Virtualizing Vehicular Node Resources:
Feasibility Study of Virtual Machine Migration

Benjamin Baron[a,*], Miguel Campista[b], Prométhée Spathis[a], Luís Henrique M. K. Costa[b], Marcelo Dias de Amorim[a], Otto Carlos M. B. Duarte[b], Guy Pujolle[a], Yannis Viniotis[c,d]

[a] *LIP6/CNRS – UPMC Sorbonne Universités*
[b] *COPPE/GTA – Universidade Federal do Rio de Janeiro*
[c] *North Carolina State University*
[d] *Cisco Systems*

Abstract

With emerging geo-distributed services, there is a need to coordinate the use of resources offered by field-area networks. In the case of vehicular networks, such resources include the processing, sensing, and storage capabilities offered to service providers for urban sensing or intelligent transportation. In this paper, we propose to virtualize the resources embedded on the vehicular nodes to allow multiple tenants to coexist and deploy their services on the same underlying mobile substrate. Virtualization is the task of an infrastructure provider that controls the mobile substrate and allocates sliced resources to the tenants. A service results from a collection of virtual machines hosted on the mobile nodes allocated by the infrastructure provider. Efficient utilization of the node resources may trigger virtual machine migrations. We study the problem of virtual machine migrations through V2V communications between mobile nodes. To evaluate the impact of such migrations on the resource allocation process, we use the real traces of a bus transit system to simulate a vehicular network where virtual machines migrate via V2V communications. Our results show that virtual machines of several hundreds of Megabytes can migrate between moving buses. We then discuss design principles and research issues toward the full virtualization of opportunistic networks.

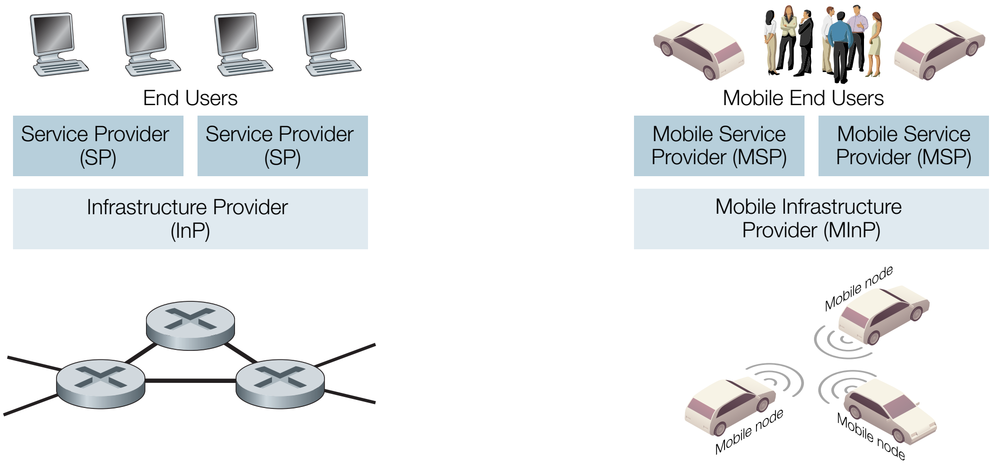
Keywords: Vehicular network, virtualization, virtual machine migration.

1. Introduction

New services and network usages have emerged with the recent networking paradigm of the Internet of Things, like context-based sensing or crowdsensing [1]. The applications and

[*] Corresponding author

Email addresses: benjamin.baron@lip6.fr (Benjamin Baron), miguel@gta.ufrj.br (Miguel Campista), promethee.spathis@lip6.fr (Prométhée Spathis), luish@gta.ufrj.br (Luís Henrique M. K. Costa), marcelo.amorim@lip6.fr (Marcelo Dias de Amorim), otto@gta.ufrj.br (Otto Carlos M. B. Duarte), guy.pujolle@lip6.fr (Guy Pujolle), candice@ncsu.edu (Yannis Viniotis)


End Users
Service Provider (SP)
Service Provider (SP)
Infrastructure Provider (InP)
Mobile End Users
Mobile Service Provider (MSP)
Mobile Service Provider (MSP)
Mobile Infrastructure Provider (MInP)
Mobile node
Mobile node
Mobile node

(a) Fixed infrastructure network model.

(b) Mobile network model.

Figure 1: In the fixed-infrastructure network model, the InP manages a network of fixed nodes and virtualizes their resources to SPs, which in turn offer services to end users. In the mobile network model, the MInP must manage the mobility nodes to virtualize the resources they embed and provide guarantees to the MSPs.

services require very low and predictable bandwidth and target geo-distributed (*e.g.*, sensor-based monitoring), fast mobile (*e.g.*, connected vehicles and rail), and large-scale platforms (*e.g.*, Smart Grid and Smart City). These services rely on field-area networks with greater capacities in terms of processing, storage, and bandwidth. Thus, associated services depend on a large collection of distributed mobile nodes to collect, process, and aggregate data to perform real-time analytics and make fast operational decisions [2]. Vehicles will definitely play a major role in such a scenario. Enabling virtualization on top of these mobile nodes, and in particular on top of vehicles, would offer service providers immediate access to these new emerging paradigms; in return, they bill their services to their end users, leaving the maintenance of such highly mobile and intermittently-connected networks to infrastructure providers.

Most of the existing work on network virtualization focuses on fixed network infrastructures [3]. In the literature, the traditional Internet Service Provider (ISP) is divided into two independent entities with distinct roles: (i) Infrastructure Providers, which manage the fixed infrastructure, and (ii) Service Providers, which form virtual networks from the resources allocated by possibly multiple Infrastructure Providers to offer end-to-end services to their customers. Recently, SDN (Software-Defined Networking) has facilitated the virtualization of wired networks with network hypervisors such as FlowVisor [4] or OpenVirtX [5]. Conversely, the existing work on mobile network virtualization focuses on the virtualization of mobile infrastructure networks, such as cellular or local wireless networks [6, 7]. To our knowledge, our work is the first one that proposes to virtualize the resources hosted on mobile nodes.

