Analytical modeling of context-based multi-virtual wireless mesh networks

Ricardo Matos [a,*], Carlos Marques [a], Susana Sargento [a], Karin Anna Hummel [b], Harald Meyer [b]

[a] Instituto de Telecomunicações, Universidade de Aveiro (IT-UA), Aveiro, Portugal
[b] University of Vienna, Department of Distributed and Multimedia Systems, Vienna, Austria

ARTICLE INFO

Article history:
Available online 17 May 2011

Keywords:
Context-awareness
Wireless mesh networks
Virtual Networks
Modeling
User connectivity
Discovery mechanisms

ABSTRACT

The flexibility and self-$*$ properties of wireless mesh networks (WMNs) and the programmable management of network resources brought by the innovative network virtualization techniques, are a twofold force to enable personalized access over wireless environments. Therefore, we design a context-aware multi-virtual architecture for WMNs to deal with the requirements of mesh clients and their applications (context can be defined as a set of requirements such as cost, security, mobility, applications' Quality-of-Service - QoS). In this approach, a WMN is split into several adaptable Virtual Networks (VNs), each one appropriate to specific levels of context. This approach requires the proper configuration of VNs' topologies and resources, and the definition of local and global (distributed) mechanisms to reconfigure VNs that best fit users' requirements. In this paper, we propose an analytical model to evaluate the impact of network virtualization and the complexity of the discovery and extension mechanisms defined for VN reconfiguration. Through a delay-based approach, we show the effectiveness of the architecture to deal with different communication requirements and with distinct scenarios for user connectivity establishment, even in the presence of user mobility or using a real WMN topology. The analytical model is compared against a simulation one, showing similar results.

1. Introduction

Wireless mesh networks (WMNs) [1] are easily adaptable, self-configuring and self-organizing networks, being a key technology to enable broadband access in wireless environments. WMNs provide access to mobile mesh clients that have, for instance, different trust, security, mobility, and cost preferences. Moreover, the services required by such clients have distinct Quality-of-Service (QoS) requirements. These preferences and requirements are denoted as context [2] of users and services. In order to deal with these context demands, WMNs have already shown high-potential since the topology of their infrastructure is flexible enough to promote the switching of routes and transport connections for different context-aware purposes. However, the integration of the plethora of context features in the management and control of WMNs is a topic not fully addressed in the literature.

The application of network virtualization techniques [3] to commodity network infrastructures brings attractive advantages. It is increasingly seen as a clear path for the simultaneous support of emerging paradigms and architectures. Through network virtualization, a physical infrastructure can be split into a number of different specific Virtual Networks (VNs). This concept can be used to personalize networks to users' context up to a high degree. Although network virtualization brings benefits for different types of networks, its suitability, when applied to WMNs, has not been extensively evaluated.

* Corresponding author.
E-mail addresses: ricardo.matos@ua.pt (R. Matos), car.manuel@ua.pt (C. Marques), susana@ua.pt (S. Sargento), karin.hummel@univie.ac.at (K.A. Hummel), harald.meyer@univie.ac.at (H. Meyer).

1570-8705/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2011.05.004

The heterogeneous structure of WMNs, containing a large number of nodes handling different context requirements with several conflicting targets, can benefit from the separation of mesh clients in specific, programmable, and adaptable VNs. Moreover, the application of network virtualization to WMNs will certainly play an important role in the creation, maintenance, extension, and termination of communication flows in future wireless environments, increasing the users' Quality-of-Experience (QoE).

In this scope, we define an architecture that takes advantage of context-awareness and network virtualization, integrating these two concepts for WMNs to enable personalized communications for a group of users, according to their expectations and preferences. Each VN is configured to meet distinct levels of context requirements of users and their services, providing personalized, isolated, and non-interfering context-based communications in WMNs. In our approach, it is required to properly configure the assigned resources and topologies of the context-based VNs, and to define local and global control mechanisms to discover and extend VNs that best fit users' context.

We described the architecture and its key raised challenges first in [4]. Basic approaches for the evaluation of the architecture were presented in [5,6], and [7]. In [5], we performed a very simple probabilistic-based evaluation of the limits of the multi-VN approach in terms of the number of VNs, attached mesh clients, and virtual nodes per VN, in order to accomplish feasible delays for different data communication requirements. In [6], we analytically evaluated basic control mechanisms to discover and extend fitting context-aware VNs for users. More recently, we developed a simulation framework to evaluate the architecture, and the first results were presented in [7].

In this paper, we present the complete analytical model of the proposed context-based multi-VN architecture. Additionally, we evaluate and validate the model against the results of the implemented simulation framework. The model is used both in data and control planes. In the data plane, it allows the estimation of the end-to-end delay of a data packet within a specific context-based VN, even in the presence of multiple VNs and flows per VN. We aim to assess the suitability of network virtualization to personalize context-based communications in this type of environments. In the control plane, it provides the measurement of the impact (delays and overhead) of the proposed control mechanisms for VNs' association and reconfiguration in case of mobility or arrival of new users at the WMN: (*i*) the discovery of the VN that best fits users' context, and (*ii*) the extension of the best fitting VN to the nodes near the users. This work also describes the control functionalities of every architectural element and the required control signaling to associate users to fitting VNs. Finally, and as a proof of concept of the modeling work, we also perform simulations with a real WMN topology, using the node locations of the Funkfeuer Vienna WMN [8].

The obtained results show that the requirements of the data communications are indeed met, with a small impact of the proposed control mechanisms for VN reconfiguration both in terms of delays and overhead. As expected, mobility of users has a significant impact on the communications, and this impact depends on the required level of reconfigurations. Finally, the results obtained with the topology of a real WMN are similar to the ones of a specified WMN grid topology, still with a slight decrease on the delay of data communications. This shows that this approach is suitable for real running networks.

This paper is structured as follows. Section 2 summarizes the literature proposals that are related with this work. Then, Section 3 introduces the context-based multi-virtual architecture for WMNs, and the required control mechanisms to enable users' connectivity to the best fitting available VNs. Following, Section 4 presents the analytical model to evaluate the architecture and the proposed control mechanisms. Section 5 starts with the description of the simulation environment, and then presents the analytical results and their comparison against simulation results. Finally, Section 6 presents final conclusions and future work.

2. Related work

The envisioned architecture makes use of two fundamental concepts, that are context-awareness and virtualization to personalize WMNs. Context-awareness refers to the ability of the WMN to incorporate information about the user, user behavior, user preferences, and environmental conditions. The description, detection, and use of context are the prerequisites to propose context-based selection of appropriate network services and adaptations. To ease adaptation and flexible resource sharing, and more important, to be able to build and use on-demand networks with different networking mechanisms and protocols, network virtualization is applied to WMNs. Finally, since we investigate our approach based on analytical modeling for WMNs, this section also discusses related WMN modeling approaches.

2.1. Context-awareness

Context-awareness in networking takes advantage of information about terminals' or users' context such as the Quality-of-Service (QoS) of users' applications, mobility behavior, and cost, security and privacy preferences. The objective of using context information is to adapt routing or flow control, network selection, mobility management, or resource reservation in WMNs.

Several QoS-based routing protocols have been proposed for WMNs [9,10]. In [9], the levels of signal interference of the communication links are dynamically integrated in the routing metrics, whereas in [10], several context data (security, traffic priority, mobility, etc.) are used to describe and adapt the routing metrics. In [11], performance metrics, such as the path availability, bandwidth, and delay, as well as outages based on network monitoring and statistical evaluation, are used to assess the robustness and perform the selection of the best available path for a real-time communication. Furthermore, several works are in particular aware of mobility behavior of mesh clients [12]. In [13], a proactive and reactive routing algorithm is selected according to the intensity of mobility within the WMN. On the other hand, in [14], the current position of mesh clients is taken into account by channel assignment and router selection. Concerning security constraints, [15] addresses authentication and key exchange mechanisms within the context of WMNs.

While in related work the network resources are selected according to general network characteristics, in our approach, single users' preferences are taken into account. These users' (context) preferences and their applications' requirements are modeled as multivariate context data and automatically mapped to corresponding (virtual) network structures (appropriate connections and paths). Context information is also used to define control mechanisms that improve the performance of WMNs, enabling the adaptation of the WMN in the presence of context changes, dynamics of networks, and mobility of users.

2.2. Network virtualization

Network virtualization can be used for distinct purposes. From one side, it can be used to share and to combine physical resources for reasons of performance improvements of single users. From the other side, it can be used to create networks in a flexible way within the same physical infrastructure that are independent in terms of assigned resources and also in terms of the mechanisms and protocols implemented. In the scope of wireless networking, network virtualization has to face particular challenges [16], where the most important are the dynamic mapping, scheduling, and switching of the virtual channels/links/nodes to the wireless channels/interfaces/routers. This mapping is based on the meta-information of user flows competing for increased throughput. Hereby, wireless interference is a major problem and has to be decreased, while throughput has to be increased on the links.

Concerning the performance improvement of WMN communications through network virtualization, [17] describes an approach of a joint optimization streaming rate allocation of traffic flows and power consumption of links for forwarding data flows in multicast overlays [18], whereas in [19], a wireless ring over a regular WMN is proposed in order to carry high bandwidth data.

In order to split a WMN into a different number of VNs, several solutions and mechanisms are available. Overlay networks proposing "network slicing" in WMNs are presented in [20]. Moreover, MadWifi [21] proposes a scheme to run different virtual interfaces on a single physical interface, in which the virtual interfaces have to share and distribute the physical interface in a non-interfering way, and [22] presents WiSwitcher, which is a wireless client able to connect to multiple access points. In [23], a framework is presented to emulate a WMN using OMNeT++ [24], which can receive traffic from real or virtual mesh nodes.

Besides enabling the support for different networks on the same infrastructure, related virtualization approaches for WMNs provide basic methods for wireless resource sharing. Our approach adds novelty by providing further mechanisms to dynamically create, setup, adapt, and terminate different VNs in WMN environments. As our architecture includes these management and control mechanisms, we provide an analytical study including these functions.

2.3. Analytical modeling of WMN performance

Many interesting analytical models have been proposed to evaluate the performance of WMNs which are related to the approach presented in this paper.

In [25], a model is presented for determining the optimal multi-path flow that minimizes the mean delay in the network, based on a Markov chain queuing model of mesh nodes, and on an interference aware delay analysis between neighbors. In [26], the average end-to-end delay and capacity in random access MAC based WMNs is characterized based on the diffusion approximation method [27]. In [28], a lower bound is derived for the delay-bound violation probability, as well as lower bounds for the average delay and jitter in order to evaluate the WMN multi-hop delay performance under different traffic loads and wireless channel conditions. In [29], a model is presented to define individual node throughput in any arbitrary IEEE 802.11 network topology given traffic arrival rates at each node, with the restriction that each node only transmits to another node within its transmission range. In [30], a model is proposed which accounts for blocking by neighboring nodes and for interference by hidden nodes. The throughput of each link is computed along a path, under the assumption that each node is saturated and attempts to transmit a packet whenever it senses the channel idle.

These works evaluate the WMN performance in different ways. However, they do not consider the splitting of a WMN into a different number of context-based VNs, which requires proper creation, configuration, and adaptation schemes, and therefore a proper modeling of their impacts in the WMN.

3. Multi-virtual architecture for context-based WMNs

This section presents the architecture defined to support context-awareness and virtualization in WMNs. It highlights the main challenges raised by the approach, giving special emphasis on the required control entities, mechanisms, and signaling to enable the context-based association of users to VNs that best fit their requirements.

