

How to Study Wireless Mesh Networks: A hybrid Testbed Approach

Alexander Zimmermann Mesut Güneş
Martin Wenig Ulrich Meis Jan Ritzerfeld

Department of Computer Science, Informatik 4
RWTH Aachen University, Ahornstr. 55, 52074 Aachen, Germany
{zimmermann, guenes, wenig, meis, ritzerfeld}@cs.rwth-aachen.de

Abstract—Simulation is the most famous way to study wireless an mobile networks since they offer a convenient combination of flexibility and controllability. However, their largest disadvantage is that the gained results are difficult to transfer into reality since not only the abstraction of the upper network layer are typically high, but also the environment of mobile and wireless networks is very complex. This is due to two reasons. First there are typically many simplifications in the models of the upper networking layers, and second the environment of mobile and wireless networks is in particular complicated and thus difficult to be considered in all details.

In this paper we introduce *UMIC-Mesh*, a hybrid testbed approach, that consists of real mesh nodes and a virtualization environment. On the one hand the virtualization allows the development and testing of software as if it was executed on real mesh routers, but in a more repeatable and controllable way. On the other hand the results and conclusions gained by a software evaluation in the testbed can be easily transferred into reality, since the testbed represents a high degree of realism.

I. MOTIVATION

In recent years, wireless networks became widely accepted as an alternative to wired networks for connecting end-user devices. Standardization and decreasing costs enabled their success in the mass-market and lately the wide-spread deployment in private households. However, such systems usually only bridge the distance between a base station and the end-user device wirelessly. Behind the base station, there is a more or less complex infrastructure based on traditional wired network technology. For example, the access points (AP) of a wireless local area network (WLAN) are generally interconnected by Ethernet cables.

Over the past years, an approach has received a great deal of attention that consistently adopts wireless network technology: *mobile ad-hoc networks (MANET)*. MANETs are spontaneously formed by wireless mobile nodes having no need for a pre-existing infrastructure like the wireless APs. Here, nodes communicate directly with other nodes within their wireless transmission range. In order to communicate with the remaining nodes, they use intermediate nodes as relays. In other words, every node acts as a host as well as a router and, therefore, contributes both to the network architecture and to the routing. MANETs are decentralized networks, there are no special nodes whose failure might tear down the network completely. Thus, it is possible to establish MANETs in situations where traditional wired infrastructure is damaged, or does not even

exist, and cannot be (re-)established in a timely manner. Typical application scenarios are the deployment of relief units in disaster areas or of combat units in battlefields.

Despite intensive research effort, MANETs have not become widely accepted by the mass market, yet. This is comprehensible insofar as the above and frequently mentioned application scenarios are very specific and mainly either military-driven or solely applicable in special cases. However, the masses demand versatile networks providing them *high bandwidth* and *access to the Internet*. [1]

There is a new class of networks fulfilling these requirements, the so-called *wireless mesh networks (WMN)* [2]. Even though MANETs and WMNs share the same idea, the latter do not have the aim to establish utterly self-sufficient, isolated networks without any pre-existing infrastructure. In contrast, WMNs can be integrated into wired networks and can easily extend them at low cost without losing the mobility or flexibility of MANETs.

A. Contribution of this Paper

The contribution of this paper is twofold. The first aspect of this paper is to introduce a novel approach to realize a WMN testbed. Starting from a comparison of the characteristics that the different environments for studying wireless and mobile networks have, we outline the drawbacks of a typical testbed for WMNs. On that basis we present *UMIC-Mesh*, our hybrid testbed approach. The focus is on the discussion of the new *system* and *network architecture* of the testbed rather than to present an extensive performance evaluation. However, for research purposes it is not enough to set up a WMN testbed only consisting of hardware and the software, i.e., operating system and router functionality. The testbed architecture should also ease the execution of research studies. We achieve this easement through a combination of real mesh nodes and a virtualization environment. On the one hand the virtualization allows the development and testing of software as if it was executed on real mesh routers, but in a more repeatable and controllable way. On the other hand the results and conclusions gained by a software evaluation in the testbed can be easily transferred into reality, since the testbed represents a high degree of realism.