In this paper, we address the problem of virtualizing the resources hosted on the mobile nodes. In our solution, we propose a centralized architecture where a Mobile Infrastructure Provider (MInP) manages the physical substrate mobile network and virtualizes the resources of the mobile nodes. We consider the case of a large-scale mobile network such as a vehicular network (VANET) or a vehicular delay-tolerant network (V-DTN). As depicted in Figure 1, virtualizing vehicles allows multiple tenants, here denoted as Mobile Service Providers (MSP), to coexist and deploy their services on the same underlying physical substrate. The MInP slices resources of the vehicles (CPU, memory, storage, forwarding, on-board sensors) into virtual machines and allocates the virtual machines to the MSP so as to match their demands.

More specifically, the MInP leverages a centralized architecture with a controller that runs the allocation procedure with the MSPs' requirements as input. The allocation procedure allocates a set of virtual machines hosted on different mobile nodes so as to accommodate the MSPs' requirements. We consider that the central controller is always on-line (*e.g.,* connected via a cellular network); thus, re-allocations of the MSPs' demands result in *virtual machine migrations* from one vehicle to another. In this paper, we focus specifically on this problem as it is central to the feasibility study we conduct. Virtual machine migrations also happen when the current set of allocated virtual machines does not satisfy the requirements of the MSPs' demands (*e.g.,* the mobile node hosting the virtual machine leaves an area to be sensed and needs to be moved to the original area), or just before the node hosting the virtual machine becomes inactive.

Our main contributions are in the following three areas:

Mobile node resource virtualization. We propose a novel architecture to virtualize the resources embedded in mobile nodes. The MInP slices and allocates the resources to tenants, or MSPs, that will in turn aggregate the virtual resources to form virtual networks.

V2V Virtual machine migration. Changes in the physical topology trigger reallocations of the virtual resources and virtual machine migrations. Instead of using cellular connectivity to migrate the virtual machines, we leverage the V2V communications between vehicles.

System capacity. We evaluate the feasibility of V2V virtual machine migrations using real traces of a bus transit system. The results show that it is feasible to trigger virtual machine migrations at specific locations where the mobile nodes are most likely to encounter each other.

This paper is organized as follows. In Section 2, we overview the existing work on fixed and mobile network virtualization. In Section 3, we discuss the business model and design requirements involved with the virtualization of mobile node resources. In Section 4, we evaluate our proposal with real traces of a bus transit system under different scenarios and show that a vehicular network can serve as a communication substrate to be shared among multiple tenants. Finally, in Section 5, we overview some of the challenges and open issues involved with virtualizing mobile opportunistic networks as a perspective to our work.

2. Virtualization platforms

Let us first recall design principles of virtualization and review related work.

Wired network virtualization. Network virtualization refers to the decoupling of the physical routers (line card and CPU) from the forwarding functions. Multiple virtual router instances are then created on top of a single physical router. The collection of virtual nodes connected together by a set of virtual links forms a virtual network, whose topology is a subset of the underlying physical topology [3, 8]. Each virtual node is hosted on a physical node, although a virtual link spans over a physical path in the underlying network topology. The dynamics of virtual networks are different from those of the underlying physical networks on which they rely, as it is possible to turn the virtual routers on and off, and to migrate them. Virtual router migration consists in moving both the control and data plane states and processes from the hosting physical router to a target physical router [9].

Works on software-defined networking facilitate the deployment of network virtualization by capitalizing on the decoupling of the control plane from the data plane [10]. A central controller manages the control plane of software-defined networks and modifies the forwarding states of the networking devices. FlowVisor [4] and more recently OpenVirteX [5] define network hypervisors capable of mapping the forwarding states corresponding to the control logic of the virtual networks belonging to the tenants onto the physical networking equipment. In an SDN network, migration then consists in copying the flow tables from the host router to the target router, as SDN provides packet-forwarding abstraction [11]. Keller *et al.* propose LIME to migrate an entire network composed of virtual machines to a different collection of physical resources [12]. In their experiments, they show that the procedure can be achieved without any network interruption. These work focus on virtualizing fixed network infrastructures, however, in this paper, we are interested in virtualizing a large-scale mobile network, and in particular the opportunistic migration of virtual machines during contacts between nodes.

Mobile network virtualization. The virtualization of mobile networks usually refers to the virtualization of mobile infrastructure, such as cellular or wireless local area networks [6, 7]. The virtualization allows multiple tenants to use a common substrate mobile network. Mobile network virtualization decouples the cellular infrastructure from the services it provides by sharing the infrastructure and radio spectrum resources. Different services can then coexist on the same underlying infrastructure, maximizing its utilization. Cellular network virtualization benefits to Mobile Virtual Network Operators (MVNOs) that lease slices of networks operated by infrastructure providers. These virtual carriers can deploy and offer their own services (*e.g.,* VoIP, video conference, live streaming), in addition to voice services for their own customers. Instead, our work focuses on virtual machine migration between physical mobile nodes in the context of *opportunistic* networks. In our use case, the connectivity of virtual machines inherits the connectivity of the mobile nodes on which they are hosted.

Vehicular clouds. Virtual machine migrations in the context of vehicle-based clouds have been addressed in the literature [13, 14, 15] as a way of managing the resources shared by the participating vehicles. Typical resources include vehicles computation resources, storage resources, and spectrum resources [16]. The dynamic nature of a vehicular network leads to frequent changes in the topology.