3.1. Concept overview

Fig. 1 presents the architecture and some of the intrinsic challenges. The WMN, which can be supported by one or more infrastructure providers, is split into a multitude of context-based VNs. These VNs are properly configured to meet specific levels of certain expectations of mesh clients and requirements of their applications, which are designated as context parameters (e.g., security and trust constraints, mobility, QoS requirements, cost preferences). For instance:

- Security and trust constraints of the users of a specific VN have to be considered to choose the authentication, accounting, and authorization protocols of such VN, as well as the encryption and privacy approaches.

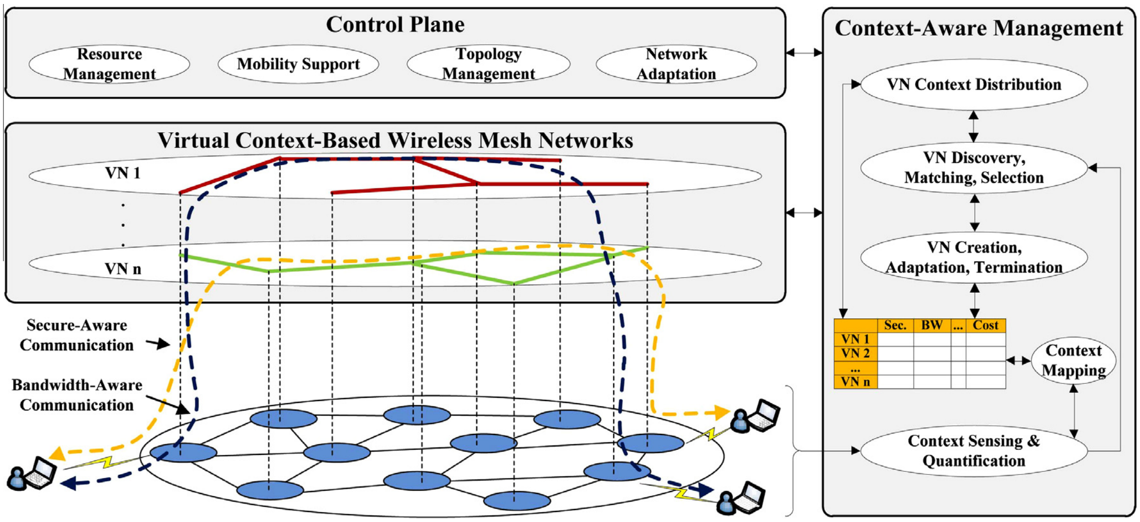

Fig. 1. High-level overview & challenges.

- Mobility of users, or the percentage of users that leave or enter a VN, are important features to choose a reactive or proactive routing protocol for such VN.
- QoS requirements of users' applications (e.g., throughput, packet loss, delay) have a direct impact on the configuration of the bandwidth, buffer size, processing power, and memory of the VN nodes that provide support for such applications.
- Several WMN providers define different VNs with different characteristics and different cost, which needs to be matched against the cost preferences of the users of such VNs.

There are several specific environments that may benefit from the deployment of this type of networks, such as group-based communications, community networks, content distribution. Those environments can span from residential areas, schools, universities or college campus, to business, industrial or tourist attractive areas.

Our main goal is the provisioning of personalized networking for users, which are constantly changing their locations, required services and preferences. Therefore, our architecture is endowed with mechanisms to create, configure, adapt, and terminate the context-based VNs, making them suitable and scalable to dynamically meet changing context demands. Since a user accesses the WMN through the best fitting VN according to his/her context, intelligent mechanisms are required to: (*i*) sense and quantify users' and applications' requirements in specific levels or policies, which are then mapped into appropriate VNs' features, (*ii*) discover, match, and select a fitting VN or virtual node for a user, according to his/her context requirements, and (*iii*) create, configure, and adapt the VNs' topologies and assigned resources, according to VNs' purposes.

This paper concentrates on modeling and evaluating the control mechanisms required to support user connectivity establishment, which comprises the discovery of the best fitting VN for a user according to his/her context requirements, and the extension of such VN to adapt to the user's location. To better understand the modeling approach, the next sub-section details the control entities that compose the architecture, and the required control signaling.

3.2. Users' context-aware connectivity and VNs' reconfiguration

In this sub-section, we define the control mechanisms to dynamically discover and reconfigure fitting VNs for users' association. Such mechanisms may be performed in the node where the user attaches to, or in its physical neighborhood, or even they may require the involvement of more WMN entities.

Note that this paper does not focus on the definition of mechanisms to create VNs on demand, which certainly are more time and resource consuming than the VNs' reconfiguration mechanisms. However, and if there is a high number of users with similar context in a VN that is not able to exactly fit their requirements, a new VN can be created.

3.2.1. Control entities

Fig. 2 describes the main important entities of the architecture and their control functionalities in the scope of the proposed control mechanisms.

(A) WMN node

The WMN node is the key component of our architecture.

In the one hand, it performs routing functionalities within the physical WMN infrastructure for control purposes. Thus, this node has knowledge of its neighbor WMN nodes.

On the other hand, a WMN node is the physical substrate for a set of virtual nodes, and therefore, it stores the context information about its local virtual nodes. Then, after receiving the context requirements of a user that

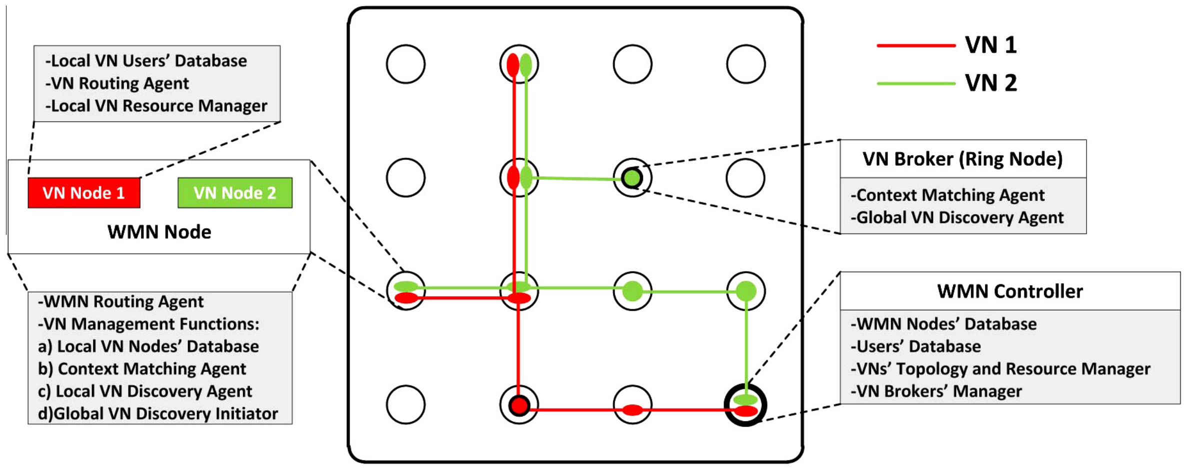

Fig. 2. Control entities and functionalities.

wants to be connected to it, a WMN node quantifies such requirements in specific levels or policies, and performs context matching functionalities to select the best fitting local available VN for the user. If a fitting VN for the user is not available, a WMN node triggers (or is involved in) the discovery of a point of attachment of a fitting VN in the WMN.

For simplicity, this work considers that users are connected to the exactly fitting VNs. In future work, we aim to introduce semantic similarity metrics and thresholds to perform the multi-level and multi-context matching among users' requirements and VNs' features.

(B) VN node

As previously referred, each WMN node is split into several virtual nodes, each one belonging to a specific context-based VN.

Each VN node stores the relevant information about its attached users (e.g., context requirements, users' identifiers), and also performs data routing functionalities inside its VN. Moreover, and based on the profile or connectivity time of its users, a VN node manages its local assigned resources; for instance, it can: (*i*) fairly schedule its assigned wireless resources among the communication flows of its attached users, or (*ii*) assign more local resources' usage to specific communication flows (in this work, we target the first option).

(C) WMN controller

Our architecture is endowed with a control entity, the WMN controller, which stores information about the WMN nodes that are part of the physical infrastructure. Moreover, the WMN controller collects the context information of mesh clients, allowing communications to and from other nodes of the network (inter-WMN communications), or secure communications between users that belong to different VNs (inter-VN communications). For simplicity, this work only considers intra-VN intra-WMN communications.

The most important functionalities performed by the WMN controller are the resource and topology management of the available context-based VNs.

In the one hand, the WMN controller is responsible to schedule, allocate, and switch the wireless channels (and other wireless resources) among the available VNs of the architecture, enabling isolated and non-interfering communications. The dynamic channel allocation approach will be subject to future work.

On the other hand, and intrinsically related with the resource management functionalities, this entity also controls the topology of the available VNs. Due to the dynamics of WMN environments, where mesh clients are constantly changing their requirements and locations, VNs need to be dynamically created, reconfigured, or terminated. In this work, the extension of a fitting VN for a user is performed through the available shortest path, between the user attached node and the nearest VN node from the user's location, with a highest level of available wireless resources. As future work, we aim to enhance the VN topology control algorithm by using other network metrics.

(D) Distributed control framework and VN brokers

Due to the dynamic characteristics of mesh clients, we adopt a distributed control framework to find specific virtual nodes or VNs in the WMN. The challenging requirements raised by our dynamic multi-level multi-context architecture (such as, the need to quickly find an available fitting VN for a user, or to extend such VN to a neighbor WMN node, or even to move the user from one VN to a similar semantic one due to a specific context changing) cannot be easily overcome by using an unstructured-based distributed control solution. Therefore, we propose a structured-based distributed control ring (see Fig. 3), based on semantic-aware and topology-aware Distributed Hash Tables (DHTs). Its most important features are:

- Within each VN, it is selected one virtual node (VN broker) to perform the control and management functionalities inside its VN. In Fig. 3, node *A* is the selected broker of VN_1, *C* of VN_2, *G* of VN_3, *D* of VN_4.
- The set of brokers of the available VNs of the WMN also composes the global distributed control ring structure.

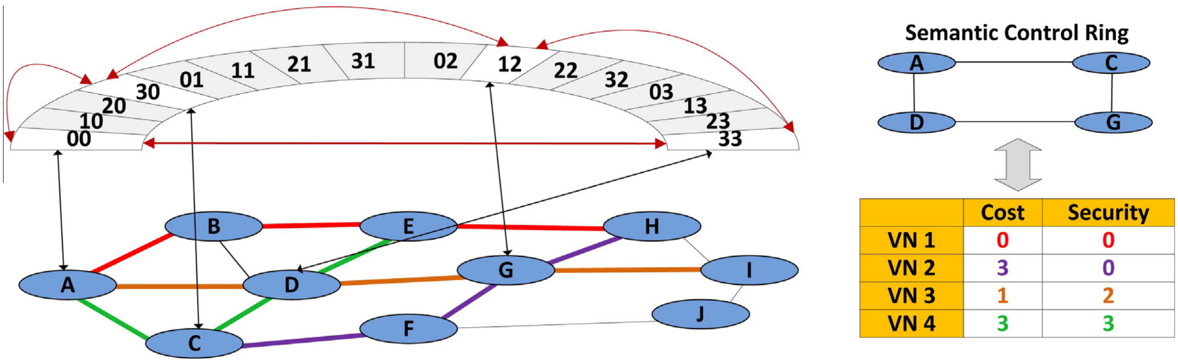

Fig. 3. DHT-based semantic-aware control ring.

