority differentiation. Moreover, under the tempting market prospect, more and more Machine-to-Machine (M2M)applications have been implemented in current 2G/3G networks. Due to various advantages, the cellular networks have been considered as one of the best choices to bear M2M service. Because of the specific characteristics of M2M communication, there must be many incompatible factors in practice, and the random access performance of H2H communication will also be affected severely. Hence, we will try to propose a new algorithm, To evolve and develop competitive capabilities to support M2M communication, the system model of random access is built firstly, and then one power ramping strategy based on Logarithm is proposed for M2M.

3.2. Resource Allocation for Wireless Network Virtualization

3.2.1. Introduction

Following the IT resource virtualization in cloud computing such as CPU, memory and storage, network virtualization becomes the natural next step aiming to provide network or infrastructure providers with the ability to manage and control their networks in a more dynamic fashion. The concern of the EVANS project is on the virtualization of wireless networks, which present more challenging issues than wired networks due to specific features of wireless channels. Actually virtualization has already taken place in mobile cellular networks. For instance, there are many mobile network operators (MNOs) such as Lebara that don't have their own physical network infrastructure such as base stations (BS), etc. They typically rent such physical network infrastructure from other MNOs that own the infrastructure. For example, Lebara rents Vodafone's networks to provide mobile services to its end users. To distinguish these two types of MNOs we call the network infrastructure owners as *physical* MNOs (PMNOs) whereas the MNOs renting resource from others are called *virtual* MNOs (VMNOs). Actually this kind of virtualization is just a form of resource renting. On the other hand, the essence of wireless network virtualization, to our understanding, is to provide elasticity of resource allocation and thus the consequent pay-as-you-use business model, just like in cloud computing. However, these features are yet to be reflected in the current network renting market.

The aim of the work here is to provide a wireless network virtualization mechanism which helps enable a more agile business model where a VMNO can request and thus pay a PMNO in a more dynamic and pay-as-you-use manner. The concerned network scenario is illustrated in Figure where, e.g., base station (BS) *A* transmits packets from both MNO #1 and MNO #2. At the BS there is a need to differentiate and isolate traffics from these two VMNOs. Isolation gives the flexibility for a VMNO to manage and control its own virtual network in a way as if they physically owned this network. This flexibility can be to introduce a new tariff scheme or a different resource allocation algorithm, etc. From a PMNO's perspective, its network node such as a base station is partitioned into *slices* each representing a virtual mobile network (VMN), as depicted in Figure 5.

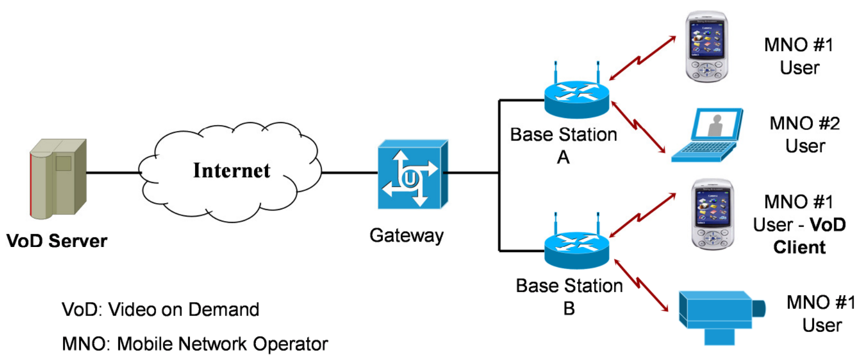


Figure 5. Wireless Networks, their End Users and their Connection to Internet

Apart from illustrating the concept of virtualizing one physical network node into multiple virtual network nodes, Figure 6 also shows the introduction of two types of networks:

*local*and *foreign*. Local network refers to the original physical network, which, though, is now only part of the whole physical network infrastructure as the other part is rented out to virtual networks. Foreign networks refer to virtual mobile networks. There is only one local VMN but there can be more than one foreign VMN. No research work in the literature makes this distinction. However, in real-life practice, a PMNO needs to know this difference as it serves not only foreign traffic but also its own traffic. These two types of traffics are dealt with differently. As can be shown later in the proposed network virtualization algorithms, they follow different optimization objectives. From resource allocation's perspective, the aim for foreign networks is to use as little as possible or just enough bandwidth to satisfy resource requirement whereas the aim for local networks is to use as much bandwidth as possible after the foreign traffic has been served satisfactorily.

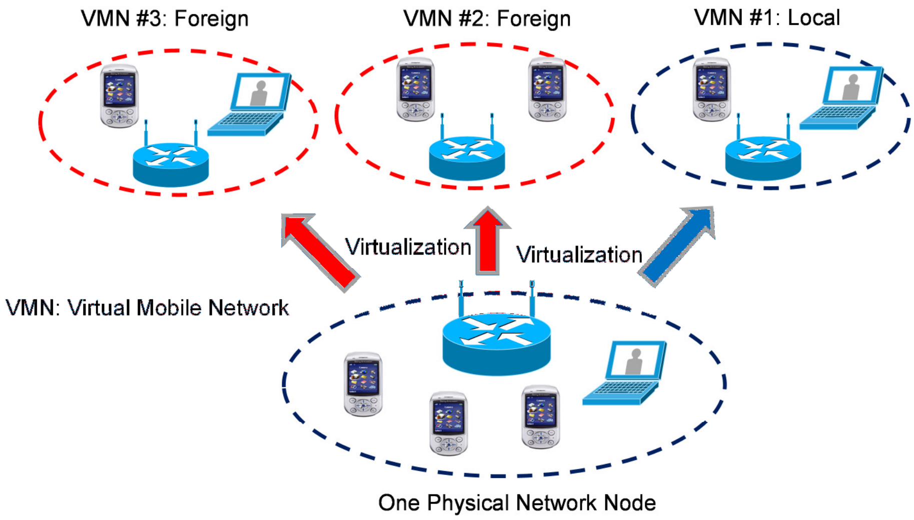


Figure 6. Wireless Network Virtualization

As far as the wireless network technology is concerned, this section uses IEEE 802.16 or so-called WiMAX. With high data rate, large network coverage, strong QoS capabilities and cheap network deployment and maintenance costs, IEEE 802.16 is viewed as a disruptive wireless technology and has many potential applications [Essex1].Depending on the applications and network investment, IEEE 802.16 networks can be configured to work indifferent modes, point-to-multipoint (PMP) or mesh mode. In the PMP mode, a BS serves multiple subscriber stations (SSs) within its coverage. In the mesh mode, SSs communicate with each other without a BS. This section assumes a PMP mode and considers traffic going from SSs to BSs, namely, uplink. These BS stations can be connected to the Internet via a gateway – refer to Figure 5.