The second aspect of this paper is to give detailed information about the testbed. Although there are some other WMN testbed projects, the reports rarely give an insight how to build and run

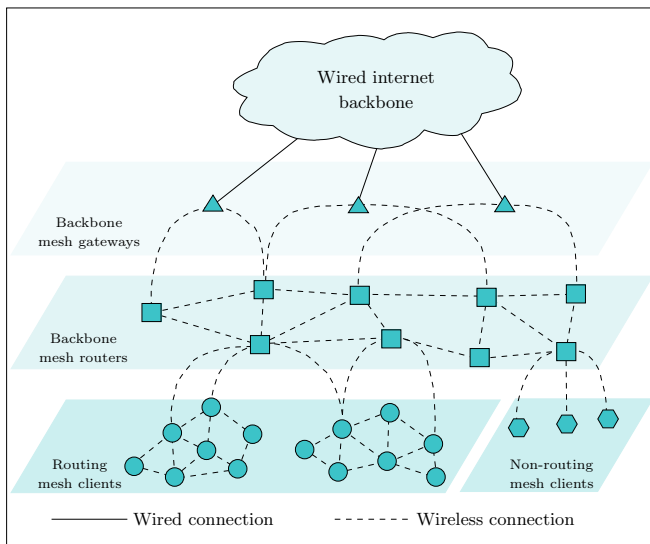


Fig. 1: Architecture of wireless mesh networks

a WMN testbed. This is valid for the hardware components as well as for the software components. However, this is important, since the setup of a testbed is very labor intensive.

B. Structure of the Paper

The remainder of the paper is organized as follows. Section II gives an overview of WMNs. In Section III we discuss environments used to study mobile and wireless networks, in particular WMNs. Our solution to investigate WMNs is presented in Section IV. Previous work, that is existing testbeds for WMNs, is surveyed in Section V. Finally, in Section VI we draw some conclusions and outline future work.

II. WIRELESS MESH NETWORKS

WMNs are an emerging technology. Unfortunately, there is no one and only one definition or architecture of WMNs and, thus, no common practice for designing such networks. On the one hand, there exist WMN definitions regarding MANETs as the simplest variant of WMNs [2]. On the other hand, WMNs are considered to be special MANETs [3]. In order to face the ambiguity of the term “wireless mesh network” a common definition of WMNs is presented here.

A. System and Network Architecture

Figure 1 depicts a hierarchical and layered architecture that integrates various approaches and, thus, helps to identify the main parts of a WMN. This view is more general than that usually presented insofar as that other approaches often leave out some layers, e.g., they consider only clients without routing functionality [1].

On the top level of Figure 1, there are the *backbone mesh gateways* connected to the Internet by wire, indicated by solid lines. They provide wireless Internet access (dashed lines) to the second level entities, the so-called *backbone mesh routers*. These wireless routers form the core by building the wireless, meshed backbone of the WMN. On the lowest level, there are

the mobile user devices, the *mesh clients*. As Figure 1 shows, these clients are subdivided into two groups. On the left hand side, there are *routing mesh clients* that also communicate among each other in a multi-hop fashion. They form a MANET with gateways that are not directly connected to the Internet but to the backbone mesh gateways. On the right hand side, there are *non-routing mesh clients*, which connect to mesh routers in the same way as conventional IEEE 802.11 clients associate to wireless APs.

The architecture outlined above needs further discussion. First of all, the mesh gateways are specific mesh routers that have a wired, high-speed connection to the Internet. These wired connections are considered not to be part of the WMN. Thus, the WMN itself is fully wireless.

The mesh routers and gateways are installed at certain fixed positions. They establish a permanent infrastructure. However, new routers and gateways can easily be added, since the connection is wireless. Thus, the infrastructure and, therefore, the network topology is not completely static but has a low dynamic character. Mesh routers and mesh gateways together establish a wireless multi-hop network that serves as a backbone. Traffic that cannot be delivered within local MANETs formed by mesh clients directly is routed hop-by-hop through the backbone. Furthermore, it routes external traffic from a mesh client to a mesh gateway that can forward it to the Internet and vice versa. This way of communication is a major difference to conventional wireless APs, which provide only gateway or bridge functionality. In contrast, a mesh router has additional multi-hop routing capabilities. Furthermore, the hierarchy achieved by the distinction between clients and routers promotes the utilization of multiple radios, separating the traffic in the backbone from the one of the clients. Routing and configuration tasks are assigned to mesh routers in order to unburden mesh clients that are probably power-constrained because of their inherent mobility.

Due to the mobility of the mesh clients the WMN has got a spontaneous and dynamic character. Mesh clients can leave the WMN at any time and new clients that want to join the WMN might arrive at any time. Figure 1 introduces two groups of clients. The non-routing mesh clients are confined to direct communication with mesh routers only. They do not participate in the routing process of the WMN but use the mesh routers similar to conventional wireless clients communicating with their AP. The routing mesh clients are able to connect not only to mesh routers but also to other routing mesh clients. Since they participate in routing, these clients build up a sub-network on their own, which can also be considered as a MANET.

B. Network Characteristics

Our definition of the architecture of a WMN leads to several characteristics. These are quite general and many of them also hold for other views of WMNs.

Wireless: The most obvious property is the wireless nature of WMNs. Thus, WMNs must cope with the challenges that arise from wireless communication. On the one hand, they need to take into account the limited transmission range and

the potentially high loss rates due to packet collision and fading of the wireless channel during the transmission. On the other hand, they have to deal with the mobility of nodes.