Refatt *et al.* propose the VVMM (*Vehicular Virtual Machine Migration*) framework which includes three extensions for selecting the target vehicle of a migration: randomly, depending on the load of the target vehicle, or depending on its location [15]. Nevertheless, none of those extensions take into consideration the suitability of a contact by means of its capacity or its duration though they have a direct effect on the success of a migration. What is more, the VVMM framework relies on road side units (RSU) where the virtual machines are temporarily stored if a vehicle fails in finding a target vehicle.

Yu *et al.* [13, 14] propose a hierarchical cloud architecture for vehicular networks organized in three layers. The first layer is a vehicular cloud that consists of a group of vehicles and a cloud controller that manages the vehicle resources. The second layer is a roadside cloud that consists of dedicated local servers; the vehicles can directly access these servers through roadside units that act as *cloudlets* [17] to which the servers are attached. The third layer is a central cloud that consists of dedicated servers, which vehicles can access using the roadside infrastructure or a cellular network. To handle virtual machine migrations from vehicles to the roadside cloudlets or between cloudlets, the authors propose a re-allocation scheme which prevents a virtual machine from migrating to maintain the performance of the virtual machines already allocated in the target host.

Further details on virtual machine migration between the various cloud layers are given in a companion paper [14]. The authors borrow the pre-copy approach originally proposed by Clarke *et al.* [18] and the dirty page transfer strategy used in Xen [19] to support the live migration of virtual machines. However, the mechanisms they propose can only handle VM migrations from a vehicle to a cloudlet connected to a roadside unit without considering the capacity of the contacts. Additionally, the placement of the cloudlets is not addressed, though it may affect the VM migration performance if cloudlets are not placed in well-frequented locations for the vehicles participating to the vehicular cloud.

3. Virtual large-scale vehicular networks

3.1. Business model and roles

We propose to virtualize vehicular network substrates by decoupling the role of the party that manages the network infrastructure and runs the services into two separate entities: mobile infrastructure provider (MInP) and mobile service provider (MSP). We adapt the terminology from Chowdhury and Boutaba's survey to differentiate the mobile case we consider from the fixed case [3]. Refer to Figure 1 for a schematic representation of the main differences between the fixed and mobile architectures.

Mobile Infrastructure Providers (MInP) are in charge of managing the underlying physical resources of the mobile networks. The physical resources are the compute, storage, and sensing capabilities of the mobile nodes they manage. The MInP offers these resources to multiple tenants by virtualizing the underlying physical substrate. The MInP slices the resources of the mobile physical nodes into a collection of virtual machines and allocates a set of virtual machines hosted on (possibly different) physical mobile nodes to the tenants. The MInP accommodates the tenants resource demands by following the priorities associated with the tenants and their demands, and according to the resources available when the demands are issued. Updates to allocation decisions and re-allocations are required to maintain a high utilization of the underlying resources in face of changes in the substrate or to accommodate new demands from the tenants. The new allocations of the resources may result in changes in the virtual topology, and virtual machines must be moved from one physical node to another. The MInP triggers virtual machine migrations from one vehicular node to another through vehicle-to-vehicle (V2V) communications between the nodes.

Mobile Service Providers (MSP) create virtual networks by aggregating the resources (virtual machines) provisioned by the MInPs. The mobile service provider is then in charge of managing the resources of virtual machines to deploy and offer services to end users. For instance, an MSP can deploy a crowdsensing network on virtual machines hosted on vehicles to sense various parameters (*e.g.,* pollution or noise) throughout the city.

Mobile End Users use the services offered by MSPs. To enhance their experience, the users may leverage different MSPs offering similar services and connect to multiple virtual networks. It is worth mentioning that MInPs do not have access to end users' personal wireless devices. This avoids the dual role of infrastructure and end user at the same time, which would introduce additional complexity in the model.

3.2. Target use cases

Virtualizing vehicular network substrates enables the deployment of various services. In the following, we describe two possible target use cases.

Crowdsensing application. In this first use case, a service provider can take opportunistic advantage of the daily trips made by car to measure the environmental conditions in a specific area. Such service involves cars equipped with computation, storage, sensing and communication resources and the environmental conditions include data such as air temperature, noise and pollution levels, or traffic congestion. The resources of the cars participating to the monitoring service are virtualized so they can host a set of virtual machines in charge of monitoring a specific target area. In this use case, a virtual machine may need to migrate from one vehicle to another, therefore the virtual machine can remain in its area of operation. In this paper, we aim at showing the feasibility of migrating virtual machines between two vehicles in contact.

Vehicle as part of the Fog computing paradigm. In the case of a bus transit network, buses can be equipped with on-board computing resources made available to their passengers who can offload processing-intensive tasks from their handled devices including smartphones and tablets. Tasks to offload include AI or image processing for instance. Computational offloading was shown to improve performance and reduce energy consumption of the mobile terminals [20]. Alternatively, the bus can act as a proxy to cache popular content (e.g., news media or popular videos) and serve it to the passengers in lieu of using cellular connections to fetch the content. The buses as part of the Fog computing paradigm enable local interactions with the passengers, with low latency and high bandwidth [2].

3.3. Architecture, entities, and virtual machine migration

As previously mentioned, the architecture features three parties: (i) the Mobile Infrastructure Providers that manage the mobile physical nodes, (ii) the Mobile Service Providers that aggregate resources allocated by the MInPs into virtual networks, and (iii) the end users who use the services offered by the MSPs.