There are two goals to be achieved: virtual network isolation and resource efficiency. Namely, in addition to achieving the basic functions of virtualization (as mainly represented by isolation), resource efficiency also needs to be obtained. To this end, a joint algorithm for both network virtualization and resource allocation of IEEE 802.16 networks is proposed. The algorithm involves the following two major steps: firstly to virtualize a physical wireless network into multiple slices each representing a virtual network; secondly, to carry out physical resource allocation within each virtual network (or slice). In particular, EVANS concerns OFDM (Orthogonal Frequency Division Multiplexing) as its physical layer to achieve more efficient resource utilization. Therefore, the resource allocation is conducted in terms of sub-carriers. Though the motivation and algorithm design are based on IEEE 802.16 or WiMAX networks, the principle and algorithmic essence are also applicable to other OFDMA-based wireless networks.

3.2.2. Problem Clarification

Resource allocation is performed by the BS and is needed for both downlink and uplink. Since a BS usually knows downlink data information, downlink resource allocation is relatively straightforward. EVANS focuses on uplink resource allocation which requires an effective cooperation of a BS and its corresponding SSs, as illustrated in Figure 7. Quality of Service (QoS) is also important for VMNs but is not considered within the scope of this document.

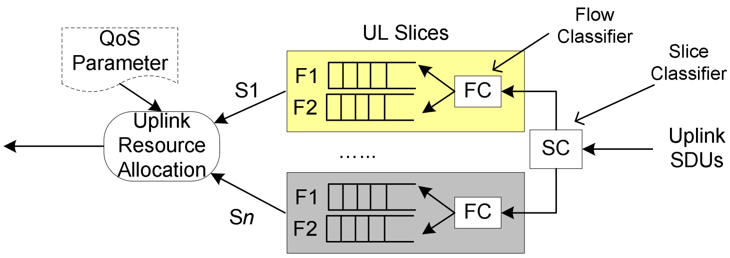


Figure 7. Uplink Packet Processing in a Virtualized IEEE 802.16 Network

Figure 7 also shows that the proposed network virtualization algorithm operates on two levels: slice and flow. Each slice represents a VMN and isolation is provided between slices. Each slice has a slice ID. There are multiple flows within a slice. A flow represents a session and each flow has a flow ID which is equivalent to WiMAX's session ID. Each uplink packet has both a flow ID showing its belonging to which VMN and a session ID identifying its session within a VMN. Virtualization is applied on network resources, namely, OFDMA sub-carriers. The problem to be solved here is how to allocate OFDMA sub-carriers to both slices and flows while aiming to maximize resource utilization.

A list of major notations used in this document is summarized in Table 1.

Table 1 Notations

Φ	Total bandwidth of system
P_T	Total transmit power of system
N_0	Noise power spectral density
N	Number of system subcarriers

M	Number of foreign slices
A	Universal set of system subcarriers
B	Subcarrier set allocated to local network flows
r_{S_i}	Rate assigned to slice S_i
$r_{f,S_i}^{\text{REQ-REAL}}$	Real rate requirement of traffic flow f in slice S_i
$r_{f,S_i}^{\text{REQ-ALG}}$	Algorithm rate requirement of traffic flow f in slice S_i
$I_{f,c}$	Assignment index indicating$f$-th flow occupy c-th subcarrier
$P_{f,c}$	Power allocated to f-th flow in c-th subcarrier
$h_{f,c}$	Channel gain of f-th flow in c-th subcarrier
$BER_{f,c}^{\text{target}}$	Target BER of f-th flow in c-th subcarrier

3.2.3. Proposed Elastic Wireless ResourceVirtualization Algorithm

1) Flow rate vs. BER in subcarrier

Consider a multiuser OFDMA system with K users and N subcarriers. Our design is based on flows instead of users in order to satisfy the QoS more conveniently in the future work. Each user may produce one or more traffic flows. Then it is natural to obtain that the channel gain of f-th flow in c-th subcarrier $h_{f,c}$ is equivalent to the channel gain of the user which produce f-th flow on c-th subcarrier. For easy expression and without losing generality, this section omits the user identification in the follow description.

Assuming power being allocated averagely among subcarriers, if M-ary quadratureamplitude modulation (MQAM) is employed the the BER for an AWGN channel can be given by [8]:

$$BER_{f,c} \approx 0.2\exp\left(-\frac{1.5P_{f,c}h_{f,c}^2}{\left(2^{r_{f,c}}-1\right)N_0\dfrac{\Phi}{N}}\right) \tag{7}$$

Then the maximum rate of f-th flow in c-th subcarrier is:

$$r_{f,c} = floor\left(\log_2\left(1-\frac{1.5P_{f,c}h_{f,c}^2}{\ln(5BER_{f,c}^{\text{target}})N_0\dfrac{\Phi}{N}}\right)\right) \tag{8}$$

2) Problem Formulation

As mentioned earlier, there are two types of slices in the design. One type represents the original physical network operator itself, which is denoted as Slice 0. It serves traffic from this network operator's users. Another type of slices is for foreign VMNOs that are renting the network resources from this PMNO. There can be more than one such slice and they are denoted as Slice1 to *M*.