- The key that identifies each VN broker in the control ring semantically embeds the context features, C, and levels of these features, L, that characterize its VN. In Fig. 3: (*i*) $C = 2 \rightarrow$ cost and security, and *(ii)* $L = 4 \rightarrow 0\ldots3$, where 3 is a very high level of context.
- The VN brokers are inter-connected in a consecutive context-aware order (see Fig. 3), enabling a "semantic-driven" mechanism to quickly find a point of attachment of a fitting VN for a user.
- In order to reduce the number of communications to find a fitting VN for a user, each WMN node acquires control knowledge of the WMN, being able to: (*i*) select and extend a fitting VN for a user that is available in its physical neighborhood, or (*ii*) trigger the global VN discovery process in a zone of the control structure that is semantically closer from a possible fitting VN for the user.
- The broker of a VN needs to be updated, since VNs are dynamically created, reconfigured, or terminated to adapt to context changing.

In this work, the WMN controller randomly selects a virtual node of each VN to be its broker, which has to be updated in case of VN topology adaptation due to context changing. Moreover, Chord DHT [31] is used to establish the links and shortcuts among the ring nodes.

3.2.2. Functionalities

After discussing the major entities of our architecture, the following paragraphs present the major functionalities that we aim to perform within the proposed semantic-aware distributed control framework.

Broadcasted control information. To efficiently distribute the control knowledge in the WMN, each WMN node will periodically broadcast the information about its available virtual nodes (and VN brokers) to its physical neighborhood. Then, such information can be disseminated in the WMN along a pre-defined maximum number of hops, and used to autonomously perform distributed control functions.

VN broker selection algorithm. Based on the acquired control information, a WMN node will be able to notify its local virtual nodes about the existence of other virtual nodes (or VN brokers) in the physical neighborhood. Such information will then be used in the VN broker selection algorithm performed within each VN. In the one hand, the VN broker needs to be closely located (in average) to every VN node to efficiently perform control functions within the VN. On the other hand, and to avoid topological mismatching problems between the physical WMN infrastructure and the proposed semantic-aware control ring, the VN broker needs to be closely located (in average) to the successor and predecessor VN brokers of the control ring, reducing the number of hops of the control communications performed within the ring.

Local VN discovery and extension. To endow the distributed control framework with intelligence to perform resource-aware and topology-aware local VN adaptations, the control messages used to disseminate the control knowledge stored by a specific WMN node will also convey information on the available resources in the traversed path (between the WMN node and the neighbor WMN node that contains a specific virtual node). Then, when a user arrives at a WMN node, such node will perform the context-aware matching among user's demands (and thresholds for context adaptation) and the features of the VNs available in its physical neighborhood. If there are more than one fitting VN, such node will make use of resource- and location-aware information to perform an optimized selection and adaptation of the best fitting VN for the user.

Global VN discovery redirection. If the WMN node does not have the knowledge of any fitting VN for the user in its physical neighborhood, it will inspect the stored information about the available VN brokers in its neighborhood, and trigger the global discovery process in a zone of the control structure that is semantically closer from a possible fitting VN for the user.

Global VN discovery and extension. The control paths (ring links) between consecutive VN brokers (ring nodes) will be periodically refreshed to adapt to the VN brokers locations' update. The control messages used to perform such mechanism can also convey qualitative information on the VN that is represented by each VN broker (for instance, the overall perceived QoE of the VN users). Then, within the control ring, and after finding a fitting VN for the user at a specific ring node by means of context-aware matching functionalities, the ring node will inspect the information stored about its 1-hop ring neighbors. Since the control ring is organized in a consecutive order of levels of context information, the fitting VN represented by a reached ring node may be semantically followed by a fitting VN for the user too. In such situations, the distributed control framework will take advantage of the stored information about the consecutive semantic VN, allowing an optimized mechanism to select and extend a specific VN up to the user's location, at the cost of one more control ring communication.

VN creation and termination. If the VN discovery process performed within the control ring does not give a desired result, several situations can occur: (*i*) the user will not be allowed to access the network, (*ii*) the user will connect to a default VN, or (*iii*) a exactly fitting VN can be created. The decision will be dictated by the policies defined and by the number of users with similar context. Due to the constant mobility and arrival of mesh clients at the WMN, and the frequent changing of their context requirements, the nodes of each VN (or even the whole VN) may be released. Both VN creation and termination mechanisms may trigger the update of the distributed control framework.

3.2.3. Control scenarios and signaling

Algorithm 1 describes the proposed control scenarios to associate a user that is arriving at a specific WMN node to the exactly fitting VN. These scenarios are then analytically evaluated in this paper, giving us an insight of the impact of different scopes of decision (local or global) for VN adaptation.

Scenario 1

When a user (u) arrives at a WMN node (R_u), and after the establishment of the user's link-layer (L2) connection, u sends its context requirements to R_u. In the following, R_u quantifies the user's requirements in context levels, and matches them against the local available VNs. If a fitting VN for u is already available in R_u (e.g., V_1), R_u notifies the WMN controller that a new user will be associated to V_1. The WMN controller stores the context of u, and performs the resource management algorithm to update the assigned wireless resources to the virtual node of V_1 that is available in R_u. Then, the WMN controller notifies R_u, which updates the assigned wireless resources to the user's associated V_1 node. Finally, u is notified about the establishment of its connection to V_1.

The interaction among the control entities of the architecture in this specific scenario is depicted in Fig. 4.

Scenario 2

If R_u detects that there is no available fitting VN for u after the matching among the context requirements of u and the features of the local available VNs, it triggers a local VN discovery mechanism in its 1-hop physical neighborhood. Such local mechanism can provide a fitting VN for u (e.g., V_2). In this case, and after receiving the replies from its neighborhood, R_u notifies the WMN controller that V_2, which is available in one of its physical neighbors, needs to be extended. After such notification, the WMN

Algorithm 1. User Connectivity Establishment

```
u: User accessing the WMN
R_u: Attached Mesh Router of u
V_u: Fitting VN for u
PoA_{V_u}: Point of Attachment of V_u
P_{R_u}: Physical Neighborhood of R_u
p: Selected Physical Neighbor of R_u
-----------------------------------------------------------------
/*Initialization Phase                                          */
L2 Association between u and R_u; u sends Context Requirements to R_u;
/*VN Discovery & Extension Phases                               */
/*Initialization Phase
if (PoA_{V_u} ⊂ R_u) then                     /* Scenario 1     */
    break;
else
    R_u asks P_{R_u} to know their current VNs; P_{R_u} answer R_u;
    if (PoA_{V_u} ⊂ P_{R_u}) then             /* Scenario 2     */
        Create a virtual link between R_u and p;
    else                                      /* Scenario 3     /*
        Global Discovery Mechanism to find PoA_{V_u};
        Create a virtual link between R_u and a node of V_u;
    end
end
/* Conclusion Phase                                             */
/*Initialization Phase
Notify u;
```

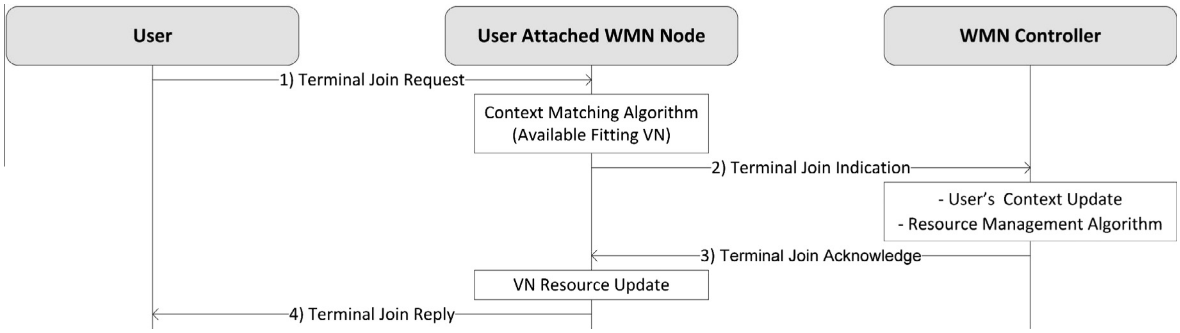

Fig. 4. Scenario 1: overall process description.

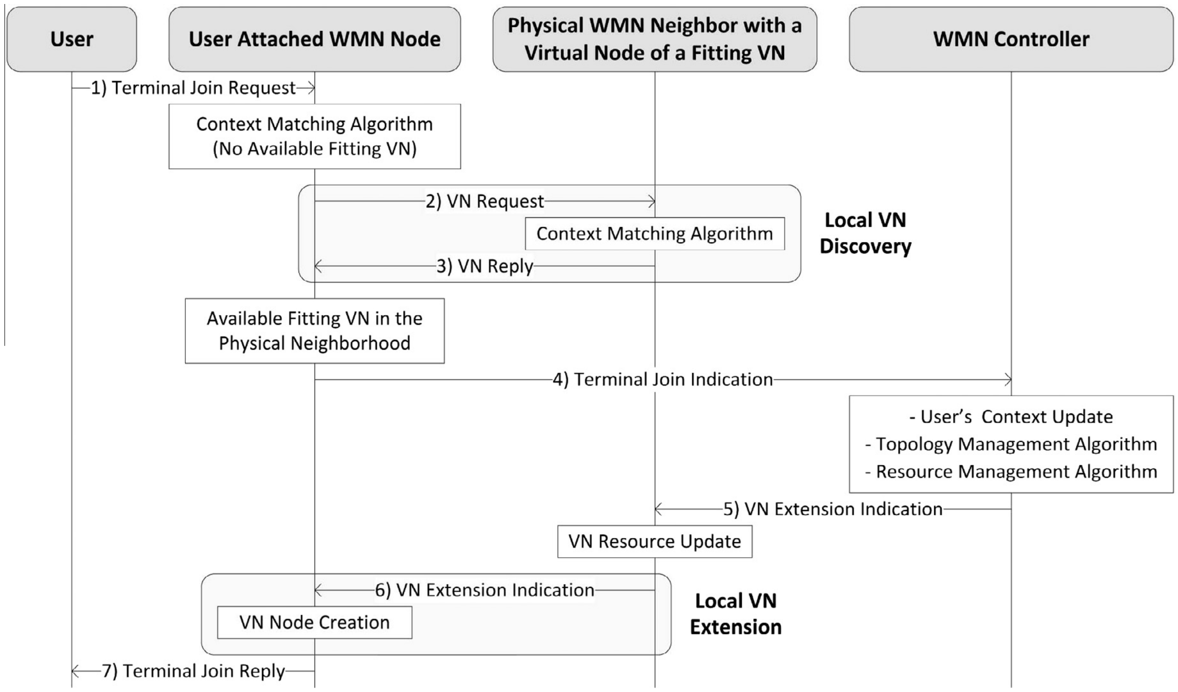

Fig. 5. Scenario 2: overall process description.

controller stores the context of u, and runs the topology and resource management algorithms to update the topology of V_2, and the assigned resources to the nodes that will be part of the new V_2 link. In the following, the WMN controller notifies the physical neighbor of R_u to update the assigned wireless resources to its V_2 node, which then notifies R_u to create a new V_2 node. Finally, u is notified about the establishment of its connection to V_2. This scenario is depicted in Fig. 5.

Scenario 3

The last proposed scenario for user's association is depicted in Fig. 6.

After checking its physical neighborhood, R_u may learn that there is no available fitting VN for u in the near WMN nodes. In this case, R_u triggers a global discovery mechanism in the WMN to find a point of attachment of a fitting VN for u, using the proposed semantic-aware distributed control ring. For instance, remembering Fig. 3, and assuming that u is attached in node F and the exactly fitting VN for u is VN_1, scenario 3 will take place since any virtual node of VN_1 belongs to node F or to any of its physical neighbors.