The MInP features a centralized architecture with a central controller in charge of managing the mobile network substrate. The controller receives demands from MSPs for resources in their network substrate. The demands depend on the services the MSPs aim to deploy (*e.g.,* collection and aggregation of data when *crowdsensing* or security application in a VANET context). The MInP translates the demands into allocation rules so as to match the requirements. The controller maintains a holistic view of the mobile network's current state. The control channel between the controller and the mobile nodes transmits data through a fixed network infrastructure (*e.g.,* cellular network). We assume the control channel is available at all times. Periodically, or upon an arrival or a departure of an MSP, the controller retrieves the state of the mobile nodes, and, with the information retrieved as an input, computes an allocation of the physical resources that best accommodates the current and new demands of the MSPs.

Challenge: Virtual machine migration. The output of the allocation for each MSP consists in a set of virtual machines hosted on a subset of mobile nodes of the mobile network substrate. If the current allocation of the virtual machines on the mobile nodes does not match the output of the allocation procedure, the controller triggers virtual machine migrations on selected physical nodes. The virtual machine migration procedure consists in transferring the whole virtual machine from the physical node where it is hosted to a target physical node that will in turn host the virtual machine. The data transferred during the virtual machine migration includes the operating system, the running processes, and the forwarding states. With a large virtual machine size, it would be impractical to transfer it through the control channel or another cellular infrastructure. Instead, we leverage device-to-device communications between the mobile nodes. A contact occurs when two nodes are in each other's communication range. Figure 2 describes the virtual machine migration procedure between two mobile nodes through V2V communication. The overall architecture of our system is depicted in Figure 3. In this figure, we show the specific locations where virtual machine migrations can be triggered when the MInP is operating a bus transit network. In Section 4, we evaluate the feasibility of such an approach with real-world traces.

4. Methodology and case study

In this section, we first present a generic methodology for analyzing contact opportunities in urban scenarios where mobility traces are available for various types of vehicles. Our methodology draws on the results presented for various cities including San Francisco and Warsaw where it has been shown that vehicles come in contact more often and for longer period of time in specific areas [21, 22]. By limiting the virtual machine migrations to those specific areas, our service can benefit of an improved ratio of successful migrations. To evaluate its benefits, we apply our methodology in a case study involving VM migrations among the public transit buses of the City of Dublin.

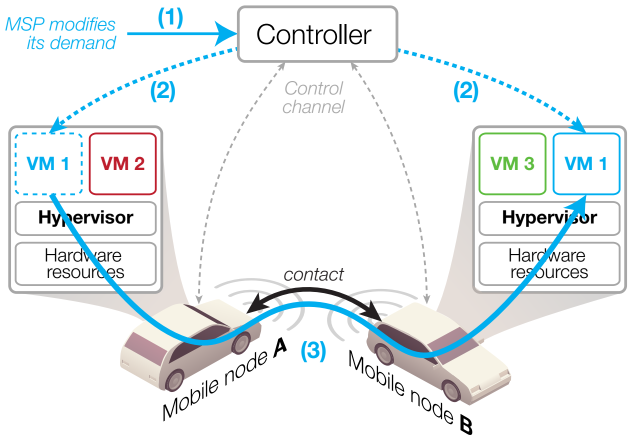


Figure 2: Virtual machine migration between two mobile nodes in contact. (1)-(3) represent a VM migration procedure between the two mobile nodes A and B operated by the same MInP. A tenant (MSP) initially owns a virtual machine VM 1 hosted on mobile node A. (1) The MInP receives a VN request which requires the dynamic reallocation of the resources on nodes A and B: VM 1 has to move from node A to node B. (2) The MInP via its controller notifies the two nodes to initiate the transfer during their next contact. (3) When there is a contact, the VM is transferred from node A to node B.

4.1. Methodology

The objective of the methodology we here describe is to isolate the specific geographical areas, we will refer to as *hotspots*, where vehicles come in contact more often and for longer period of time compared to other areas. By migrating the VMs at the hotspots, the likelihood of a migration succeeding is higher compared to attempts in other locations. To this end, our methodology consists first in processing the mobility traces with the purpose of generating a heatmap representing graphically the spatial variation of the contact density. The next steps consist in determining the location of the hotspots matching specific values regarding the duration and the rate of the contacts occurring in those areas.

Contact heatmap. In this first step, we compute the spatial variation of the contact density. We start by dividing the geographical area of interest in a grid pattern where square cell has a fixed size, e.g. 100m by 100m. In each cell, we plot the only relevant contacts, which are in our scenario, the contacts lasting for at least 200 s which amounts for the duration needed to transfer a typical virtual machine of 200 MB. We then characterize the contact density of each cell by using a kernel density estimator which consists in clustering the points representing the contacts occurring in the cell. Larger numbers of clustered points result in larger density values [23]. The heatmap obtained for the city of Dublin is shown in Figure 4b where different contact densities are depicted using different shades of red, the darker shade indicating a higher density of contacts. We notice that the contacts between buses are concentrated in the city center, but also on the outskirts of the city center, along the major traffic arteries and at some of the main intersections of Dublin. The next steps of our methodology consist in deriving the hotspot locations from the contact density heatmap.