Multi-hop: WMNs use multi-hop routing to overcome the challenges mentioned above. Conventional wireless networks extend their network coverage by higher transmission power or additional APs that have to be interconnected by wire. In contrast, a node of a WMN forwards traffic wirelessly on behalf of other ones which are not within their direct transmission range.

Redundancy: The wireless backbone of a WMN forms a meshed network. It provides redundant links between mesh routers, mesh gateways, and mesh clients. Thus, failure of one link or node will not necessarily lead to failure of large parts of the network. Trying to adopt this approach in conventional wired networks might be expensive, time-consuming, or even impossible. This is because of the large amount of cabling required for such a meshed network. Furthermore, clean in-wall installations could be impossible depending on the environment.

Mobility: Since both mesh routers and mesh gateways have low mobility, the backbone can support client mobility in a predictable and reliable fashion. Mobile clients that leave the communication range of one mesh router can easily connect to the next one coming into their communication range. The dynamic multi-hop routing will ensure that the traffic is still correctly forwarded to its destination.

Dynamics: All nodes have to establish the network spontaneously (*self-organizing*) and to maintain their connectivity continuously (*self-healing*). Leaving or newly joining nodes cause topology changes the network must adapt. Nodes must reorganize their routes, invalidate paths that are not available anymore and include new paths that have become available. Additionally, the WMN should pass configuration information to new nodes in order to reduce or remove the need for user intervention (*self-configuring*).

The characteristics mentioned above are essential, but MANETs share similar qualities. The following characteristics are specific to WMNs, they clearly distinguish WMNs from MANETs.

Infrastructure: Unlike MANETs, WMNs have got a pre-existing, hierarchical architecture. Mesh gateways and routers forming the backbone infrastructure are nearly static and therefore less limited with regard to power consumption and computing power. They can be equipped with multiple radios, and they can take over routing and configuration tasks. The hierarchical static infrastructure simplifies such functionalities since the backbone is more reliable than mobile nodes are.

Integration: The non-routing mesh clients can join a WMN without the need for sophisticated routing support. Thus, light-weight and power-constrained clients can be attached to a WMN. They need not be an active part, in contrast to

MANETs that require all nodes to be cooperative. Supporting such passive clients enables the integration of devices, or whole networks, into the wireless backbone.

The latter characteristics influence all the ones mentioned before. The introduced hierarchy adds several improvement opportunities in comparison to MANETs.

C. Network Classification

In the previous sections, a technical definition of WMNs and a description of its consequent characteristics there have been given. This section presents a classification that takes into account the *functional* differences. The main idea here is to bring into focus the aspect of service provision and management of WMNs. In the following, three types of management are derived from the operation and control of deployment, maintenance, and administration processes found in existing WMNs.

1) *Fully Managed:* In such networks, all nodes providing any service are managed by a central administration. As a consequence, there are no routing mesh clients, since they would indeed offer services, namely routing. Therefore, the most apparent difference between fully managed WMNs and the general WMN depicted in Figure 1 is the lack of these routing mesh clients. Thus, a clear separation is made between the infrastructure itself, formed by backbone mesh routers and gateways only, and its users, the non-routing mesh clients. As a rule, a single company deploys, maintains, and controls the infrastructure of the WMN.

2) *Semi-managed:* In contrast to fully managed WMNs, in a semi-managed WMN just a fraction of the nodes providing any service is controlled by a single organization. This means that there is a great number of such nodes not being under control of this organization. These nodes may not only be static but also mobile and might join or leave the network at any time. Nonetheless, they are an integral part of the network, in contrast to clients in a fully managed WMN. That is, they perform tasks such as the participation in the routing process, the contribution to the autoconfiguration support, and the provision of service discovery.

3) *Unmanaged:* In an unmanaged WMN, there is nothing like a central administration that manages stable and service providing resources. Instead, such networks are formed by nodes that are under control of individual enterprises or persons. Similar to MANETs, there is no dedicated, pre-existing infrastructure required. An unmanaged WMN adopts the available infrastructure and, therefore, might persist only for a transitory period of time. However, nodes of a MANET are supposed to move more frequently achieving a higher mobility. In contrast, nodes in an unmanaged WMN move less often and, hence, they are rather static. An unmanaged WMN may emerge, e.g., in a conference room or on a university campus. Moreover, some nodes may act as gateways connecting the WMN to external networks like the Internet, e.g., via GSM, UMTS, or WLAN.