To start the global discovery process, R_u checks its stored information about the VN brokers that are located on it or on its 1-hop physical neighborhood, in order to redirect the discovery process to the zone of the control structure that is semantically closer to the context requirements of u. If R_u has no information about any VN broker, it starts a hop-by-hop discovery process until finding a node that belongs to the control ring; such hop-by-hop discovery process will usually take a small number of hops.

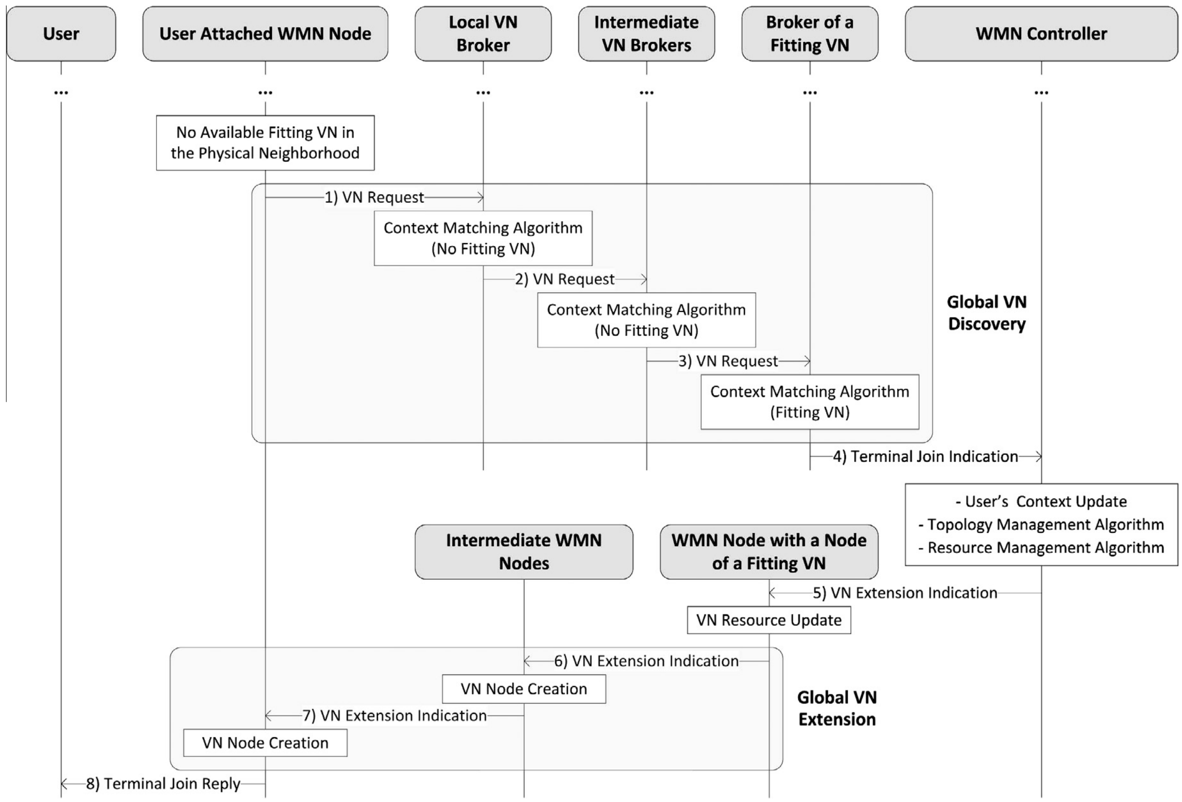

Fig. 6. Scenario 3: overall process description.

Then, and within the control ring, the global distributed discovery process continues through the links and shortcuts that are established according to the Chord overlay [31], until finding the broker of the fitting VN for u (e.g., V_3).

After reaching the broker of V_3, such VN needs to be adapted, through its extension up to R_u. First, the broker of V_3 notifies the WMN controller about the required extension. Then, the WMN controller stores the context of u, and runs the topology and resource management algorithms to update the topology of V_3, and the assigned resources to the nodes that will be part of the new V_3 links. As previously referred, the extension of V_3 is performed through the available shortest path, between R_u and the nearest V_3 node from the user's location, with the highest level of available wireless resources.

In the following, the WMN controller notifies the selected WMN nodes that will be part of the path for the extension of V_3.

Finally, u is notified about the establishment of its connection to V_3.

4. Analytical modeling

In this section, we provide an analytical model to evaluate several performance aspects of the multi-VN approach, addressing the previously presented control scenarios for user association in a fitting VN.

First, we define a closed-form expression for the end-to-end (E2E) delay of a data communication over a generic WMN. Then, we adapt this expression to evaluate the delays of the: (*i*) context-based data communications, taking into account not only their context requirements, but also the proper division of the total available bandwidth of mesh links among the context-based VNs that provide support for such communications, and (*ii*) local and global discovery and extension mechanisms to reconfigure fitting VNs for users.

4.1. E2E delay of a context-based data communication

In this sub-section, we extend the analytical model used in [26] to derive the mean E2E delay of a context-based data communication that occurs in a specific VN. In [26], a WMN is represented as an open queuing network through the adaptation of the diffusion approximation method [27]. Such an approach is commonly used to solve an open $G/G/1$ queuing network where all nodes in the network are single servers with infinite buffers and First-Come First-Serve service strategy (please see more details in [27]).

4.1.1. Model preliminaries and definitions

Considering a WMN as a square unit area divided into non-overlapping square zones of area a each, we have $N = 1/a$ zones in the WMN, each one covered by a mesh router (please see Fig. 7). Each router has an utilization

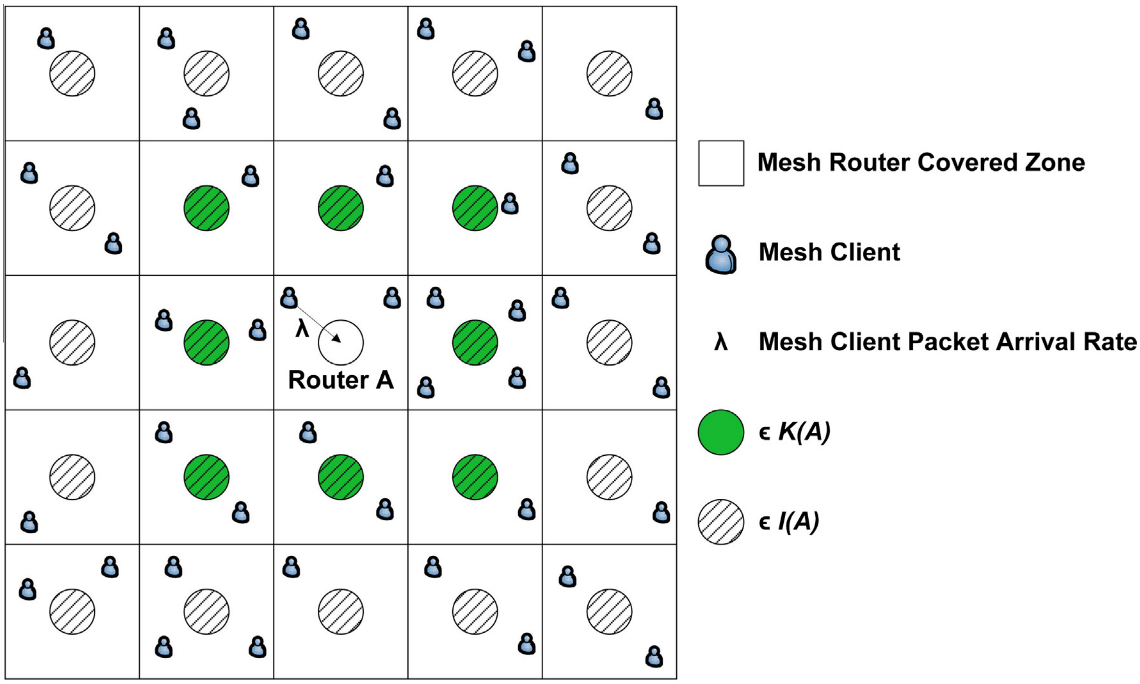

Fig. 7. Network model with squared zones of area a.

factor of ρ_i, and is equipped with an interface with capacity W_T. The set of neighbors of router i, $K(i)$, is considered equal to K in our study.

There are c mesh clients accessing the WMN, which are uniformly and independently distributed among all mesh routers.

The WMN data packet arrival rate is a renewal process with a mean packet inter-arrival time of $1/\lambda_e$, and Squared Coefficient of Variance (SCV) of C_A^2. Each client may be source or destination of data packets with size S, and generated according to a renewal process with rate λ. Hence, $\lambda_e = c\lambda$.

We denote $p_{0i} = a$ as the probability that a data packet enters the WMN through the zone covered by router i. As soon as a packet is generated by a client, it is forwarded to the router within its zone, which will relay it over the backbone to the zone of the target client. Denoting p as the probability that a packet received by a router is destined to a client within its zone, the probability to forward a packet to router i after completing its service at router j, p_{ij}, is defined by:

$$p_{ij} = \begin{cases} \frac{1-p}{K} & j \in K(i), \\ 0 & \text{otherwise}. \end{cases} \tag{1}$$

The visit radio of router i, e_i, is defined by the average number of times that a data packet visits the router. Thus, and considering from symmetry $e_i = e_j$, we have:

$$e_i = p_{0i} + \sum_{j=1}^{N} p_{ji} e_j = a + \sum_{j \in K(i)} \frac{1-p}{K} e_j = \frac{a}{p}. \tag{2}$$

The data packet arrival rate at router i is now defined by:

$$\lambda_i = \lambda_e e_i = c\lambda \frac{a}{p}. \tag{3}$$

Denoting s as the number of routers that forward a data packet before it reaches the destination, we know that $P[s = k] = (1-p)^{k-1}p$, for $k \geqslant 1$. Thus, the mean number of hops traversed by a data packet is defined by:

$$\bar{s} = E[s] = \frac{1}{p}. \tag{4}$$

A transmission made by a router i is successfully performed, if none of its one or two hop neighbors transmits concurrently on the same channel. The set of possible interfering neighbors of router i, $I(i)$, is considered equal to I in our study. In order to have an analytical tractable random access MAC model, similar to the IEEE 802.11 DCF MAC model, [26] defines expressions for: *(i)* number of active interfering neighbors of router i, H_i, which is a sub-set of $I(i)$, *(ii)* duration of the back-off timer of router i, T_i (exponentially distributed with mean $1/\xi$), and *(iii)* number of times that the back-off timer of router i freezes during a transmission epoch, M_i, assuming that it is frozen each time that an active interfering neighbor starts transmitting.

We can now define the service time of router i for the transmission of a data packet, X_i, as a sum of the following terms: *(i)* duration of its back-off timer (T_i), *(ii)* time for which the back-off timer remains frozen ($M_i \times (S/W_T)$), and *(iii)* packet transmission time (S/W_T).