Contact hotspots. Firstly, we rank the cells according to their contact density and select the 25% top cells. The purpose is to exclude cells where contacts are not occurring in significant

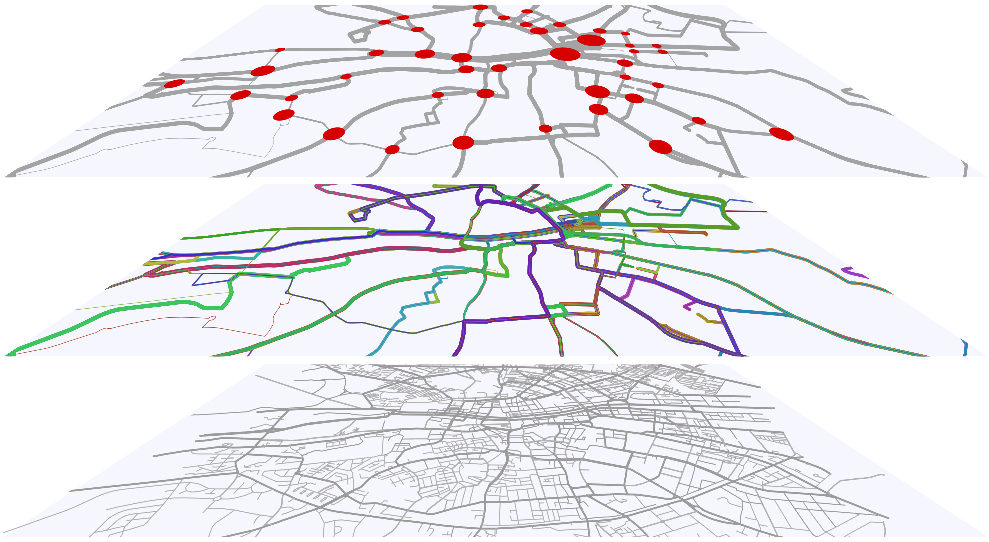
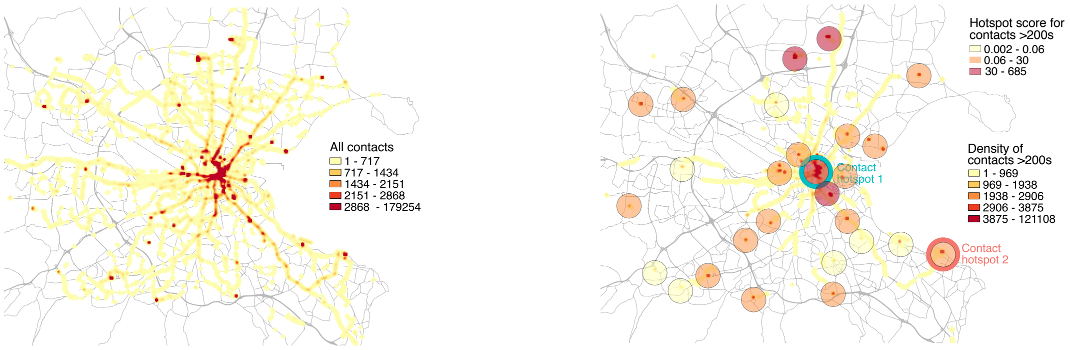


Figure 3: Virtualization of the bus transit system of Dublin, Ireland. The buses are supposed to be equipped with computational and storage resources. The bottom layer shows the street map of Dublin city center. A MInP is appointed to operate and manage the on-board resources of buses running on one or more lines depicted in the middle layer. The dots on the top layer indicate the locations where virtual machines belonging to the same MSP may migrate between buses. These locations match the intersections between lines operated by the MInP or other places such as road intersections or bus depots where buses are likely to come in contact with each other.

numbers. We then aggregate the contiguous cells in clusters and draw the hotspots as circles centered so they can cover the corresponding cluster. The diameter of the hotspots is fixed and selected so the largest cluster is fully covered by a single hotspot.

The hotspots depicted in Figure 4b are further characterized from the perspective of our VM migration service. For each hotspot, we compute the two following variables: The total number of contacts with a duration of at least 200 s (n) and the average inter-contact duration (x). The inter-contact duration refers to the duration between two consecutive contacts occurring within the same hotspot. Those two variables are then used to assign a score to each hotspot we compute as: ${}^{n}/_{\max(1,x)}$. Finally, we use the Jenks optimization method to identify three value ranges which provide the best arrangement of the hotspots according to their scores [24].

In Figure 4b, we use different shades of color to represent the hotspots according to their classes. For the purpose of the performance evaluation, we select two significant hotspots, each belonging to the two higher classes: “Hotspot 1” has the highest score which indicates that long-duration contacts occur with high frequency. Note that three other hotspots share a similar score. The second hotspot indicated as “hotspot 2” is selected arbitrarily among the 16 occurrences sharing a similar score.



(a) Heatmap of the contacts of all durations in Dublin, Ireland.

(b) Heatmap of contacts whose duration is greater than 200 seconds and the hotspots centered at location centered at the aggregated 25 % densest cells.

Figure 4: Heatmaps of contacts between buses in Dublin, Ireland. Out of all the possible hotspots, we choose the two represented on the maps. The figures denote the number of contacts per cell (of size 100m × 100m).

4.2. Case study

We assess the feasibility of migrating virtual machines in the context of the public transit system of Dublin city. In our simulations, we focus on the capacity of the opportunistic contacts between buses while in transit.

Experimental setup. For our feasibility study, we use a publicly available dataset of real traces of Dublin city buses provided by Dublinked, part of the Dublin City Council. The mobility traces span the month of January 2013 and provide timestamped GPS coordinates of all DublinBus buses in service [25]. We analyze the movements of 823 buses during the service day of Tuesday, January 29th, 2013 from 10am to 1pm. We assume that two buses are in contact whenever the buses are in each other's communication radius denoted R.

We use the ONE, a discrete-event simulator to run our experiments [26]. The ONE can use real-world mobility traces to reproduce the movements of DTN nodes and infer their contacts. We used the default settings of the ONE to simulate IEEE 802.11 on the mobile nodes. However, the only layers supported by the ONE simulator are the network layers and above. To account for the properties of the physical layer including the link-level packet losses, we select appropriate values regarding the node communication radius and contact transfers goodput. We draw on the results presented by Rubinstein *et al.* which show a 1 Mbps plateau in the TCP goodput when the contact distance between vehicles is less than 150 meters [27]. The choice of a conservative node radius is further supported by Hadaller *et al.* who showed that the Packet Delivery Ratio (PDR) measured at the MAC layer remain stable between vehicles in contact within a similar distance of 150 meters [28]. In our simulations, we considered a conservative communication radius R of 150 meters with an homogeneous average goodput of 1 MB per second.