The admission of VMNOs is subject to strict admission control imposed by the PMNO. But once they are admitted to this physical mobile network, their network resource requirements shall be guaranteed. Therefore, from PMNO's perspective, it should always allocate resources to foreign slices 1 to *M* first. And then the remaining resources can be used by its own traffic. Here in this section, the resources concerned are OFDMA subcarriers. The problem of the subcarriers assignment to Slices 1 to *M* can be described by the following optimization problem:

$$\min \sum_{f \in S_i} \sum_{c \in A} I_{f,c} \tag{9}$$

s.t.

$$\text{AC1:} \quad \sum_{f \in S_i} I_{f,c} \le 1$$

$$\text{AC2:} \quad I_{f,c} = \{0,1\}$$

$$\text{AC3:} \quad r_{f,S_i}^{\text{AL}} \ge r_{f,S_i}^{\text{REQ-ALG}}$$

$$\text{AC4:} \quad r_{f,S_i}^{\text{AL}} = \sum_{f \in S_i} \sum_{c \in A} I_{f,c} r_{f,c}$$

Here AC means Type A Constraint as below is another optimization problem for resource allocation in the local slice where constraints are denoted as type B for easy reference. The objective (3) is to allocate minimum subcarrier resources to foreign flows under constraints AC1 to AC4. AC1 denotes that each subcarrier is only occupied by at most one user at any time, AC2 indicates whether the c-th subcarrier is allocated to f-th flow (value 1) or not (value 0). AC3 ensures that the rate of each flow in foreign slices meet their requirements by the inputs of the algorithm. AC4 is the total rate of subcarriers allocated to the f-th flow.

For the local slice, Slice 0, there is no explicit resource requirement and the optimization objective is to maximize the system throughput as represented by $\max \sum_{f} r_{f,S_0}^{\text{AL}}$. So there is only BER constraint, which is expressed by $BER_{f,c}^{\text{target}}$ in Eq. (8) during the resource allocation process. The optimization problem can be formulated as:

$$\max \quad \sum_{f} r_{f,S_0}^{\mathrm{AL}} \tag{10}$$

s.t.

$$\text{BC1}: \quad r_{f,S_0}^{\mathrm{AL}} = \sum_{\{c\in B | f\in S_0\}} I_{f,c} r_{f,c}$$

$$\text{BC2}: \quad \sum_{f\in S_i} I_{f,c} = 1$$

$$\text{BC3}: \quad I_{f,c} = \{0,1\}$$

The optimization objective (4) denotes the system throughput of local Slice 0, BC1 and BC3 is similar to AC4 and AC2, BC2 ensures that each subcarrier is only occupied by at most one user at any time and all the subcarriers are allocated.

3) *Proposed Algorithms*

For the foreign slices, the optimization objective is to assign subcarriers to meet the requirements of all flows in the slices while occupy the subcarrier as little as possible. The optimization problem (9) is a typical binary integer programming problem. So it is suitable to be converted to *Office Assignment Problem (OAP)*, and use a linear programming (LP)-based branch-and-bound algorithm to solve the problem.

The form of *Office Assignment Problems* is

$$\min \quad \boldsymbol{q}_c^{\mathrm{T}} \boldsymbol{x} \tag{11}$$

s. t.

$$\boldsymbol{A_{eq}} \boldsymbol{x} = \boldsymbol{b_{eq}}$$

$$\boldsymbol{A_{neq}} \boldsymbol{x} \le \boldsymbol{b_{neq}}$$

$$x_i = \{0,1\}, \mathbf{x}=[x_1, \cdots x_i \cdots x_n]$$

Then, the optimization problem as expressed in Eq. 9 can be solved as follows. The operators are similar to these used in MATLAB.

ALG # 1 Virtual Resources Allocation Algorithm for Foreign Slices

1. Initialization:
2. Set $A = \{C_1, \cdots, C_N\}$, $P_{f,c} = P_T / N$, $I_{f,c} = 0$, $\boldsymbol{A_{eq}} = []$, $\boldsymbol{b_{eq}} = []$
3. Create optimization coefficients:
4. for all $f \in S_i, c \in A$
5. $(\boldsymbol{x})_j = I_{f,c}$

6. $(\boldsymbol{q}_c)_j = 1$

7. $r_{f,c} = floor\left(\log_2\left(1 - \frac{1.5P_{f,c}h_{f,c}^2}{\ln(5BER_{f,c}^{\text{target}})N_0\frac{\Phi}{N}}\right)\right)$

8. end for

9. for all $f \in S_i$

10. $\boldsymbol{A}_{neq}(i,:)$, $b_{neq}(i,:)$ = the constraint of (AC1)

11. $\boldsymbol{A}_{neq}(M+i,:)$, $b_{neq}(M+i,:)$ = the constraint of (AC3)

12. end for

13. Solve OAP to get $I_{f,c}$

For the local slice, the optimization problem in Eq. 10 can also be converted to OAP, which operation is described as follow:

ALG # 2 Virtual Resources Allocation Algorithm for Local kind of Slices

1. Initialization:

2. Set $P_{f,c} = P_T / N$, $I_{f,c} = 0$, $\boldsymbol{A}_{neq} = []$, $\boldsymbol{b}_{neq} = []$

3. Create optimization coefficients:

4. for all $f \in S_i, c \in B$

5. $(\boldsymbol{x})_j = I_{f,c}$

6. $(\boldsymbol{q}_c)_j = r_{f,c}$

7. $r_{f,c} = floor\left(\log_2\left(1 - \frac{1.5P_{f,c}h_{f,c}^2}{\ln(5BER_{f,c}^{\text{target}})N_0\frac{\Phi}{N}}\right)\right)$

8. end for

9. for all $f \in S_i$

10. $\boldsymbol{A}_{eq}(i,:)$, $b_{eq}(i,:)$ = the constraint of (BC2)

11. end for

12. Solve OAP to get $I_{f,c}$

To efficiently utilize the channel resources, an elastic wireless resource virtualization algorithm has been designed. In this algorithm, the basic design principles for such a virtualized network need to be satisfied. Namely, the foreign flow's total data rate or throughput must be guaranteed; and then on top of this the remaining channel resources should be used as much as possible by the local traffic to provide best possible service to local traffics. For this purpose, an Elastic Resource Virtualization Algorithm called ERVA is proposed, which is a combination of Algorithm 1 and Algorithm 2, as described below.