4.1.2. Data communication delay model

Considering the previously defined variables, and according to the diffusion approximation method, we have:

- The SCV of the inter-arrival time of data packets at router i, C_{Ai}^2 (where $C_{B0}^2 = C_A^2$, and C_{Bi}^2 is the SCV of X_i):

$$C_{Ai}^2 = 1 + \sum_{j=0}^{N}(C_{Bj}^2 - 1)p_{ji}^2 e_j e_i^{-1}. \quad (5)$$

- The probability that the number of data packets at router i equals k, $\pi_i(k)$:

$$\pi_i(k) = \begin{cases} 1-\rho_i & k=0 \\ \rho_i(1-\hat{\rho}_i)\hat{\rho}_i^{k-1} & k>0 \end{cases}, \quad \text{with}$$
$$\hat{\rho}_i = \exp\left(-\frac{2(1-\rho_i)}{C_{Ai}^2\rho_i + C_{Bi}^2}\right). \quad (6)$$

- The mean number of data packets at router i, Z_i:

$$Z_i = \frac{\rho_i}{1-\hat{\rho}_i}. \quad (7)$$

So, and according to Little's law, the mean data packet delay at router i, $\overline{D_i}$, is defined by:

$$\overline{D_i} = \frac{Z_i}{\lambda_i}. \quad (8)$$

By symmetry, the mean data packet delay at all routers is the same, and therefore, the mean E2E delay of a WMN data communication, D_T, is defined by:

$$D_T = \bar{s}\overline{D_i} = \frac{\rho_i}{ca\lambda(1-\widehat{\rho_i})}. \quad (9)$$

4.1.3. Context-based characterization

Since our main goal is the provision of fitting VNs for users, which are characterized by different preferences and applications' requirements, it is required to adapt D_T (Eq. (9)) to derive the E2E delay of a context-based data communication that occurs in a specific context-based VN. Therefore, we need to describe $\overline{D_i}$ (Eq. (8)) according to the features of distinct context-based communications (S,λ,p), and the assigned bandwidth to the VNs that provide support for such communications (W).

The typical values for the size, S, and arrival rate, λ, of the data packets of common applications are used to differentiate the context purpose of such applications. For instance, VoD or IPTV are characterized by a high size and arrival rate of their packets, whereas VoIP is characterized by small size packets.

Concerning the context-based characterization of the probability of locality of data traffic, p, we know that, for instance, delay-sensitive communications are characterized by a high value of p, since it is required to have fast communications with a small number of hops ($\bar{s} = 1/p$). Therefore, and considering three types of delay constraints, we have:

$$p = \begin{cases} p_0 & \text{for small delay-sensitive VNs} \\ p_1 & \text{for medium delay-sensitive VNs} \\ p_2 & \text{for high delay-sensitive VNs} \end{cases}$$
$$0 < p_2 < p_1 < p_0 < 1. \quad (10)$$

The total bandwidth of mesh links, W_T, has to be distributed among the available networks of our architecture (context-based VNs, and the physical WMN infrastructure, where take place the control communications performed in the scope of the proposed control scenarios for user connectivity establishment) according to the throughput requirements of the communications that occur over them. Having in mind that the control communications need a small amount of the available WMN resources, and considering three levels of throughput requirements of data communications, the assigned bandwidth to the links of the available networks, W, is defined by:

$$W = \begin{cases} \alpha_c W_T & \text{for physical WMN infrastructure} \\ \alpha_0 W_T & \text{for small throughput-aware VNs} \\ \alpha_1 W_T & \text{for medium throughput-aware VNs} \\ \alpha_2 W_T & \text{for high throughput-aware VNs} \end{cases}$$
$$0 < \alpha_c < \alpha_0 < \alpha_1 < \alpha_2 < 1. \quad (11)$$

For simplicity, the presented analytical model assumes a single-radio WMN, that is, each WMN node has only a single wireless interface with a single server ($G/G/1$ model). However, and to deal with a multi-radio WMN, where WMN nodes may have multiple wireless interfaces (say M), and each one may have its own server ($G/G/M$ model), our analytical model can be extended. In such case, each WMN node can work in parallel in more than one wireless interface, and each interface can have assigned a different wireless channel. In fact, it would be very interesting to dynamically assign a wireless channel to a specific context-based VN. This would require the integration in the model of a scheme to dynamically control the assigned wireless interface (and the respective working channel) to a specific VN at a specific WMN node.

4.2. Delay of VN discovery and extension control processes

In this sub-section, we adapt the analytical model to derive closed-form expressions for the delays of the proposed local and global discovery and extension mechanisms to reconfigure fitting VNs for users.

4.2.1. Model preliminaries and definitions

Since clients and VN nodes are uniformly distributed in the WMN, the WMN topology influences the mean number of hops traversed by a control packet ($\overline{s_c}$). Following the assumption of Kumar-Gupta model [32]:

$$\overline{s_c} = \sqrt{\frac{N}{log(N)}}. \quad (12)$$

In the one hand, such expression is directly applied to a global VN discovery mechanism, since it is performed between two VN brokers that are randomly located within the physical WMN infrastructure. On the other hand, in a global VN extension mechanism, N needs to be replaced by the mean number of virtual nodes per VN, n. From the relation achieved for $\bar{s}$ (Eq. (4)), the probability of locality of control traffic is defined by:

$$p_c = \frac{1}{\overline{s_c}}. \quad (13)$$

The mean control packet delay at router i, $\overline{Dc_i}$, can be translated from $\overline{D_i}$ (Eq. (8)). However, in a VN discovery process, $\overline{Dc_i}$ only depends on the control traffic that flows over the physical WMN infrastructure. In a VN extension mechanism, $\overline{Dc_i}$ is totally related with the data traffic that flows over such VN, since it is required to create new virtual nodes, or to update the assigned resources of the existent ones.

4.2.2. VN discovery

In a local VN discovery process, the user attached router sends a message to its physical neighbors, which then reply to the router. So, the delay involved in such control process is defined by:

$$T_{Local} = 2\overline{Dc_i}. \tag{14}$$

A global VN discovery mechanism is performed along the semantic-aware distributed control ring. Since mesh clients, VN nodes, and VN brokers are uniformly distributed in the WMN, the starting point of this control process is also uniformly distributed among all available nodes. The semantic control ring is structured based on Chord protocol [31]: it is already known from the literature that the mean number of messages required by a lookup procedure in Chord protocol is $log(\#Peers) = log(L^C)$. Therefore, the delay involved in this global distributed discovery process is defined by:

$$T_{Global} = log(L^C)\overline{s_c}\overline{Dc_i}. \tag{15}$$

4.2.3. VN extension

The VN extension process involves the communication between a point of attachment of the chosen VN for the user and its user attached router. So, the delay involved in this control process is defined by:

$$T_{Extension} = \overline{s_c}\overline{Dc_i}. \tag{16}$$

In case of adding a link between the user attached router and one of its physical neighbors, $\overline{s_c} = 1$.

5. Evaluation: results & discussion

In this section, we make use of the defined analytical model to evaluate two important aspects of the data and control planes of our multi-VN approach: (*i*) the characterization of the context-based data communications, the configuration of the VNs that provide support, and the suitability of network virtualization to enable distinct WMN context-based communications, and (*ii*) the delays of the proposed control scenarios to locally or globally discover and extend fitting VNs for users.

The analytical results are compared against a NS-2 simulation environment. The simulation setup is also used to evaluate other key aspects of the architecture: (*i*) overhead of the proposed scenarios for user connectivity establishment, (*ii*) influence of users' mobility in the VNs' reconfiguration process, and (*iii*) the behavior of the architecture and its mechanisms using the topology of an experimental WMN testbed.

Table 1 summarizes all variables of the analytical and simulation studies.

5.1. NS-2 simulation framework

The defined architecture is implemented in NS-2 [33]. We create an AODV-based [34] IEEE 802.11 WMN. The WMN is characterized by a grid topology with $N = 7 \times 7$ mesh nodes, in order to be consistent with the analytical model. Each mesh router has a transmission radius of 100 m (700 m $\times$ 700 m of simulated area), and each mobile node has a transmission radius of 40 m.

By means of configurable options, the simulated environment allows to change the number of mesh nodes, mobile nodes (users), available VNs, number (and levels) of context features, and flows per VN.

In each simulation, we simultaneously establish 3, 6, or 9 VNs per considered VN type (as will be described), with a total number of 9, 18, or 27 context-based VNs in the WMN. Each VN has $n = 5$ virtual nodes uniformly distributed in the WMN. The number of flows per VN range from 1 to 4, all characterized by the arrival rate (λ), packet size (S), and probability of locality (p) of the context-based communications supported by such VN. Each VN flow is generated by a VN user (mobile node), and the location of the target of the user's flow is totally related with the probability of locality of the traffic supported by such VN, as will be described.

A well-known NS-2 DHT (Bamboo-DHT [35]) was used to simulate the proposed global distributed discovery of

Table 1
Details and assumptions of our study.

	Physical WMN infrastructure	Context-based VNs		
		S_0	S_1	S_2
N	49 ($7 \times 7 \rightarrow 700$ m $\times$ 700 m of simulated area)			
V	3, 6, or 9 per virtual slice			
n	5 virtual nodes per VN			
c (flows)	1, 2, 3, or 4 per VN			
C	2 (throughput and delay applications' requirements)			
L	3 (small, medium, high)			
$W(W_T = 54$ Mb/s)	$0.05W_T$	$0.1W_T$	$0.35W_T$	$0.5W_T$
K/I	8/24		$n-1$	
S	–	64 Bytes	256 Bytes	512 Bytes
λ		11 Kb/s	45 Kb/s	180 Kb/s
$\bar{s} = 1/p$		3	4	4
S_c		48 Bytes		

available fitting VNs for users. A node of each VN is randomly chosen to be its broker in the DHT-based semantic control ring. We add new functionalities to the original Bamboo-DHT code to: (*i*) enable the automatic mapping of context information in specific Bamboo keys that identify VN brokers (as described in Fig. 3), (*ii*) update the broker of each VN and the respective ring connections in case of adaptation of VN topologies due to context changing, or mobility of VN users, and (*iii*) adapt the number of Chord-based shortcuts among Bamboo nodes, according to the number of WMN nodes, VNs, context features, and levels of such features.

5.1.1. Network virtualization emulation

Since NS-2 has no network virtualization scheme, it is emulated using an available NS-2 Multi-Interface module [36].

In our approach, each node has 4 "virtual" interfaces (one for control purposes, and the remaining for data), each one using a non-interfering and dedicated wireless channel. Moreover, the bandwidth of a typical physical wireless interface, W_T = 54 Mb/s, is distributed among these 4 "virtual" interfaces, according to proper context-based rules described below.

Therefore, we create 4 virtual slices in the WMN (S_c for control, and S_0, S_1, S_2 for data), through the emulation of two major concerns of network virtualization: (*i*) network isolation, and (*ii*) distribute the level of wireless resources' usage at each WMN node among the different types of communications that are performed through it.

5.1.2. Context quantification and mapping

In order to differentiate VNs' topologies and assigned resources, we consider $C = 2$ context parameters to characterize different context-based data communications: delay and throughput applications' requirements. Each context parameter can have $L = 3$ distinct levels: high, medium, or small.

Throughout this study, it is assumed three types of context-based communications with distinct packet sizes, S, arrival rates, λ, and probability of locality, p, each one occurring in a specific VN: (*i*) high delay-sensitive and small throughput-aware, (*ii*) small delay-sensitive and medium throughput-aware, and (*iii*) small delay-sensitive and high throughput-aware.

Throughput requirements are differentiated by using three types of constant bit rate traffic, with distinct packet sizes and arrival rates: (high) 512 Bytes and 180 Kb/s; (medium) 256 Bytes and 45 Kb/s; (small) 64 Bytes and 11 Kb/s. Since there are three different virtual slices for data communications, VNs are grouped among them according to distinct throughput requirements. In this sense, VNs supporting communications with small/medium/high throughput requirements are built in $S_0/S_1/S_2$, respectively. Therefore, we introduce tailored factors to distribute the total bandwidth of mesh links (see (11)): $S_0/S_1/S_2$ have assigned $W = 0.1/0.35/0.5 \times W_T$, respectively. The remaining bandwidth ($W = 0.05 \times W_T$) is assigned to the physical WMN infrastructure for control purposes, which is built in S_c.