VM migration simulations. In the remainder of this section, we simulate the migration of virtual machines of various sizes running on every bus. A bus entering a specific hotspot initiates the migration of a virtual machine with the first bus it comes in contact with,

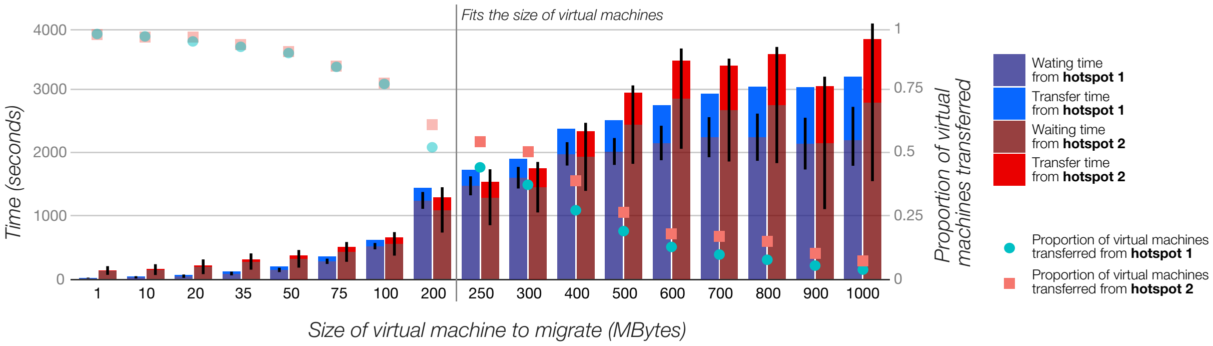


Figure 5: Proportion of successful virtual machine migrations and mean time to transfer the virtual machine as a function of the virtual machine size. The black solid bars represent the 95% confidence interval for the waiting time before successfully transferring the virtual machines.

following the First Contact routing policy implemented by the ONE simulator. We make no assumption regarding the capacity of the buses to predict the duration of a contact. A transfer will be aborted in case the contact duration is not long enough to accommodate the transfer. Hence, multiple attempts may be needed before the migration succeeds. The migration is considered as failed if the bus exits the hotspot without succeeding in transferring the virtual machine.

The results we present reflect averages over 100 virtual machine migration trials for different sizes of virtual machines we try to migrate at each of the two hotspots introduced in Figure 4b. The range of values we use captures the usual sizes of virtual machines that can be found, that is, from a few hundreds of kilobytes (*e.g.*, TinyOS [29]) to a few hundreds of Megabytes [18]. For each size of virtual machine, we measure the average time needed for a bus entering each hotspot to find a suitable contact and the time it actually takes to migrate the virtual machine. The plot in Figure 5 shows the mean time to transfer virtual machines of different sizes, as well as the proportion of virtual machines that were actually transferred.

First, we note that the ratio of virtual machines successfully migrated at both hotspots decreases as their size increases. This is due to the lack of enough long-lasting contacts between buses. Buses fail several times before finding a suitable contact to transfer the virtual machine. This observation is confirmed by the time spent waiting for a contact suitable in duration with large-sized virtual machines. Overall, it takes less time to find a suitable contact at hotspot 1 than at hotspot 2. Buses entering hotspot 1 spend less time waiting for a suitable contact in comparison with hotspot 2 expect for virtual machine sizes ranging from 200 to 400 MB. To further explain this trend, we plot in Figure 6 the total number of contacts measured for different contact durations as well as the ratios of those contacts against the total number of contacts occurring in hotspots 1 and 2. We can see that hotspot 2 has a higher ratio of contacts lasting long enough (e.g. 200 s and 400 s) to accommodate transfers of virtual machines with a size ranging from 200 MB to 400 MB. Despite the higher number of contacts at hotspot 1, most of them are not suitable for the transfers of such virtual machines. Buses spent more time trying transfers that

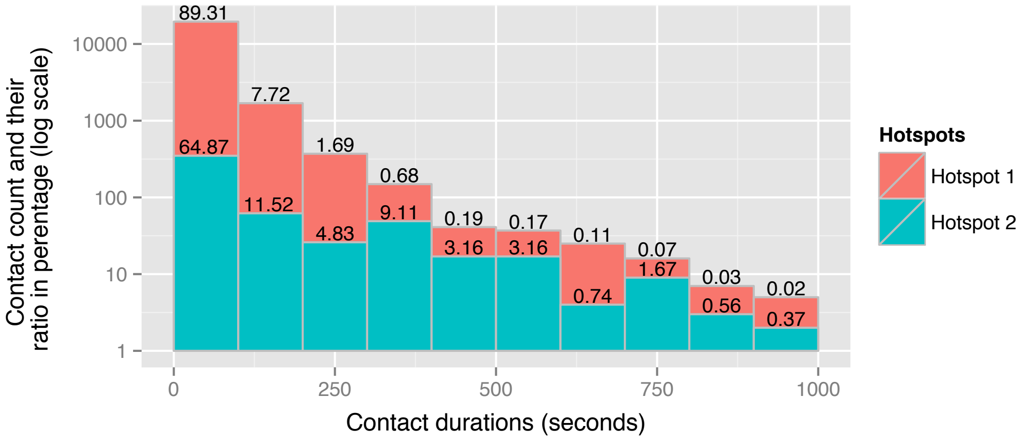


Figure 6: Number of contacts in log scale accounted for different duration ranges and their ratios (in percentage) against the total number of contacts in the hotspots 1 and 2, respectively.

will eventually fail before finding a suitable contact. This also explains why the ratio of successful migrations is higher at hotspot 2 when virtual machines have sizes ranging from 200 to 400 MB.