III. HOW TO STUDY WIRELESS MESH NETWORKS

After having designed a new network protocol there are typically several possibilities to evaluate and validate it. In general, approaches of evaluation and performance analysis of network protocols can be classified into five categories. These are *theoretical analysis*, *simulation*, evaluation through *emulation* or *virtualization*, and the direct measurement in a *real world testbed*. All these evaluation methods are very different in their degree of abstraction, relative to the real application. A mathematical analysis has the highest abstraction and, in descending order, is followed by simulation, emulation, virtualization, and finally reproduction in a real world testbed. The use of simplifying quantitative models leads thereby to a deviating behavior of the experimental setup. The more parameters remain unconsidered in such a model, the larger the inaccuracies will be that can occur in the evaluation.

A. Theoretical Analysis

Theoretical analysis uses mathematical models to evaluate network performance. Queueing theory is one of the most common mathematical tools in network performance studies. Unfortunately, theoretical analysis of WMNs is very difficult, since the mathematical constructs get very complex for realistic considerations; useful mathematical tools do not exist.

B. Simulations

A simulation environment offers a high degree of control and repeatable results to the researcher. This is especially useful when studying highly distributed networks like MANETs or WMNs. During the study of such a network, typically few parameters are varied and most remain fixed. This allows the study of the network in respect to the varied parameters. Simulation studies are therefore very flexible and the related costs are low, since it is possible to conduct complex experiments even with only one computer.

However, a simulation study has also its disadvantages. The simulation environment is typically an abstraction of the reality and therefore contains many simplifications. In the case of mobile and wireless networks, which have a very complicated and dynamic environment, the simulation environments are far from being *realistic*. This leads to results that do not fit with real-world measurements.

C. Emulation

Emulation is a hybrid study environment that consists of two parts: existing hardware and real network layers or parts, and a simulated environment. Which elements are real and which are simulated depends on the study goals and may differ considerably. However, with emulation it is possible to increase the quality of the study environment by making it more realistic.

An important advantage of emulation environments over simulation environments is the possibility of validation against real traffic. The advantage of emulation environments over real world experiments is the possibility of scaling to larger topologies by multiplexing simulated elements on physical resources, e.g., network interfaces [4].

D. Virtualization

In general, virtualization environments can be classified into three classes. The first class is the *system virtualization* with the virtual machine monitor (VMM) *inside* the host system. With this type the virtual machine (VM) simulates the complete hardware, allowing an unmodified operating system (OS) for a completely different CPU to be run. The second class of virtualization is also a type of system virtualization, but in contrast to the previous one, the VMM is *underneath* the host OS. Thus, the VMM runs directly on the hardware. In general this allows multiple OS to run, unmodified, at the same time. The third class of virtualization is the *operating system-level virtualization*. It virtualizes a physical server at the OS level, enabling multiple isolated and secure virtualized servers on a single physical server. The guest OS is the same OS as the host system, since the same OS kernel is used to implement the guest environments.

Beside the technical aspects the virtualization offers an adequate tool to evaluate communication protocols. With the aid of virtualization, it is possible to create several VMs on a single host system. Each VM can run a separate OS and hence represents an entire computer system. By coupling several VMs over the network, it is possible to create a whole virtual network of VMs. The most important advantage of virtualization is that the software development can be done on real machines with a real OS, and tested on the virtual network of VMs.

E. Real Testbeds

The best environment to study network protocols is to conduct experiments in a real testbed. Typically, this is done by prototype implementations. The results and conclusions can be easily transferred to reality, since not only the prototype, but also the testbed represents a high degree of realism.

However, in the case of distributed and mobile networks, it is very difficult to conduct experiments. The researcher has only limited control over the environment, since there are many influences from the study environment, e.g., interference with production networks. Experiments are typically difficult to repeat, and the experiment setups are restricted in size as well as in complexity. It is also very expensive to conduct experiments in the real world from the hardware point of view as well as from labor intensity. Last but not least, these kind of experiments are limited to existing technologies.

F. Summary

The characteristics of the discussed environments are depicted in the upper part of Table I. We focused on the following categories.

Applicability: Evaluates the degree of transferability of the results, and conclusions into the real world.

Repeatability: Rates how straightforward the repetition of a given experiment in that study environment is.

Controllability: Assesses the degree of control the researcher has over the study environment.

TABLE I: Overview of the characteristic of environments for wireless networks.

Characteristic	Environments				
	Theoretical Analysis	Simulation	Emulation	Virtualization	Real Testbeds
Applicability	poor	low	middle	high	high
Repeatability	–	high	low	low	poor
Controllability	high	high	middle	middle	poor
Maintainability	–	high	middle	middle	poor
Scenario creation	–	simple	middle	middle	compl.
Scalability	–	high	middle	middle	low
Duration	–	var.	real	real	real
Cost	–	low	middle	middle	high
Application	–	low	high	high	high
Transport	–	low	middle	high	high
Network	–	low	middle	high	high
Data Link	–	high	middle	middle	high
Physical	–	high	middle	low	high

Maintainability: Describes the ability to maintain the environment, i.e., how much effort is necessary to keep the system runnable.