Concerning the characterization of the context-based communications supported by each VN according to their delay constraints, and following (10), we randomly select the source and the target of each communication flow, ensuring that the hop distance among them ($\bar{s} = 1/p$) is lower in case of high delay-sensitive communications. Therefore, and since our simulated WMN has 7×7 nodes, VNs supporting applications with high/small delay constraints are characterized by flows with mean number of hops ($\bar{s}$) of 3/4, respectively.

5.2. Analytical model: assumptions

Since in our analytical model the WMN follows a squared topology, each physical router of the WMN infrastructure has $K = 8$ physical neighbors, and $I = 24$ possible interfering neighbors. For the context-based VNs, K and I are considered equal to $n - 1$. The size of the packets for the proposed control mechanisms is S_c = 48 Bytes.

In the simulated environment, VNs are grouped in three different virtual slices for data communications. However, the assigned bandwidth to each virtual slice, W (see (11)), has to be distributed among the VNs running in such slice. We tried to avoid bottlenecks (saturated links) in our simulations through a correct distribution of VN links among the physical links. However, when the number of VNs per virtual slice starts to increase, the probability of these VNs making use of the same physical links also increases. Consequently, the assigned bandwidth to each VN decreases, being dependent from: (*i*) VNs per virtual slice (V), (*ii*) virtual links per VN ($l = n - 1$), and (*iii*) links of our grid topology ($L = \sqrt{N} \times (\sqrt{N} - 1) \times 2$). Based on a probabilistic approach, and to follow the simulation environment, the assigned bandwidth to each VN of a virtual slice in the analytical model can be approximated by:

$$W_{VN} = W \times \left(1 - \frac{l \times V}{L}\right). \tag{17}$$

The mesh clients (c) are equally distributed among all VNs. To follow the simulation model, the analytical model considers 1, 2, 3, or 4 clients per VN, each one generating a VN flow.

5.3. Data communications

In this sub-section, we assess the suitability of our approach to enable distinct context-based virtual data communications in WMNs.

Using both analytical and simulation models (20 simulations per value, with a confidence degree of 90%), we evaluate the E2E delay of the communication flows supported by the considered context-based VNs, varying the number of VNs per virtual slice (S_0, S_1, S_2), and flows (users) per VN (see Fig. 8). We also present simulation results with the same setup (same flows' types, sources, and targets), but now running in a normal AODV-based WMN without distributing data flows through VNs that run in isolated virtual slices. Following this way, we aim to evaluate the impact of network virtualization in our approach.

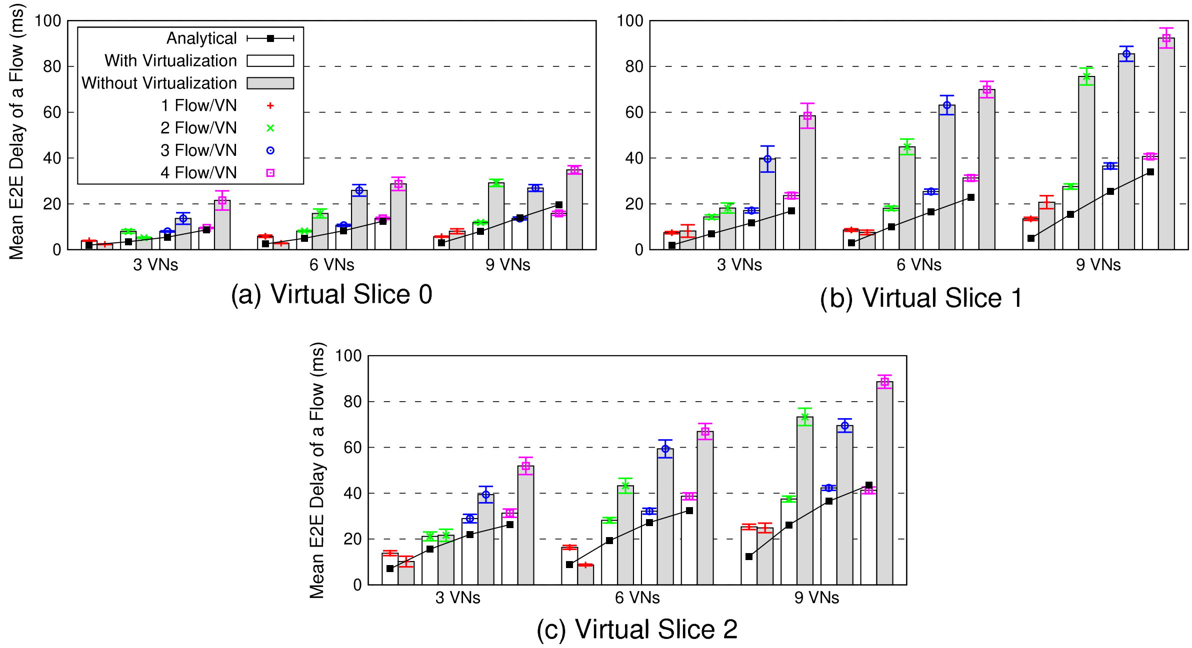

Fig. 8. Analytical & simulation models: E2E delay of VN flows: (a) high delay-sensitive and small throughput-aware; (b) small delay-sensitive and medium throughput-aware; (c) small delay-sensitive and high throughput-aware, with or without slicing the WMN, varying the number of VNs and flows (users) per VN.

From the obtained results, we show the benefits of providing distinct VNs, each one appropriate to different levels of delay and throughput requirements of users' applications. In the one hand, high delay-sensitive communications, which are supported by VNs running in S_0, present the lowest delays. On the other hand, S_1/S_2 present significant (but supportable) delays. The delays of high throughput-aware communications supported by S_2 are slightly higher, due to the larger size and arrival rate of their packets. These results also show the effectiveness on the distribution of the available WMN bandwidth among the virtual slices ($S_0/S_1/S_2$) for context-based data communications.

The increasing number of VNs (and flows per VN) introduces more traffic in the WMN, increasing the interference among different flows, which results in higher delays. Concerning this, we conclude that virtualization has a stronger impact when the WMN starts to be overloaded. Since virtualization enables a higher isolation among distinct communications, its benefits are more notorious when the WMN has a large amount of data traffic (more interference). On the other hand, if there is only one flow per VN (lower interference), the results are usually better when no virtualization scheme is applied. In this case, the static bandwidth sharing among VNs has no meaningfully advantages, since most of the assigned bandwidth at a particular VN will be wasted, and could be used in a more effective way.

Comparing the analytical and simulation results, we can clearly state that the results of the analytical approach and the NS-2 simulated environment are very similar. However, the delays of the probabilistic model are slightly lower, since it does not consider the limitations on buffer size and processing power of network nodes. On the other hand, the simulation model has to deal with more protocols of the NS-2 stack, which increases the delays.

Finally, we also perform some simulations with a 10×10 physical WMN. Even increasing in one unit the mean number of hops of the data communications (reducing the probability of locality of data traffic, p), the delays were always slightly lower than the ones of Fig. 8. For instance, it was observed a mean delay of 38 ms for the data communications performed in S_2, when there were 9 VNs per virtual slice, and four flows per VN. Such was expected, since due to the larger WMN size, the VN links have more probability to make use of non-overloaded physical links, which reduces the wireless interference and the network delay.

5.4. User connectivity establishment

In this sub-section, we evaluate the signaling delay and overhead of the proposed control scenarios for user connectivity establishment.

Assuming that the number of VNs and flows (mesh clients) per VN varies as in the previous sub-section, we randomly place mesh clients in the WMN to trigger the reconfiguration of available fitting VNs (update of a fitting available VN in the user attached router, or the local or global VN discovery and extension in the WMN). Then, we analyze the number of times each scenario occurs, and compute the mean of the reconfiguration delay (with a confidence degree of 90%). Fig. 9 presents the mean delays of the VN discovery and extension mechanisms involved in each scenario according to the analytical and simulation

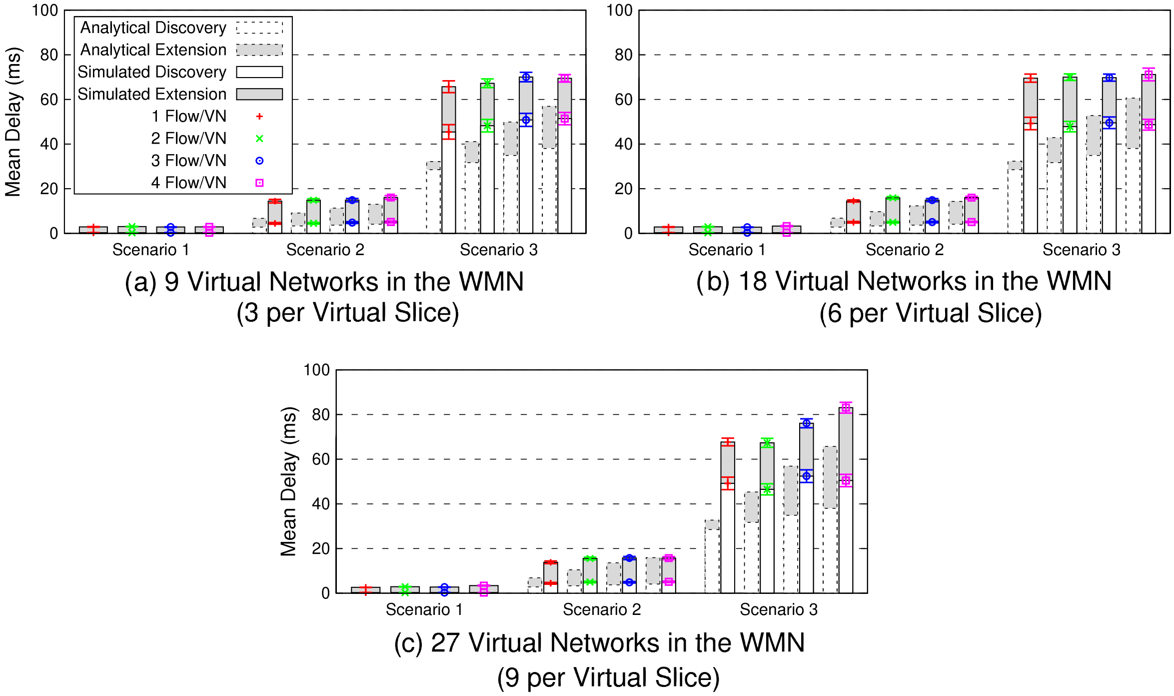

Fig. 9. Analytical & simulation models: VN discovery/extension delays for user connectivity establishment, varying the number of VNs: (a) 3 VNs per virtual slice; (b) 6 VNs per virtual slice; (c) 9 VNs per virtual slice and flows (users) per VN.

models, and Fig. 10 shows the overhead introduced by the local and global distributed VN discovery processes. Note that the overhead reflects the percentage of control traffic of the two discovery mechanisms that flows in the network.

Scenario 1 only depends on the processing power of routers, since a fitting VN for a user is already available in its attached router. Therefore, no discovery mechanisms are required, and extension delays are very small, only comprising the creation of a virtual link between the user and its attached router (this scenario was not addressed by the analytical model).

Scenario 2 involves a local VN discovery mechanism in the physical neighborhood of the router where the user is attached to, and the creation of a new virtual link to extend a fitting VN up to the router. Due to the small scope of these mechanisms, the delays are also small, even if the number of VNs and flows increases. Moreover, the control traffic introduced in the WMN by the local VN discovery process is clearly negligible, when compared to the data traffic that flows in the WMN.