Elastic Resource Virtualization Algorithm (ERVA)

---

1. Input: M , $r_{S_i}^{\text{REQ-STATIC}}$, $r_{f,S_i}^{\text{REQ-REAL}}$
2. for i=1 to M // *foreign slice*
3. if $\sum_{f\in S_i} r_{f,S_i}^{\text{REQ-REAL}} \le r_{S_i}^{\text{REQ-STATIC}}$, then
4. $r_{f,S_i}^{\text{REQ-ALG}} = r_{f,S_i}^{\text{REQ-REAL}}$
5. else
6. $r_{f,S_i}^{\text{REQ-ALG}} = \dfrac{r_{f,S_i}^{\text{REQ-REAL}}}{\sum_{f\in S_i} r_{f,S_i}^{\text{REQ-REAL}}} r_{S_i}^{\text{REQ-STATIC}}$
7. end for
8. Choose $\{S_i\}, i = 1 \cdots M$ // *foreign slices*
9. Call ALG #1
10. Set $B = \left\{ C_l \,\middle|\, \sum_{\substack{f \\ l\in A}} I_{f,l} = 0 \right\}$
11. Choose $\{S_i\}, i = 0$ // *local slice*
12. Call ALG #2

From line 3 to 6, the total real traffic flow loads in foreign slices are checked to see if they have already reached the *upper bound* of the static requirement of foreign slices. If not then they are assigned invariably to input of allocation algorithm ALG #1. Otherwise, the excess portion will be clipped proportionally across all flows according to their real traffic loads, as shown in line 6. Then, the foreign slice static requirement contracts are guaranteed. And at the same time the margin between real traffic loads and static requirements of foreign slices is fully utilized. In line 8 and 9, subcarriers are allocated to the flows in foreign slices by call subroutine ALG #1. In line 10, the remainder subcarriers are aggregated to set B which is use in ALG #1 for resource allocation in local Slice 0. Line 11 and 12 denote the resource allocation process in local slice.

3.2.4. Performance Evaluation and Analysis

1) Overall Simulation Design

For comparison purpose, two other virtualization algorithms have been designed and implemented. The first one is the so-called static resource allocation, denoted as SRVA, where the available physical network resource is equally divided among M+1 networks, namely, M foreign networks and one local network. SRVA is the baseline algorithm serving to show how much improvement can be achieved by our proposed ERVA. In addition, another slightly more dynamic resource virtualization algorithm called FRVA is also proposal. FRVA allocates the same amount of resources to a foreign slice as requested in the SLA (Service Level Agreement) between the VMNO and the PMNO regardless of the real-time traffic from this particular VMNO. In contrast, our ERVA algorithm also takes into consideration the real-time traffic demand when allocating network resources to VNs (or slices), i.e., more elastic.

The simulations are designed in accordance with the system design principles, namely, VN isolation and resource efficiency in terms of overall network throughput.

2) Network Parameter Setup

In the simulation below, the IMT-2000 Vehicular Model A channel model as suggested by ITU-R M.1225 is utilized. It is a six-path Rayleigh fading model. The detailed parameters are shown in Table 2.

Table 1 Simulation Environments and Parameters

Carrier frequency		2.0 GHz	
Mobile velocity		100km/hr	
Multipath delay		Delay (ns)	Power (dB)
	Path 1	0	0
	Path 2	310	−1
	Path 3	710	−9
	Path 4	1 090	−10
	Path 5	1 730	−15
	Path 6	2 510	−20

The bandwidth is set to 2 MHz and divided into $N=128$ subcarriers. The cycle prefix length CP=16 for OFDM system, so the OFDM frame $T_f = 62.5\mu s$, and the bandwidth of subcarrier is $\Delta f = 16\ \mathrm{kHz}$. The Vehicular Model A channel coherence bandwidth $B_c \approx 0.54\ \mathrm{MHz}$ which is far larger than the bandwidth of OFDM subcarrier. Then, the frequency-selective channel is converted to frequency-flat sub-channels. Chunk-based algorithm is employed in the simulations. The subcarriers are grouped into $Z=16$ chunks, each consisting of $n_b=8$ subcarriers. The SNR on the subcarriers is

$$\gamma_{f,c}=\frac{P_{f,c}h_{f,c}^2}{N_0\frac{\Phi}{N}}=\gamma_0 h_{f,c}^2 . \qquad (12)$$

The following three slices are simulated: two foreign slices and one local slice. Traffic load follows the Poisson-distributed model for all flows. Each foreign slice has two flows and the local slice contains four flows. Table 3 summarizes the simulations setup for the different slices and flows. Flow 1 and Flow 2 are in foreign slice 1 and Flow 3 whereas Flow 4 are in foreign Slice 2. Since the local Slice 0 does not have maximum rate restriction, the traffic loads of Flow 5 to 8 in this slice are not limited aiming to simulating a demanding local traffic so as to stretch the physical network.