Scenario creation: Describes the freedom in creating different experiment scenarios in terms of network topology, the number of nodes etc.

Scalability: Assesses the feasibility of large scale experiments with respect to the number of nodes etc.

Duration: Describes the experiment time. Variable means, that experiments can be conducted over long periods of time. In contrast, realtime means that experiments are conducted in real-world time.

Cost: Evaluates the cost of experiments. The cost is related to hardware and software costs.

In the upper part of Table I we have only two categories evaluated in the case of theoretical analysis. The other categories do not restrict the environment, since it depends heavily on the modeling capabilities of the researcher, e.g., scalability is not an issue here. In summary, we argue that a *theoretical analysis* of a complete WMN is not possible, but can only be done for particular components of the network. This environment provides a high degree of control and abstraction and at the same time a poor applicability of the results and conclusions. The *simulation* combines low cost with high flexibility for different types of network studies. The most important disadvantage is the limited applicability of results to the real world. The *virtualization* provides a healthy tradeoff between maintainability, scalability, and applicability. From our point of view, virtualization has some inherent advantages. The virtualization allows the development and testing of software as if it was executed on real mesh routers, but in a more repeatable and controllable way. It is easy to port the software to the real nodes of the testbed afterwards.

The highest degree of applicability and therefore transferability of results, conclusion, and system environment is given in the case of *real testbeds*. The main disadvantage of this environment is its low scalability and the complexity in experiment scenario generation.

When designing experiments to study performance parameters of WMNs it is important to have an idea which degree of realism can be expected from the study environments. In the lower part of Table I we have summarized them with respect to networking layers. This helps to determine which of the environments provides the researcher with the required degree of realism. The *theoretical analysis* approach does not provide any realistic instances of the network layers. In contrast a *real testbed* provides realism on all layers. *Simulation* typically provides a high degree on the data link- and physical layer. The upper layers are typically simplified. The degree of realism in *emulation* depends heavily on the parts which are represented by real hard- and software. In the case of *virtualization* the upper layers are real, since the virtualized machine and the OS provide all necessary functionalities. However, if VMs are coupled via a network the physical layer may have low realism, if both VMs are run on the same physical computer.

IV. UMIC-MESH – A HYBRID TESTBED FOR WMNS

In this Section we present our project *UMIC-Mesh*. The aim of this project is twofold. From the scientific point of view, the goal is to build a large and scalable WMN to pursue various networking studies. Considering real applications, the goal is to provide the members of the Computer Science Department with an easy way to get network access.

A. Motivation for a Hybrid Testbed

As we have seen in Section III, there are different possibilities to study wireless and mobile networks. Before we present our solution, we will review the development process of a testbed to point out the advantages of our approach. Basically, this process can be divided into two phases. In the first phase, the testbed is designed and realized. Considering the UMIC-Mesh, that is to design and realize a semi-managed WMN. Nevertheless, this step necessarily includes decisions on hardware and software. Obviously, the most important software decision to be made is the choice of the OS. For this purpose, Linux derivatives are widely used. Subsequently, in the second phase, the realization of network protocols and tools starts. As it is known from software engineering, this process is an iterative one and comprises *developing*, *distributing*, and *testing*. The step of developing a protocol or tool consists of the implementation and its debugging. Furthermore, the distributing step includes the installation of the implementation and the validation of this installation to ensure a correct distribution among all testbed systems. In a final step, the functionality and performance of the implementation have to be tested and evaluated respectively. In addition, if any failure occurs in one of these steps, debugging information has to be collected and analyzed. Preferably, the environment for developing and

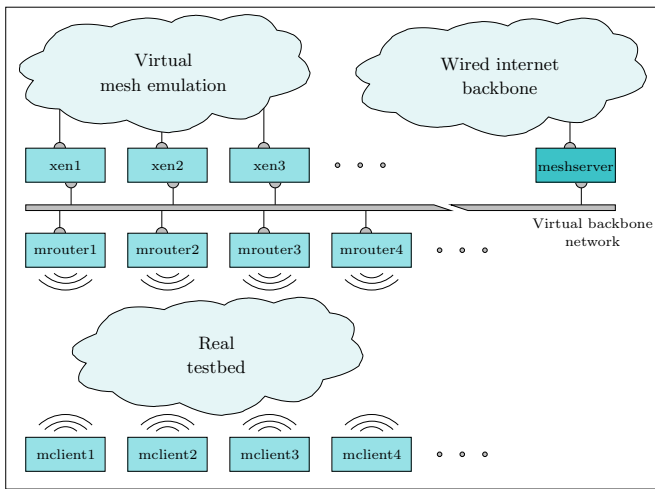


Fig. 2: Architecture of the UMIC-Mesh

testing should be as close to reality as possible—generally achieved best by utilizing real hardware and standard software.