It is observed that in scenario 3, the distributed discovery mechanism to find the point of attachment of one available fitting VN for the user has the largest delays. The broker of each VN is randomly placed in the WMN; therefore, the number of control communications in the semantic control ring to find a point of attachment of the fitting available VN for the user is variable, increasing the mean delay of this discovery process. However, these delays are not significant, and are independent from the data traffic that flows in the WMN, since the communications in the semantic control ring are performed within the physical WMN infrastructure (separation between data and control planes). The overhead introduced by this discovery process decreases as the number of VNs and data flows increase in the WMN. Such overhead is high when

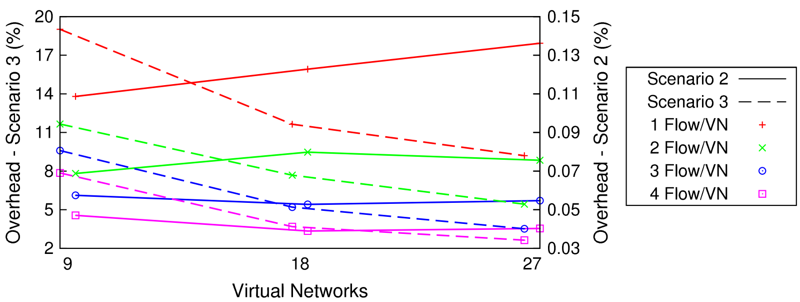

Fig. 10. Simulation model: overhead introduced by VN discovery mechanisms for user connectivity establishment, varying the number of VNs and flows (users) per VN.

there is few data traffic (one flow) per VN. Moreover, the overhead is higher than in the scenario 2, mostly because of the update and maintenance control traffic of the Bamboo-DHT. Concerning the VN extension process of scenario 3, it is more time consuming than the one of scenario 2, since it is often required to create virtual connections with more than one virtual link. Finally, this extension process lasts more time when the WMN starts to be overloaded (4 flows per VN).

Comparing the two models, we can again conclude that the analytical one is more conservative, presenting always lower delays than the ones obtained in NS-2. Both models present similar tendencies of delay increasing; however, the discrepancies between them are higher than in Fig. 8, since this model was originally deployed for data communications. In the one hand, concerning the global discovery of scenario 3, the analytical model does not take into account the maintenance control traffic of the Bamboo-DHT, which influences the NS-2 model. On the other hand, in NS-2, the VN extension mechanism is also influenced by other variables that are not addressed by the probabilistic model, such as the buffer size and processing power of WMN nodes to update or create virtual nodes.

When performing simulations with a 10×10 physical WMN, the delays and overhead of the distributed discovery performed in scenario 3 are slightly higher than the ones of Fig. 9 and 10, respectively. Due to the larger size of the network, the control communications performed within the ring may have a large number of hops, which increases the control delay and overhead. However, the distributed discovery delays are always less than 90 ms, even with increasing the number of VNs and flows in the network.

Due to the small signaling delay and overhead of our proposed discovery mechanisms, we can state that they have potential to be integrated in the multi-VN approach for context-based WMNs. Note that these delays are the small cost of providing flexible context-based WMNs, due to its dynamic control and management (this flexibility is crucial in a dynamic network environment). Finally, notice that the support of logical networks could be provided without virtualization; however, it would not be possible to develop on-demand networks with different context-aware network mechanisms employed (e.g., security, mobility) in the same infrastructure.

5.5. Influence of users' mobility

In this sub-section, we assess the influence of users' mobility in our approach.

We consider 18 VNs in the WMN (6 per virtual slice) with a variable number of flows generated by mesh clients that are randomly moving among 1-hop neighboring WMN routers at a vehicular velocity of 15 m/s. Then, we evaluate the VN reconfiguration delays (see Fig. 11a), the disruption time (see Fig. 11b) and packet loss (see Fig. 11c) of users' communications during the mobility process, considering the three proposed control scenarios.

The results of Fig. 11 show that the VN discovery delays of scenarios 1 and 2 are not significantly affected by the dynamics of the environment. On the other hand, the distributed discovery mechanism of scenario 3 has higher delays than in a static environment (see Fig. 9).

With respect to the VN extension processes, their delays slightly increase due to the mobility behavior of mesh clients. Several VN reconfigurations are simultaneously triggered in the WMN, probably occurring over the same physical nodes and links. Such fact increases the interference and delays involved in the creation of the new virtual connections.

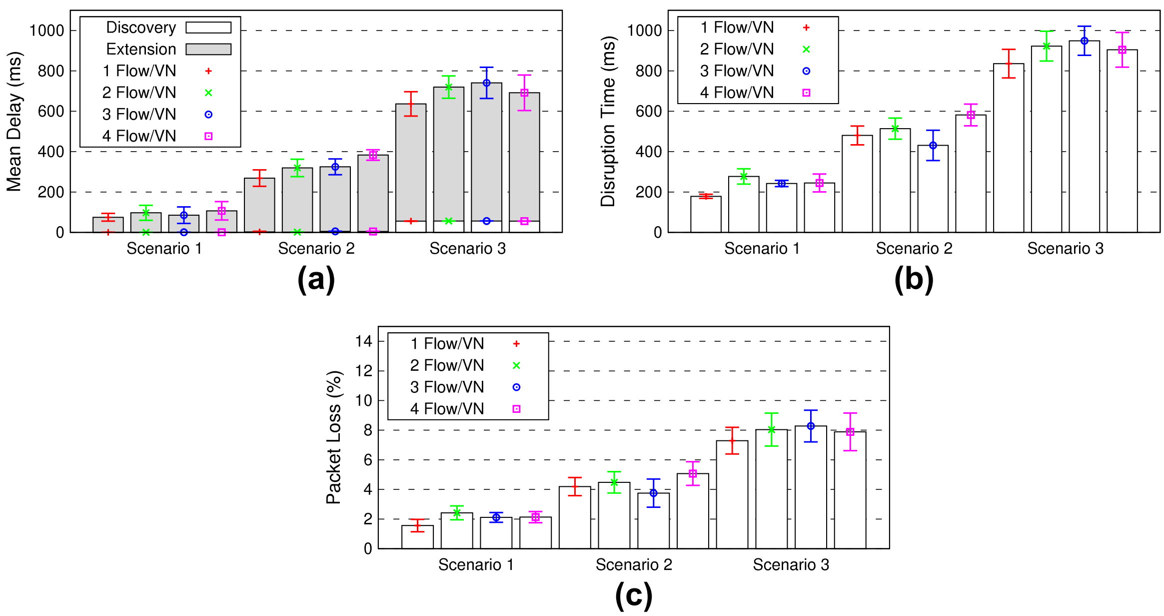

Fig. 11. Simulation model: (a) VN discovery/extension delays, (b) disruption time of each session measured at the target, (c) packet loss the each session measured at the target, considering mobility of users, 18 VNs in the WMN (6 per virtual slice), and varying the number of flows (users) per VN.

The session disruption times are always less than 1 s. Beyond the discovery and extension delays, the disruption time also includes the L2 handover delay and the communication path update. The session packet loss is always less that 10%, even if the WMN starts to be overloaded.

5.6. Influence of WMN topology

In this sub-section, and as a proof of concept of our modeling work, we evaluate the data plane of the proposed architecture using the node locations of the Funkfeuer Vienna WMN [8].

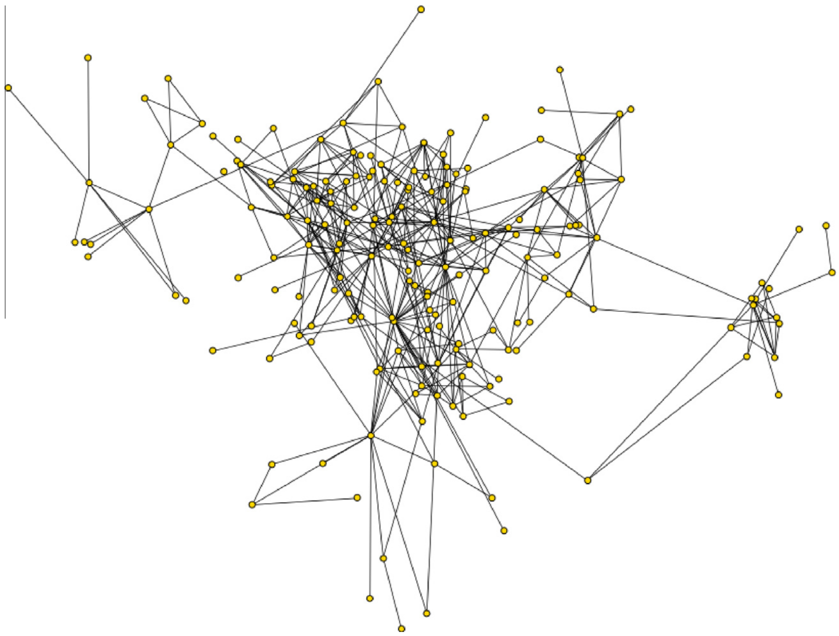

Fig. 12. Funkfeuer Vienna topology (network partitions are removed).

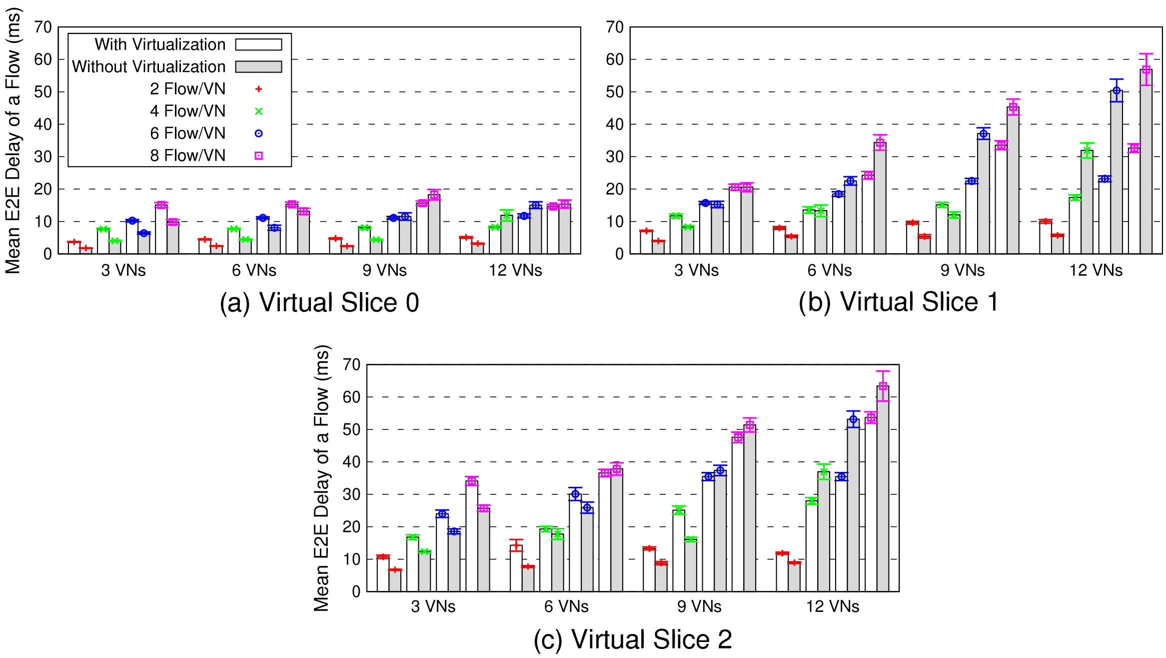

Fig. 13. Simulation model: E2E delay of VN flows (a) high delay-sensitive and small throughput-aware; (b) small delay-sensitive and medium throughput-aware; (c) small delay-sensitive and high throughput-aware, with or without slicing the WMN, varying the number of VNs and flows (users) per VN and using the topology of the Funkfeuer Vienna WMN.