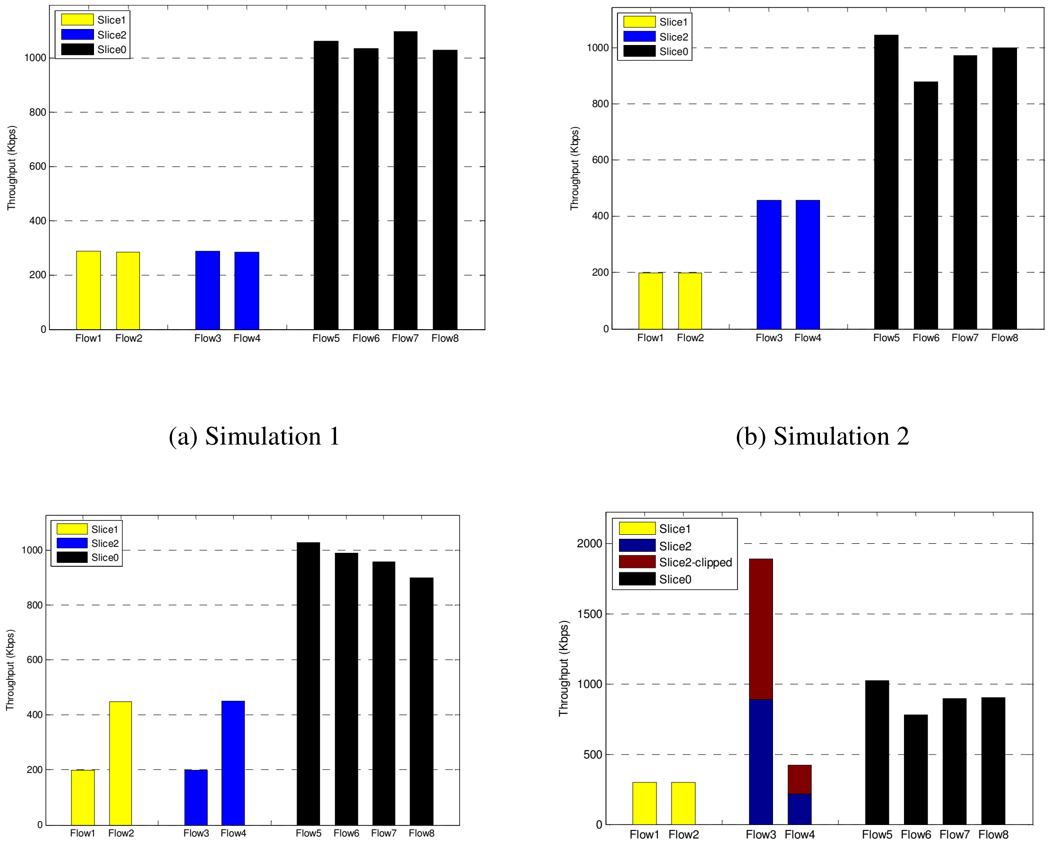
Table 3 Simulations setup for flows and slices ($\gamma_0=15\text{dB}$)

Simulation No.	Average flow load (λ) (Kbps)				Static Rate Requirements of foreign slices(Kbps)	
	Slice 1		Slice 2			
	Flow 1	Flow 2	Flow 3	Flow 4	Slice 1	Slice 2
Simulation 1	240	240	240	240	960	960
Simulation 2	120	120	360	360	960	960
Simulation 3	120	360	120	360	720	1200
Simulation 4	240	240	1800	360	960	960

3) Isolation

These simulations aim to demonstrate ERVA's efficacy in providing slice isolation for different static rate requirements and flow traffic setup. According to Table 3, four groups of setup are evaluated for the proposed ERVA scheme and their results are shown in Figure 8 (a) to (d) respectively.

In Simulations 1, 2 and 3, the foreign average traffic flow loads are all below the static rate requirements of foreign Slice1 and 2. These are common communication scenarios where foreign networks usually don't exceed the agreed bandwidth requirement. In Simulation 1, all average flow traffic loads are same as 240 Kbps. In Simulation 2, the loads are the same across flows within a foreign slice but vary across slices. In Simulation 3, the flow loads are different within a foreign slice. From the corresponding results of Figure 8 (a), (b) and (c), it can be observed that the flows in foreign slices are allocated the requested amount of bandwidth despite the fact that the wireless channel is changing. This to certain extent shows the isolation among flows.



(a) Simulation 1 (b) Simulation 2

(c) Simulation 3 (d) Simulation 4

Figure 8. ERVA's Provisioning of Isolation across Slices (Virtual Networks)

In Simulation 4, the foreign traffic loads of the flows in Slice 2 are more than the corresponding static requirements. In this case only the amount agreed at the network renting stage is guaranteed whereas the exceeding amount, as represented by the top portion of the Flow 3 bar, is not served and thus dropped. This is because the isolation is designed between different slices. Flow 1 and 2 are in Slice 1 and Flow 3 and 4 are in Slice 2. So if the sum of Flow3 and 4 exceeds, both of them will be cut proportionally and this does not affect Slice1. The sum of Flow 3 and 4 meet the requirement of Slice2 after clipping.

4) Resource efficiency

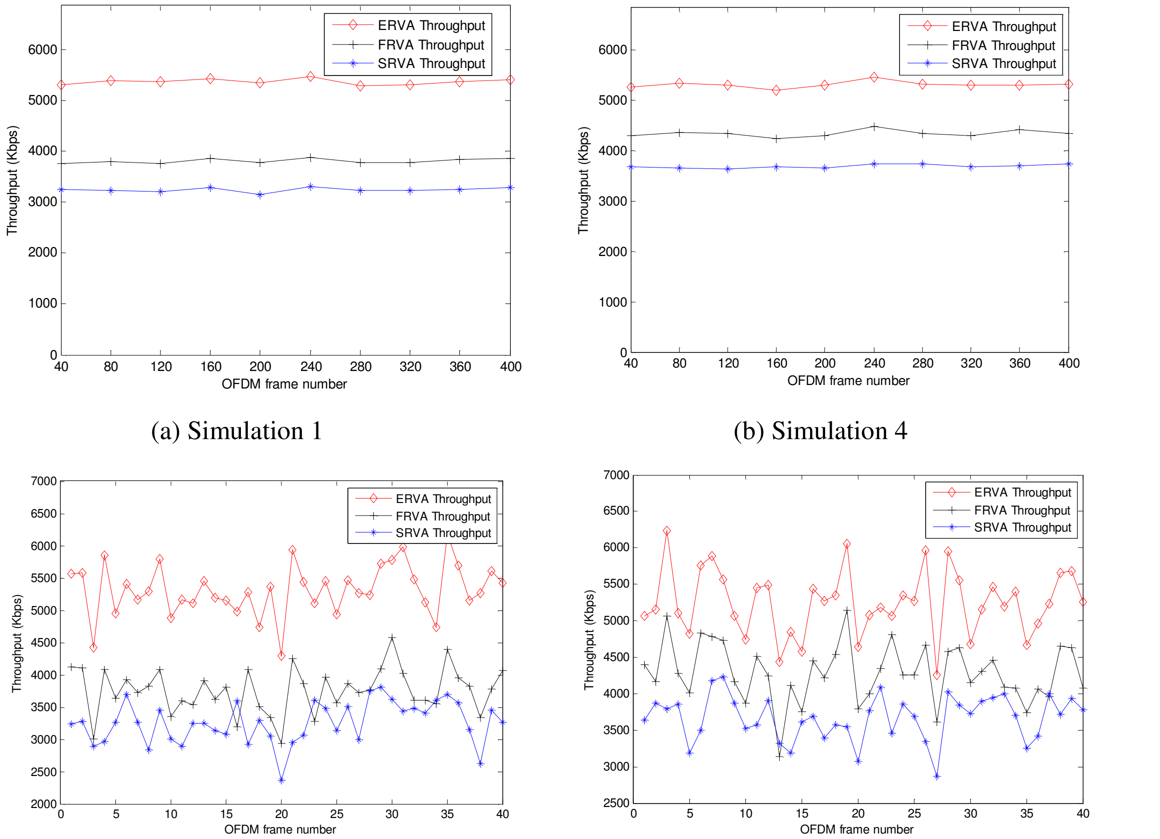
This sub-section is to evaluate the second design goal of the ERVA algorithm, namely, resource efficiency in terms of system overall throughput. Note that this throughput is measured from the PMNO's point of view, namely the throughput of the overall physical network rather than individual virtual networks. ERVA attempts to increase the throughput of the whole physical network while satisfying the bandwidth requirement from individual virtual networks.