All in all, the iterative software development process is a complex and labor-intensive undertaking. In particular, the distribution of new software versions is a challenging task, since new versions have to be frequently distributed among all systems of the testbed. A hybrid testbed that combines a real testbed and a virtualization environment offers a solution of the problems mentioned above. That means, the separate tasks of the software development process are distributed over the real part and the virtualization one. On the one hand, the virtualization environment takes over the *developing* task (implementation and debugging), the *validation*, and the *functionality testing*. On the other hand, *functionality testing* and *performance evaluation* takes place in the real testbed. Thus, there is no reason for emulating a wireless network interface, i.e., the physical and the data link layer. The following sections describe in detail the software, hardware, and virtualization component of the hybrid testbed.

B. System and Network Architecture

Figure 2 depicts the general system and network architecture of the UMIC-Mesh testbed. In accordance with the previous considerations that a hybrid testbed presents an appropriate environment to study WMNs, the testbed is realized by using two different components: a virtualization environment and a real testbed.

A main disadvantage of both real testbeds and virtualization environments is the high maintenance cost. Especially, if there is a failure in the course of a performance evaluation, the distribution of a new, corrected implementation is still labor-intensive. To minimize this maintenance cost a central configuration approach is used. As shown in Figure 2, a central server, the so-called *meshserver*, is integrated into the testbed. It serves two responsibilities, the *source functionality* and the *drain functionality*. First, the term “source functionality” describes the fact that the server provides several services to

the attached nodes. The most important service is the provision of an OS image to all nodes via network. Therefore, the basic setup is the same in each node of the hybrid testbed. The nodes may even share the same kernel including modules and drivers. Another important service provided by the meshserver is the Internet access, which is required for the mesh gateways. Second, the meshserver offers a *drain functionality*. That means, all important information about the real and emulated mesh networks is gathered and stored at the central server. Typical pieces of information are system log and SNMP messages or measurements results, which are stored in a database. This approach of a central log and information server enables a quick detection of any problem in the mesh testbed.

Besides reducing the maintenance cost, a central configuration and log server offers eased scenario creation and improved controllability. Since the configuration files are stored on a single central server, any WMN architecture (see Section II-A) can easily be realized. For example, it is possible to define which mesh routers should act as a gateways. Furthermore, the routing functionality for the clients can be enabled or disabled. Thereby, a routing mesh client can simply be reconfigured to a non-routing mesh client. Thus, the performance of different mesh architectures can be evaluated and measured.

All mesh routers and virtual machines are connected by a common network, the *virtual backbone network*. The term “virtual” emphasizes the fact that this network is solely used for booting and configuring the attached nodes as well as for the audit trail processing. That means, the clients in the real testbed cannot access the virtual backbone network. Thus, their data is not transmitted via this virtual network but is forwarded in a multi-hop fashion by the wireless network interfaces.

C. Testbed Realization

The UMIC-Mesh testbed is deployed in the complex of the Department of Computer Science at the RWTH Aachen University. The complex consists of one four- and two three-story buildings, which are all interconnected. The mesh routers are located in different offices at different floors.

As mentioned above, the UMIC-Mesh testbed consists of two parts, the virtualization environment and the real testbed. The remainder of this section describes the hard- and software used for each of these parts.

1) *Hardware*: At the moment, the real testbed part of UMIC-Mesh consists of 40 identical mesh routers. Each mesh router consists of a single board computer (SBC), two identical IEEE 802.11a/b/g wireless network interfaces based on the *Atheros AR5213 XR* and two omni-directional antennas. The SBC is a *WRAP.2E* board [5]. While the first card is reserved for router-to-router communication, the second one handles the router-to-client communication. In order to cleanly separate router-to-router from router-to-client communication, the first card uses 802.11g channel 1 and the second one channel 11. Both cards transmit at 100 mW and operate in a non-standard IBSS mode *ahdemo* [6]. The *ahdemo* omits the beacons and the BSSID mechanism of 802.11. This solves the tendency of the IBSS mode to form partitions which have different basic

service set identifiers (BSSID) despite having the same network identifier. Such partitions made it impossible to reliably operate the WMN with the IBSS mode. All mesh routers share the same ESSID pair, that is one ESSID for channel 1 and one ESSID for channel 11.

The virtualization part of our testbed consists of 7 standard Core 2 Duo PC with 1 GB RAM. With this amount of RAM it is possible to run about 10 virtual machines per host.

2) *Software*: As described in Section IV-B, one goal of the UMIC-Mesh implementation is to achieve a central configuration. For this reason a single OS image is provided to all nodes via network by using the *NFS* protocol. Consequently, there is no restriction to the use of a stripped-down Linux distribution fitting onto the CF cards of the routers. Therefore, a standard Ubuntu Linux distribution has been chosen. The central audit trail processing is realized with a combination of logging (syslog) and monitoring (SNMP).