5.6.1. Funkfeuer Vienna

Funkfeuer is a non-commercial, free-of-charge experimental WMN operated by (private) volunteering peers. It is deployed independently (not interconnected) in several Austrian areas like Vienna, Graz, Bad Ischl, and Weinviertel, and was started in 2005 in Vienna. The only requirement to join the WMN as a router is to obtain a static IP address from the Funkfeuer organizers. There are no restrictions regarding location and intended purpose (except special contents) of the router.

The Funkfeuer Vienna WMN consists of several wireless routers run by individuals which are interconnected with omni- and directional antennas using IEEE 802.11g and IEEE 802.11n standards in ad hoc mode. The employed router hardware is very diverse and ranges from stand-alone routers to Linux servers. Funkfeuer Vienna is mainly used for Internet access, which is provided through one centralized gateway. Besides Internet access, SSH tunneling, VoIP, and TV channel streaming services are offered.

Fig. 12 depicts the topology of the Funkfeuer Vienna WMN, which includes a total number of 211 routers and 701 links deployed within a city area of $414.89 km^2$. Routers without connection to the core of the network have been removed as they are partitioned, isolated networks.

5.6.2. Simulations

The coordinates of the real-world WMN Funkfeuer Vienna have been extracted and prepared for input to NS-2. The NS-2 simulation environment has been adapted to run with such a topology. The experimental setup allows us to derive the delays of distinct context-based virtual data communications in WMNs, when we are varying the number of VNs per virtual slice, and flows (users) per VN (see Fig. 13). We also present simulation results with the exactly same setup (same flows' types, sources, and targets) but now running in a AODV-based WMN (as described in sub-Section 5.3).

The delays shown in Fig. 13 are lower than the ones shown in Fig. 8. This is expected, since the number of WMN nodes in this simulation is higher than in the previous one, which increase the probability to setup VNs over non-overloaded physical WMN links, reducing the wireless interference and the network delay. From the obtained delay performance results, we can conclude that our approach can be used to contextualize and virtualize real WMN topologies.

6. Conclusion & future work

This paper proposed a probabilistic model to assess the efficiency of a context-aware multi-VN architecture that is able to provide personalized communications. The model is based on a diffusion approximation method that is used to define a closed-form expression for the delay of context-based data communications, and for the proposed local and global discovery and extension mechanisms to reconfigure fitting VNs for users. The results of both the model and simulations are similar, and show that the requirements of the data communications are indeed met, with a small impact of the network reconfiguration both in terms of delays and overhead. Moreover, the use of a real WMN topology did not lead to significant changes in the results. This reflects the fact that our approach can indeed be used in real running networks.

Although the probabilistic model is able to show the impact of the architecture in the data and the control processes, it is not built to tackle dynamics of the network, such as the study of the optimal reconfiguration approaches, which is a topic for further research. Moreover, we will enhance the proposed distributed control framework and all of its raised control mechanisms, endowing the WMN nodes with abilities to autonomously trigger or cooperate in the resource management and topology control functionalities. Furthermore, we will introduce semantic similarity metrics to perform the context matching among users' requirements and VNs' features.

References

[1] I. Akyildiz, X. Wang, A survey on wireless mesh networks, IEEE Communications Magazine 43 (9) (2005) 23–30.
[2] A.K. Dey, Providing Architectural Support for Building Context-Aware Applications, Ph.D. thesis, Atlanta, GA, USA, 2000.
[3] T. Anderson, L. Peterson, S. Shenker, J. Turner, Overcoming the Internet impasse through virtualization, Computer 38 (4) (2005) 34–41.
[4] R. Matos, S. Sargento, K. Hummel, A. Hess, K. Tutschku, H.d. Meer, Context-based wireless mesh networks: a case for network virtualization, Springer Telecommunication Systems Journal (2010). Accepted.
[5] R. Matos, S. Sargento, Data communications over context-based WMNs: delay performance evaluation, in: IEEE Workshop on Bandwidth on Demand and Federation Economics, Co-located with IEEE NOMS, 2010.
[6] R. Matos, S. Sargento, Context-based connectivity over multi-virtual wireless mesh networks: analytical study, in: IEEE ISCC, 2010.
[7] R. Matos, C. Marques, S. Sargento, Context-based connectivity and characterization of wireless mesh networks: simulation study, in: IEEE Workshop on Mobile Computing and Emerging Communication Networks, Co-located with IEEE Globecom, 2010.
[8] F. Vienna, <http://www.funkfeuer.at/>.
[9] A. Subramanian, M. Buddhikot, S. Miller, Interference aware routing in multi-radio wireless mesh networks, in: IEEE WiMesh, 2006.
[10] P. Hu, M. Portmann, R. Robinson, J. Indulska, Context-aware routing in wireless mesh networks, in: ACM CASEMANS, 2008.
[11] T.B. Thomas Staub, ATOM: adaptive transport over multipaths in wireless mesh networks, in: 2nd ERCIM Workshop on eMobility, 2008.
[12] J. Xie, X. Wang, A survey of mobility management in hybrid wireless mesh networks, IEEE Network 22 (6) (2008) 34–40.
[13] M. Oh, An adaptive routing algorithm for wireless mesh networks, in: International Conference on Advanced Communication Technology, 2008.
[14] J. Määttä, T. Bräysy, A novel approach to fair routing in wireless mesh networks, EURASIP Journal on Wireless Communications and Networking, 2009 (2009) 1–13, Article ID 379059.
[15] X. Zhao, Y. Lv, T.H. Yeap, B. Hou, A novel authentication and key agreement scheme for wireless mesh networks, in: International Conference on Networked Computing and Advanced Information Management, 2009.
[16] Group-GENI, Technical document on wireless virtualization (2006).
[17] C. Zhu, D. Wu, W. Cheng, Z. Yang, Efficient overlay multicast strategy for wireless mesh networks, in: IEEE VTC-Fall, 2008.
[18] D. Li et al., QoS-aware streaming in overlay multicast considering the selfishness in construction action, in: IEEE INFOCOM, 2007.
[19] A. Reaz, et al., Enhancing multi-hop wireless mesh networks with a ring overlay, in: IEEE SECON, 2008.
[20] G. Ding et al., Overlays on wireless mesh networks: implementation and cross-layer searching, in: IFIP/IEEE MMNS, 2006.
[21] MadWifi, <http://madwifi-project.org/>.
[22] D. Giustiniano, E. Goma, A. Lopez, P. Rodriguez, WiSwitcher: an efficient client for managing multiple APs, in: ACM PRESTO, 2009, pp. 43–48.

[23] T. Staub, R. Gantenbein, T. Braun, VirtualMesh: An Emulation Framework for Wireless Mesh Networks in OMNeT++, 2009.
[24] OMNET++, <http://www.omnetpp.org/>.
[25] J. Zhou, K. Mitchell, A delay based multipath optimal route analysis for multi-hop CSMA/CA wireless mesh networks, in: International Teletraffic Congress (ITC), 2007.
[26] N. Bisnik, A. Abouzeid, Delay and throughput in random access wireless mesh networks, in: IEEE ICC, 2006.
[27] G. Bolch et al., Queueing Networks and Markov Chains, 1998.
[28] Y. Chen, J. Chen, Y. Yang, Multi-hop delay performance in wireless mesh networks, Mobile Networks & Applications 13 (1–2) (2008) 160–168.
[29] K. Medepalli, F.A. Tobagi, Towards performance modeling of IEEE 802.11 based wireless networks: a unified framework and its applications, in: IEEE INFOCOM, 2006.
[30] Y. Gao et al., Determining the end-to-end throughput capacity in multi-hop networks: methodology and applications, in: SIGMETRICS, 2006.
[31] I. Stoica et al., Chord: a scalable peer-to-peer lookup protocol for internet applications, IEEE/ACM Networking 11 (1) (2003) 17–32.
[32] P. Gupta, P.R. Kumar, The capacity of wireless networks, 1999.
[33] The Network Simulator NS-2, <http://www.isi.edu/nsnam/ns/>.
[34] C. Perkins et al., Ad hoc on-demand distance vector (AODV) routing, in: RFC 3561, 2003.
[35] The Bamboo Distributed Hash Table, <http://bamboo-dht.org/>.
[36] NS-2 Multi-Interface Support, <http://personales.unican.es/aguerocr/>.

Ricardo Matos is a Ph.D. student at University of Aveiro. He concluded in July 2008 his Integrated M.Sc. in Electronics and Telecommunications Engineering from the Department of Electronics, Telecommunications. and Informatics of University of Aveiro. Since September 2008 he has joined Institute of Telecommunications, located in University of Aveiro Campus, as a researcher and Ph.D. student. He has been involved in several national projects, as well as in the FP6 European Project WEIRD, and FP7 European Project Euro-NF. His current research interests are related with future Internet architectures, virtualization techniques, wireless mesh networking, self-management, and context-awareness.

Carlos Marques is a researcher at Institute of Telecommunications. He concluded in July 2010 his Integrated M.Sc. in Electronics and Telecommunications Engineering from the Department of Electronics, Telecommunications, and Informatics of University of Aveiro. Since September 2010 he has joined Institute of Telecommunications, located in University of Aveiro Campus, as a researcher. He has been involved in some national projects, such as, Panorama and, and European ones, such as Euro-NF. His current research interests are related with simulation platforms, wireless mesh networks, and context-awareness.

Susana Sargento (Ph.D. in 2003 in Electrical Engineering) joined the Department of Computer Science of the University of Porto in September 2002, and is in the University of Aveiro and the Institute of Telecommunications since February 2004. She has been involved in several national and European projects, taking leaderships of several activities in the projects, such as the QoS and ad hoc networks integration activity in the FP6 IST-Daidalos Project. She is currently involved in several research projects, European Projects FP7 (4WARD, Euro-NF, C-Cast), national projects, and from CMU|Portugal (DRIVE-IN with Carnegie Melon University). Her main CMU|research interests are in the areas of Next Generation and Future Networks, more specifically QoS, mobility, self- and cognitive networks, and cloud concepts to networks.

Karin Anna Hummel received her Ph.D. in 2005 on the topic "Mobility-aware Distributed Computing in Shared Data Spaces" from the Vienna University of Technology, Austria. She is a scientific member of the Distributed Systems group, University of Vienna, leading several national and international projects and work-packages within projects in the area of mobile and mobility-assisted computing. Before joining the University of Vienna, Karin Anna Hummel was working as software engineer and project leader with Siemens Austria in the area of network management for five years. Her research interests focus on mobility modeling and mobility-assisted, self-organized networking. Karin Anna Hummel is currently leading one research work-package on mobile grids in the Austrian Grid initiative, one Euro-NF one-year project on overlays for wireless mesh networking, and one French-Austrian bilateral project on generating resource-efficient mobility-predictors. Moreover, she is key researcher in the project "Ambient Assisted Shared Living", member of Euro-NF (EU FP7 Network of Excellence on the Network of the Future), and member and co-leader of working-group 2 in the COST-action IC0804 on "Energy Efficiency in Large Scale Distributed Systems". Karin Anna Hummel is author of several journal and conference publications and co-editor of two proceedings (IWSOS'08, IWSOS'09).

Harald Meyer is a Ph.D. student at the University of Vienna, Department of Distributed and Multimedia Systems. He holds a master's degree in computer science and business informatics. Before he joined the University of Vienna, he was a project member of VRVIs research studio, Austria. In his doctoral work, Harald Meyer focuses on efficient and robust data dissemination in mobile and wireless computing environments.