Table 3 Simulations setup for flows and slices ($\gamma_0 = 15\text{dB}$)

Simulation No.	Average flow load (λ) (Kbps)				Static Rate Requirements of foreign slices(Kbps)	
	Slice 1		Slice 2			
	Flow 1	Flow 2	Flow 3	Flow 4	Slice 1	Slice 2
Simulation 1	240	240	240	240	960	960
Simulation 2	120	120	360	360	960	960
Simulation 3	120	360	120	360	720	1200
Simulation 4	240	240	1800	360	960	960

The same traffic settings as in Table 3 are employed. Only results from Simulation 1 and 4 are shown in Figure 9, as in (a) and (b) respectively, because the results from Simulation 2 and 3 are similar with these from Simulation 1. The three algorithms (i.e., SRVA, FRBA and ERVA) discussed Section 5.1 are evaluated. They are invoked following an allocation cycle that is equal to the length of an OFDM frame. Network throughput is measured in each cycle. Each point on the graphs is the average of 10 runs.

It can be observed from Figure 9 (a) and (b) that the throughput performance of ERVA is the best, which is followed by FRVA, whereas SRVA performs the worst. The main reason is that both ERVA and FRVA have employed dynamic resource allocation via optimization method whereas SRVA performs a static resource allocation and thus no selective diversity gain is obtained. The ERVA is better than the FRVA because ERVA utilizes the spare resources between static rate requirements and real traffic loads. A comparison between Figure 9 (a) and (b) shows that the throughputs of FRVA and SRVA in (b) are larger than these in (a) respectively whereas ERVA performs roughly the same. This is mainly because the traffic loads are larger in Simulation 4 (i.e., Figure 9 (b)) than that in Simulation 1 (i.e., Figure 9 (a)).



(a) Simulation 1 (b) Simulation 4

(c) Simulation 1 (zoom in) (d) Simulation 4 (zoom in)

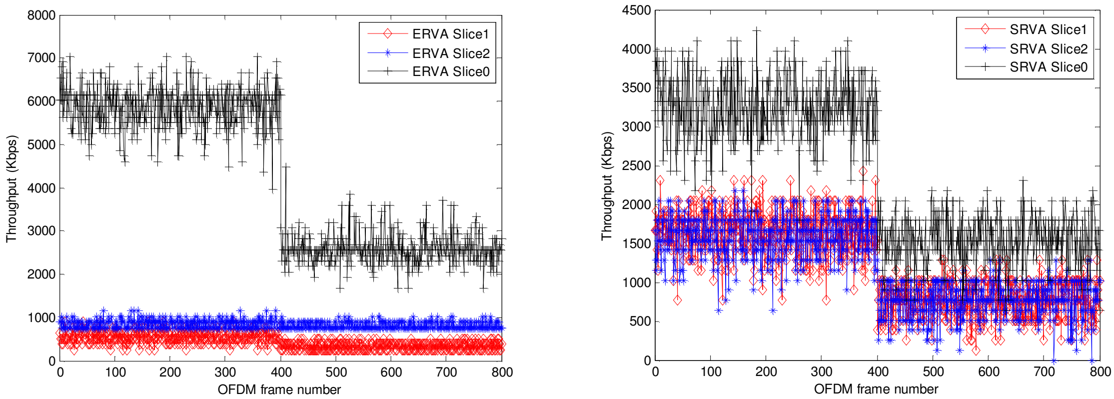
Figure 9. System Throughput of different algorithms.

Figure 9 (c) and (d) zoom into the first 40 frames of (a) and (b) respectively and show the detailed throughput performance of each algorithm. It can be observed that the trend stays the same though there are some variations between different sampling points (i.e., frame numbers).

- The Effect of Channel Quality on System Performance

In this simulation, the effects of different strategies of resource allocation among all slices are evaluated when the overall channel capacity varies due to the change of SNR (Signal to Noise Ratio). The slice requirements and flows load setup are the same as in Simulation 2, two different average SNR values, i.e., 20dB and 14dB, are used here.