On the one hand, the *madwifi-ng* driver is used to implement the WMN architecture in the real testbed. One of the most interesting features of *madwifi* is the virtual access point (VAP) mode that allows the operation of multiple concurrent virtual wireless devices running in different modes. In particular, it is possible to run one VAP in the AP mode and a second one in the ad-hoc mode. Thereby, both non-routing and routing mesh clients are connected by means of the same wireless network interface card. Otherwise, two separate cards would be needed, since non-routing mesh clients operate in the AP mode and routing mesh clients in the ad-hoc mode.

On the other hand, the nodes in the virtual environment are driven by *Xen*. The *Xen* project was chosen due to the fact that it employs the most efficient approach to a system virtualization, the paravirtualization. Furthermore, CPU manufacturers actively support its development. Thus, the upcoming integration of hardware virtualization support and the general *Xen* development are likely to make quick progress.

To emulate the multi-hop behavior a combination of the advanced networking features of the Linux kernel is used. At the core, there is a virtual network that exists on top of the virtual backbone as illustrated in Figure 2. For this, the tunneling protocol *Generic Routing Encapsulation (GRE)* [7] is utilized. It emulates a broadcast medium on top of an existing IP network by using a multicast address for its broadcast traffic. To control the communication between all participants of the network, standard packet filtering as provided by *iptables* is employed. There is no effort put in the emulation of a wireless medium, because the virtualization environment is used only for software development and functionality validation. Network performance evaluation is solely done in the real testbed.

Currently, the *DYMO* routing protocol and the *OLSR* protocol are employed. We made this choice since this protocols are typical representatives of the two routing philosophies in MANETs: reactive and proactive routing.

As previously mentioned, all our testbed nodes are booted via the network. The vital parts of this process are getting an IP configuration and a kernel to boot. We utilize a combination of *EtherBoot* and *PXELinux* for this purpose.

V. RELATED WORK – EXISTING TESTBEDS

In this Section we will present some wireless mesh network projects, which have the aim to create a testbed.

A. MIT Roofnet

The *MIT Roofnet project* [8] consists of 37 nodes based on PCs running Linux and the Click [9] modular router. Each node has an IEEE 802.11b network interface and an omni-directional antenna. The goal of the project is to provide Internet access to the students. The Roofnet nodes are run by volunteering students. All nodes are running on the same channel. There are a total of 4 gateways which provide Internet access to Roofnet. The Roofnet mesh routers can allocate IP addresses via DHCP for the mesh clients and provide them with access to the Roofnet as well as the Internet. Roofnet's routing protocol is denoted as *Srcr*, which is similar to DSR.

B. Microsoft Mesh Networking

Microsoft Research is working on a community mesh network. The goal of the project is to enable the building of a community mesh network, which allows the residents of a neighborhood to share existing Internet gateways. The core of the project is the *Mesh Connectivity Layer (MCL)*. Each node applying MCL can route data for other nodes in the mesh network. For this, MCL uses a modified version of DSR, denoted as LQSR [10]. From the network architecture point of view, MCL is located between the Networking Layer and MAC Layer. MCL uses MAC addresses for routing. By appropriate configuration of Windows it is also possible to enable Internet connectivity through a gateway.

C. UCSB MeshNet

Belding-Royer et al. [11] run a mesh network project called *MeshNet* at the University of California at Santa Barbara. In MeshNet each mesh router consists of two Linksys WRT54G wireless devices. One of the devices is used for routing within the mesh network and the other device is used for the management of the router. On the WRT54G runs OpenWRT, a special Linux distribution, and a modified version of the AODV routing protocol. The important difference is the used metric. Instead of the shortest-hop as the routing metric they use a reliability-based routing metric.

D. WMN Testbed at Purdue University

Hu et al. run a wireless mesh project at the Purdue University, which is called *Mesh@Purdue* [12]. It consists of 30 nodes. A so-called MAP mesh router is a small form-factor desktop equipped with two wireless interface cards and a wired ethernet network interface. The latter one is used for management purposes. Besides the MAP mesh routers, there are also some Laptop and iPAQ PDAs which are used as hosts. These hosts can access the Internet over the mesh network. The research group deploys a modified AODV and OLSR as the routing algorithms within the mesh network.

E. WMNs Research at Georgia Tech

Akyildiz et al. [2] run a WMN project at the BWN Lab, Georgia Institute of Technology. It consists of 15 nodes. The goal of the project is to study various performance metrics of WMNs, e.g., the effects of inter-router distance, backbone placement and clustering. Furthermore, existing protocols are re-investigated to review their performance in the testbed, e.g., end-to-end delay and throughput.