In summary, our evaluations showed that virtual machine migrations can be triggered at specific locations where contacts between mobile nodes are more likely to happen. Thus, even with few active buses, transfers of several hundreds of Megabytes can be achieved through V2V communications, enough to accommodate migrations of virtual machines.

5. Challenges involved with mobile opportunistic network virtualization

In this section, we overview the challenges and open issues related to the virtualization of *mobile opportunistic networks*. Mobile opportunistic networks, also known as delay/disruption tolerant networks (DTN), feature geo-distributed mobile nodes. The nodes communicate with each other whenever they encounter each other (when in "contact"), resulting in intermittent connectivity. Thus, at any given time, there may or may not exist a path between a source and a destination. Therefore, the main challenge with DTN over traditional ad-hoc network is to cope with the intermittent connectivity between the nodes. Specific routing protocols have been designed and rely on store-carry-and-forward policies at the nodes [30].

In the following, we discuss some of the open issues involved with the virtualization of mobile opportunistic networks.

Intermittent connectivity. The resources allocated by the MInP are hosted on the mobile physical nodes. The physical nodes feature intermittent connectivity, as they can only exchange data when they are in each other's transmission range. In this paper, we leverage the mobility of mobile nodes to migrate virtual machines as a way of maintaining a high utilization of the underlying resources. Virtualization can be pushed further by considering the communications between virtual machines. The MInP must provide connectivity between the virtual machines allocated to the tenants. The resulting opportunistic virtual networks have to cope with the characteristics inherited from the substrate network. Since we leverage V2V communications, the substrate network is partitioned and mobile physical nodes are intermittently connected to each other, limiting the availability of end-to-end paths at all times. Traditional ad-hoc routing protocols, as AODV or OLSR cannot be used, as they suppose the network to be constantly connected. Therefore, MInPs must implement delay tolerant routing protocols to transmit the data from one end of the virtual link to the other, while satisfying the MSPs' guarantees. In turn, the MSP is free to implement custom control and data plane that may or may not take into account the intermittent connectivity of the virtual machines.

Find suitable contacts. In order to successfully transfer a virtual machine between two vehicles, the contact must be long enough to accommodate the size of the virtual machine to transfer. In our evaluation, we proposed a methodology to determine the geographical areas we refer to as hotspot where long-duration contacts occur at significant ratios and rates: Hotspots are located in geographical areas within which all contacts amount for 75% of all 200 second long contacts and occur at most every 1000 seconds. However, although long-duration contacts occur statistically more often in the hotspots compared to other areas, there is still a high density of short-duration contacts (as shown in Figure 4.b). Depending on the first contact a bus experiences when entering a hotspot, the VM migration may fail if the first contact is not long enough to accommodate the target virtual machine size. One way of improving the migration success may consist for the buses in implementing prediction models regarding the contact durations. Many existing techniques may be borrowed from the DTN literature. For instance, this could be done by maintaining a list of contact with their corresponding durations for each vehicle encountered [31, 32] or by exchanging the direction and speed or itinerary between vehicles in contact to estimate the shared route with the other vehicle [33, 34].

Scalability. The mobile network virtualization enables the coexistence of multiple virtual networks managed by different tenants with different requirements. Mobile infrastructure providers must support an increasing number of coexisting virtual networks, without affecting the performance of the allocated ones. The MInP allocates the resources to MSP with a central controller that maintains a holistic view of the substrate network infrastructure managed by the MInP. The allocation may trigger virtual machine migrations between the mobile nodes. They are achieved through opportunistic device-to-device communication.

Manageability of the resources. Managing the resources is a mandatory control task for both the MInP and MSP. In particular, the MInPs must manage the mobile nodes of the substrate network to allocate the demands issued by the MSP. This task could be performed by a control plane deployed in a centralized controller. On the other hand, the MSPs must be able to access the virtual machine provisioned by the MInPs to deploy and manage services. Therefore, a control channel must be available at all times to have a continuous access to the mobile physical nodes and the virtual machines they host. The control channel must be made accessible to both the MInPs and MSPs. The communications between the controller and the vehicles through the control channel could rely on a SDN-based protocol such as

OpenFlow [10].

Hardware heterogeneity. As noted by [3], heterogeneity has two origins: (1) from the underlying technologies of the mobile substrate (e.g., in terms of processing, storage and networking capabilities), and (2) from the virtual networks of the MSPs that may span multiple MInPs. The underlying virtual machines must support heterogeneous routing protocols and algorithms the MSPs may wish to implement. Therefore, the MSPs must have access to the forwarding states of the virtual machines.

Isolation. Multiple tenants (MSPs) coexist on the same substrate network with their own virtual networks. The resources must be shared equally between the tenants, depending on their requirements. The hypervisor at each mobile node ensures the isolation of virtual machines allocated to the MSPs on each physical mobile node. The control channel must isolate the MSPs that wish to manage and control their virtual resources when they are hosted on a shared physical resource. The isolation must also prevent starvation when multiple MSPs allocated on the same physical mobile node wish to transmit at the same time over a contact opportunity. Starvation can be prevented through fair scheduling algorithms, which take into account service priorities expressed in the MSP demands. Another possibility is to guarantee the isolation of MSPs at the wireless physical level by avoiding interferences among the different MSPs when two or more want to communicate. The MInP may use special equipment, such as directional antennas or dedicated wireless channels.