The results are shown in Figure 10, where the SNR value is 20dB for the OFDM frame number of 0 to 400 and the SNR value is set to 14 dB for the OFDM frame number of 400 to 800. Figure 10 (a) and (b) show the throughput of ERVA and SRVA respectively. From Figure 10(a) it can be observed that the throughputs of foreign Slice 1 and 2 are not affected by the channel capacity variation. However, there is performance degradation for local Slice 0 as SNR goes down from 20dB to 14dB. This is because channel capacity decreases as SNR gets worse. This illustrates that the proposed ERVA algorithm guarantees that the SLAs of foreign slices are satisfied and the channel deterioration mainly affects local traffic. To compensate this kind of unfairness, ERVA allows local traffic to make use of the spare resources left over by foreign slices should the traffic from foreign slices is not demanding, as illustrated in Figure 9. In contrast, SRVA does not provide guarantee for foreign slices. In particular due to the static nature of SRVA, resources are over provisioned to low-demanding foreign slices whereas high-demanding local slice suffers squeezed traffic throughput. For example, ERVA can provide about 7000Kbps when SNR equals to 20dB whereas SRVA can only provide up to 4300Kbps, as shown in Figure 10.



(a) ERVA (b) SRVA

Figure 10. Throughput across Slices under Different SNR

- Delivery Ratio

In this simulation, the delivery ratios of three algorithms under different traffic loads are evaluated. The delivery ratio is defined as: $Ratio_D = Data_D / Data_R$, where $Data_D$ is the total amount of data delivered (i.e., the summation of real flow throughputs), and $Data_R$ is the total amount of data requested for delivery (i.e., the summation of traffic flow loads). When the network congested some data may be dropped.

The static rate requirements of foreign Slice 1 and 2 are 960 Kbps, and each traffic load of Flow 1 to 4 is set from 480 Kbps to 5760 Kbps. Since the flow loads in local Slice 0 are adaptive so that the resources allocated to Slice 0 are fully used, the system delivery ratio is dominated by foreign Slice 1 and 2. From Figure 11, we can see that our proposed ERVA have the same delivery performance as FRVA. When the flow loads are light, all schemes work well. However, with the flow loads increasing, both ERVA and FRVA perform better than traditional SRVA.

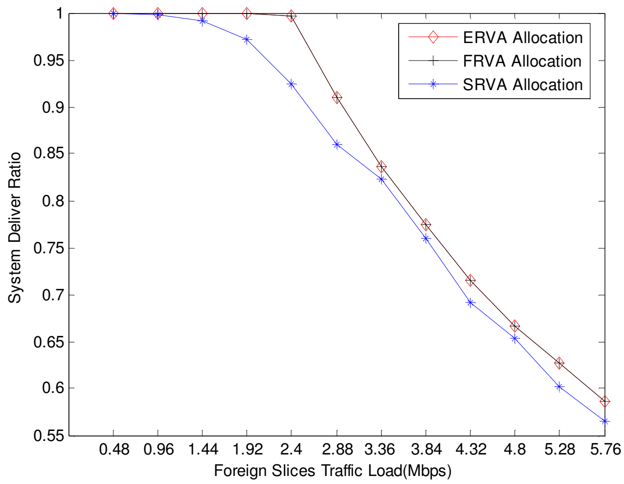


Figure 11. System Delivery ratio under different foreign flow loads

3.2.5. Summary

Based on the analysis of the existing work on wireless network virtualization, we have proposed an algorithm for joint network virtualization and resource allocation of IEEE 802.16 wireless networks. The algorithm not only provides network virtualization (isolation) but also achieves network resource efficiency. The latter is measured in terms of network throughput and packet delivery ratio. The simulation results show that the above goals have been achieved. Though the motivation and algorithm design are based on IEEE 802.16 or WiMAX networks, the principle and algorithmic essence are also applicable to other OFDMA-based wireless networks.

The next step is to introduce power allocation together with sub-carrier resource allocation during wireless network virtualization. Consideration of service differentiation (i.e., to provide QoS guarantee) is also within our future plan.

3.3. Radio Resource Management for Wireless Network Virtualization

3.3.1 Virtualizing wireless medium in LTE networks

Virtualization of the wireless resources on the air interface in the case of cellular network such as LTE and WIMAX systems means virtualizing the base station in order to schedule wireless resources among multiple Mobile Virtual Network Operators (MVNOs). Through virtualization, multiple MVNOs are able to run their own network on the same physical network infrastructure and provide custom services to their end users. In this situation, the air interface wireless resources need to be abstracted to a resource pool. Resources from this pool can be dynamically assigned to virtual networks based on some criteria (e.g., bandwidth and predefined contracts), which will improve the resource utilization. Fig. 12 shows An example architecture of virtualizing radio resources, in which multiple MVNOs share the physical eNodeB.