F. WMN at Carleton University

Kunz et al. [13] run a WMN project at the Carleton University. Each mesh router is equipped with two wireless interface cards. The first is used for the communication within the WMN among the wireless mesh routers and the latter is used for the communication with the clients. The WMN provides Internet access to the clients. A mesh router includes 64 MB RAM, and 16 MB of Flash memory. Thus, the mesh router does not need a hard drive. The mesh routers are running μ Clinux, a popular embedded Linux distribution, and QoS OLSR from CRC [14] as the routing algorithm. To provide clients with addresses DHCPv6 is used. Thus, the WMN deploys only IPv6. If clients need to access an IPv4 network, e.g., the Internet, the packets are tunneled by applying the Dual Stack Transition Method (DSTM) [15].

G. Hyacinth

Hyacinth [16] is the WMN project at the State University of New York. Each *Hyacinth* node is a small form-factor PC running Windows XP and is equipped with three IEEE 802.11a wireless interfaces. The mesh nodes obtain IP addresses by using a two-step method which is based on DHCP. In the first step, a mesh node allocates a temporary IP address from the reserved range 192.168/16. In the second step, a unique IP address is obtained from a global DHCP server, which is placed in the wired network. Each mesh node acts also as a local DHCP server and can assign IP addresses to mobile stations. For this each mesh node receives a range of IP addresses from the global DHCP server. This ensures, that each mobile station has a unique IP address in the whole WMN. Furthermore, roaming of mobile stations is supported, since all mesh nodes act also as a home/foreign agent like in Mobile IP (MIP).

VI. CONCLUSIONS AND FUTURE WORK

Nowadays, the study of wireless and mobile networks is mainly based on simulation. Although simulation environments provide the researcher with many advantages like low cost, flexibility, and controllability they also possess some disadvantages which degrade their usefulness. The prime disadvantage comes from the high dynamic and complexity of mobile and wireless networks, which is caused by the high influence of the environment. It has been shown that the performance of wireless networks in simulation and in real world differ very much. The counterpart to simulation studies are real world testbeds. They provide the same environment for the researcher as it exists in the production world. All results and inferences can be easily transferred to real world systems. However, real

world testbeds have other limitations, e.g., typically they do not scale, since it is very hard to set up a large wireless network testbed. A possible solution for this dilemma is to apply a hybrid testbed.

Based on our evaluation, we introduced the UMIC-Mesh. It is a hybrid wireless mesh networking testbed, which consists of real hardware, standard Linux software and a virtualization environment. The former ensures a high degree of realism and enables us to transfer the results and conclusions into the real world. The second part provides us with a flexible environment to develop various networking protocols. We described in detail our hardware and software setup to give other researchers the possibility to build similar testbeds.

As next we will work on a web based management console, that visualizes the current status, e.g. routing tables, of testbed and allows the comfortable configuration and control of the WMN testbed.

REFERENCES

- [1] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, March 2005.
- [2] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, March 2005.
- [3] S. Ruffino, P. Stupar, T. Clausen, and S. Singh, "Connectivity scenarios for manet," Internet Draft, July 2005. [Online]. Available: <http://www.watersprings.org/pub/id/draft-ruffino-conn-scenarios-01.txt>
- [4] M. Carson and D. Santay, "NIST Net: a Linux-based network emulation tool," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 111–126, July 2003.
- [5] PC Engines, "WRAP router platform (Version WRAP.2E)." [Online]. Available: <http://www.pce.com.tw>
- [6] Madwifi Project, "Madwifi – multiband atheros driver for wireless fidelity." [Online]. Available: <http://madwifi.org/>
- [7] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic routing encapsulation (GRE)," RFC 2784, March 2000. [Online]. Available: <http://www.ietf.org/rfc/rfc2784.txt>
- [8] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceedings of the 11th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'05)*. ACM Press, August 2005, pp. 31–42.
- [9] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, August 2000.
- [10] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'04)*. ACM Press, September 2004, pp. 114–128.
- [11] K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "A framework for the management of large-scale wireless network testbeds," in *Proceedings of the 1st Workshop on Wireless Network Measurements (WiNMe'05)*, April 2005.
- [12] Purdue University, "Purdue University wireless mesh network testbed." [Online]. Available: <https://engineering.purdue.edu/MESH>
- [13] Carleton University, "Wireless mesh networking." [Online]. Available: <http://kunz-pc.sce.carleton.ca/MESH/index.htm>
- [14] Communications Research Centre, "CRC OLSR." [Online]. Available: <http://www.crc.ca/en/html/manetsensor/home/software/software>
- [15] J. Bound, L. Toutain, and J. Richier, "Dual stack IPv6 dominant transition mechanism (DSTM)," Internet Draft, October 2005. [Online]. Available: <http://www.watersprings.org/pub/id/draft-bound-dstm-exp-04.txt>
- [16] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'05)*, vol. 3. IEEE Communications Society Press, March 2005, pp. 2223–2234.