Adaptivity. MSPs may issue new demands to MInPs or change their current demands. To this end, the MInP must determine whether the demands can be satisfied before serving them. An approach to achieve this is to check the current state of the available and allocated resources through the control channel before allocating new resources. If the demand can be allocated or modified, the MInP re-runs an allocation algorithm that triggers new slicing of the resources of the nodes, as well as virtual virtual machine migration between the mobile nodes.

6. Conclusion

In this paper, we studied the problem of virtual machine migration that arises when virtualizing the resources embedded in the mobile nodes of a large-scale vehicular network. To solve this problem, we devised a centralized architecture to manage and allocate the mobile resources to tenants. A tenant deploys a service onto a collection of virtual machines allocated by a infrastructure provider in charge of virtualizing the mobile infrastructure. Migrations of virtual machines between remote physical nodes may be required to (re)allocate the resources of one node to another tenant. We proposed to migrate VMs via V2V communications instead of relying on cellular networks because of the size of virtual machines to transfer. The migration procedure is not straightforward and relies on the opportunistic contacts between mobile nodes as a result of network intermittent connectivity. We simulated an opportunistic network of buses using real world traces. We concluded that such a scenario can accommodate transfers of several hundreds of Megabytes, sufficient for virtual machine migrations.

As future work, we propose to conduct field experiments to evaluate the capacity of a contact between mobile vehicles with 801.11p interfaces. We could also investigate the allocation procedure of the physical resources for the tenants, as it is particularly challenging in networks featuring intermittent connectivity. As a natural extension to this work, we can design and evaluate a complete virtualization system of opportunistic networks, which would include the resources hosted on the mobile nodes, as well as the opportunistic contacts between the nodes.

References

[1] Y. Xiao, P. Simoens, P. Pillai, K. Ha, and M. Satyanarayanan, "Lowering the barriers to large-scale mobile crowdsensing," in *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, ACM, 2013.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[3] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE*, 2009.

[4] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A network virtualization layer," *OpenFlow Switch Consortium, Tech. Rep*, 2009.

[5] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, G. Parulkar, E. Salvadori, and B. Snow, "Openvirtex: Make your virtual sdns programmable," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ACM, 2014.

[6] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, "Nvs: a substrate for virtualizing wireless resources in cellular networks," *Networking, IEEE/ACM Transactions on*, 2012.

[7] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri, "Splitap: leveraging wireless network virtualization for flexible sharing of wlans," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, IEEE, 2010.

[8] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1888–1906, 2013.

[9] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move: live router migration as a network-management primitive," in *ACM SIGCOMM Computer Communication Review*, ACM, 2008.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[11] S. Lo, M. Ammar, and E. Zegura, "Design and analysis of schedules for virtual network migration," in *IFIP Networking Conference, 2013*, IEEE, 2013.

[12] E. Keller, S. Ghorbani, M. Caesar, and J. Rexford, "Live migration of an entire network (and its hosts)," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, ACM, 2012.

[13] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang, "Toward cloud-based vehicular networks with efficient resource management," *Network, IEEE*, vol. 27, no. 5, pp. 48–55, 2013.

[14] R. Yu, Y. Zhang, H. Wu, P. Chatzimisios, and S. Xie, "Virtual machine live migration for pervasive services in cloud-assisted vehicular networks," in *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pp. 540–545, IEEE, 2013.

[15] T. K. Refaat, B. Kantarci, and H. T. Mouftah, "Dynamic virtual machine migration in a vehicular cloud," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pp. 1–6, IEEE, 2014.

[16] M. Gerla, "Vehicular cloud computing," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, pp. 152–155, IEEE, 2012.

[17] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[18] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, USENIX Association, 2005.

[19] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[20] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?," *Computer*, no. 4, pp. 51–56, 2010.

[21] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," in *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, pp. 110–118, ACM, 2004.

[22] N. Sarafijanovic-Djukic, M. Piorkowski, and M. Grossglauser, "Island hopping: Efficient mobility-assisted forwarding in partitioned networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, pp. 226–235, IEEE, 2006.

[23] B. W. Silverman, *Density estimation for statistics and data analysis*, vol. 26. CRC press, 1986.

[24] G. F. Jenks, "Generalization in statistical mapping," *Annals of the Association of American Geographers*, vol. 53, no. 1, pp. 15–26, 1963.

[25] "Dublin bus gps sample data from dublin city council (insight project)." `http://dublinked.com/datastore/datasets/dataset-304.php`. Accessed: May 2, 2016.

[26] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[27] M. G. Rubinstein, F. Ben Abdesslem, M. Dias de Amorim, S. R. Cavalcanti, R. dos Santos Alves, L. H. M. K. Costa, O. C. M. B. Duarte, and M. E. M. Campista, "Measuring the capacity of in-car to in-car vehicular networks," *Communications Magazine, IEEE*, 2009.

[28] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, ACM, 2007.

[29] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, *et al.*, "Tinyos: An operating system for sensor networks," in *Ambient intelligence*, Springer, 2005.

[30] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *Communications Surveys & Tutorials, IEEE*, 2006.

[31] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[32] M. Grossglauser and M. Vetterli, "Locating mobile nodes with ease: learning efficient routes from encounter histories alone," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 3, pp. 457–469, 2006.

[33] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "Geodtn+ nav: geographic dtn routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[34] N. Banerjee, M. D. Corner, and B. N. Levine, "An energy-efficient architecture for dtn throwboxes," in *INFOCOM 2007. 26th IEEE international conference on computer communications. IEEE*, pp. 776–784, IEEE, 2007.