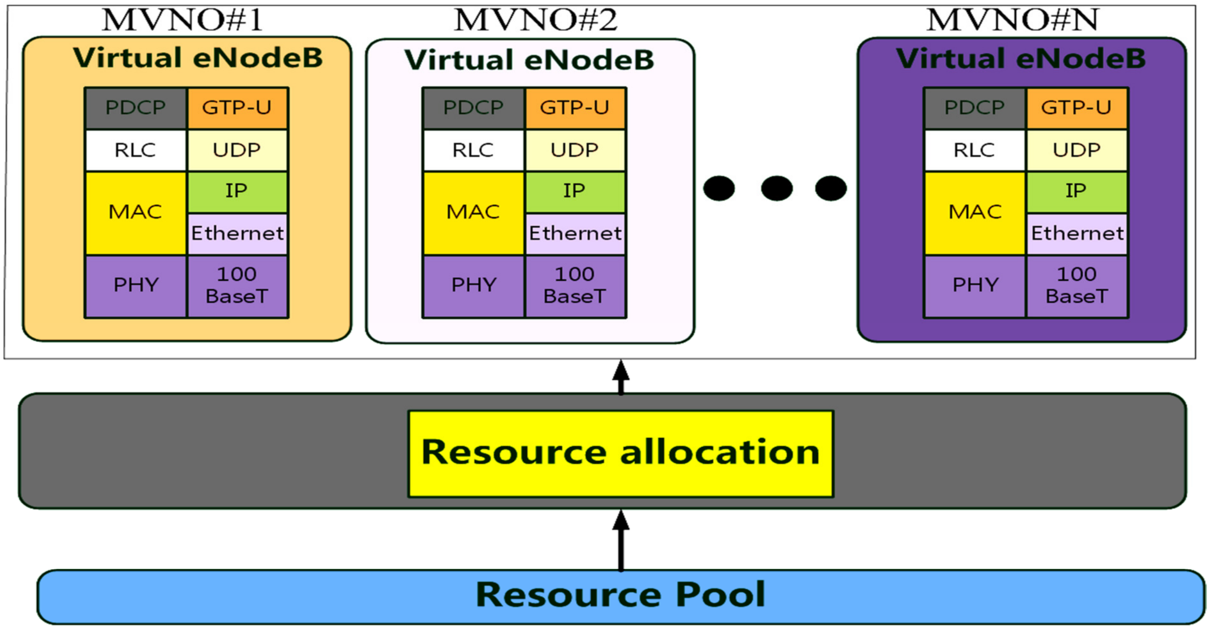


Figure 12. An example architecture of virtualizing radio resources.

LTE which was introduced by 3GPP is anticipated to be one of the promising solutions to the problems faced by today's mobile networks. Considering the characteristics of LTE, it is chosen by us as the study case of applying virtualization technology to cellular networks. In this work, we are mainly focused on LTE wireless medium (air interface wireless resources) virtualization, which is of a significant challenge. Compared with current mobile networks, where wireless resources are only assigned within one specific network, wireless resources should be allocated not only within one virtual network but also among multiple different virtual networks. Enhanced schedulers based on some criteria mentioned previously should be designed for wireless resource virtualization to allocate wireless frequency between different virtual mobile networks. In order to verify the potential performance gain through virtualization, simulations play a vital role.

3.3.2 Radio Resource Virtualized for Random Access

Random access protocols provide a more flexible and efficient way of managing channel for access. During the random access procedure, terminals must select random access channel to send access requests. If other terminals select the same channel, this situation will result in a collision and these collision terminals will back off and retransmit their access requests.

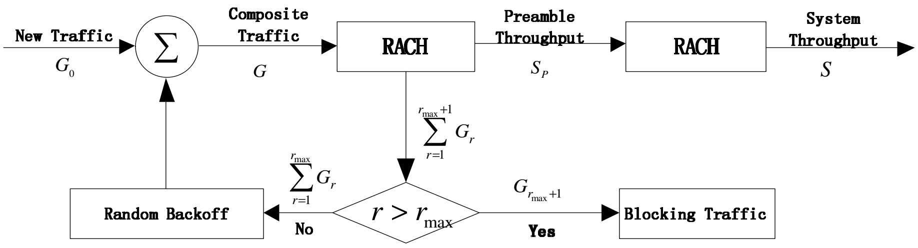


Figure 13. System model for RA

Figure 13 shows the system model of RA, which M2M traffic exists with H2H traffic. We provide a dynamic back off algorithm with fast retransmission and access priority differentiation. It can obtain high successful random access rate and low access delay, and satisfy QoS requirements. According to the access requests in uplink, the Node B estimates ATs in the next slot based on the Pseudo Bayesian algorithm. When there are collisions caused by competition in the current slot, the proposed algorithm decides whether to back off for the collided terminals in the next slot according to the load. Hence, we should firstly estimate the number of ATs. In addition, in order to provide access priority, different traffics have a different persistence value, which can be default by the systems.

With the development of Machine-to-Machine (M2M) communication, the cellular networks was one of the best choices to bear M2M service , which with the advantages of low-cost and large coverage, have nearly spread to every corner of the world. However, how to evolve and develop competitive capabilities to support M2M communication exists with H2H communication is an excellent challenge. To solve this problem, we built the system model of Random Access (RA) and proposed one power ramping strategy based on Logarithm for M2M.

The existing power ramping scheme is open-loop power control during the RA procedure. When mobile stations (MSs) fail to get through the RA procedure, they will retransmit the request with a higher power to guarantee the success rate, such as fixed ramping, linear ramping and geometric ramping. However, for the differences between M2M and H2H communication, we think that if M2M communication is treated with no difference from H2H communication, the interference generated from huge MDs is so severe that the performance of H2H communication can't be well guaranteed. Therefore, we proposed one power ramping scheme on the basis of logarithm steps for M2M to improve H2H communication performances.

4. Conclusions

The issues related to vertical management of virtualized resources can be categorized for wireless and wired virtualized networks separately due to the different nature of these networks. EVANS deals with the challenges related to these environments separately to identify the problems and solution related to each of them accordingly. Our goal is to provide the basis for static and dynamic vertical management of virtualized resources. In this regard, resource allocation, resource scheduling and energy efficient link splitting for wired virtualized networks and resource allocation and management for virtualized wireless networks are discussed.

References

[1] Y. Zhu and M. Ammar, Algorithms for Assigning Substrate Network Resources to Virtual Network Components in IEEE Infocom, 2006.

[2] M. Yu, Y. Yi, J. Rexford, and M. Chiang. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. SIGCOMM Comput. Commun. Rev., 38(2):17–29, 2008.

[3] J. Lu, J. Turner., Efficient Mapping of Virtual Networks onto a Shared Substrate, Technical Report WUCSE-2006-35, Washington University, 2006.

[4] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England., 2005.

[5] A. L. Stefan, M. Ramkumar, R. H. Nielsen, N. R. Prasad and R. Prasad, A QoS aware reinforcement learning algorithm for macro-femto interference in dynamic environments, 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, 2011.

[6] F. Bernardo, R. Agusti, J. Perez-Romero and O. Sallent, Distributed Spectrum Management based on Reinforcement Learning, Proceedings of the 4th International Conference on CROWNCOM, 2009.

[7] D. Minarolli and B. Freisleben, Utility-driven Allocation of Multiple Types of Resources to Virtual Machines in Clouds, IEEE 13th Conference on Commerce and Enterprise Computing (CEC), 2011.

[8] A. J. Goldsmith and S. G.Chua "Variable-rate variable-power MQAM for fading channels," IEEE Transactions on Communications, Vol. 45, no. 10, pp. 1218-1230, 1997.