The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
15-17 September 2011, Prague, Czech Republic

Virtualization of Wireless LAN Infrastructures

Ghannam Aljabari[1] , Prof. Dr.-Ing. Evren Eren[2]
[1] Palestine Polyetchnic University, Hebron, Palestine, Email: galjabari@ppu.edu
[2] University of Applied Sciences Dortmund, EFS 42, D-44227 Dortmund, Germany, Email: eren@fh-dortmund.de

***Abstract* – In wired Ethernet networks (IEEE 802.3), physical network interfaces can be connected to different network segments or shared among multiple virtual machines. In wireless LAN (IEEE 802.11) sharing wireless network interfaces is recognized to be a difficult task. However, virtualization can solve this problem. In this paper we will introduce a viable solution to deploy virtualized wireless networks by means of open source virtualization techniques. We present the design, implementation, and performance testing of this solution. Results have shown that the proposed solution can support multiple virtualized wireless networks without compromising the performance.**

***Keywords* – virtualization; wireless; virtual network; hypervisor**

I. Introduction

Virtualization of wireless LANs (WLANs) has become one of the important issues in network virtualization and also for cloud computing. It is useful in many scenarios: hosting multiple wireless service providers on a single shared physical infrastructure, providing wireless services with different authentication mechanisms, and for virtual testbed environments. Hence, there is some research activities in this field [1]–[3].

The goal of network virtualization is to combine network functionality into a common virtualized environment and to enable multiple logical networks to operate on the same underlying physical infrastructure [4], [5]. However, most of the virtualization approaches are mainly developed for wired Ethernet networks.

Existing virtualization approaches require a separate physical wireless LAN network interface for each virtual machine to have its own wireless network. By means of open source virtualization techniques, it is possible to create multiple wireless networks through one physical wireless LAN network interface, so that each virtual machine has its own wireless network. This paper aims at demonstrating this viable approach.

The remainder of this paper is organized as follows: Section II describes the related work and introduces virtualization in general and wireless LAN virtualization specifically; Section III outlines the proposed solution; Section IV explains the implementation of the testbed; Section V reports the performance testing results; Section VI concludes and depicts future work.

II. Related Work

A. Virtualization Concept

Virtualization techniques enable running multiple operating systems and multiple applications concurrently on the same physical machine, and in a manageable manner, eliminating the need for multiple physical machines and thus reducing costs in hardware and infrastructure resources. Each virtual machine has its own operating system and application(s) such as the physical machine [1], [6].

The primary benefits offered by virtualization are *resource sharing* and *isolation*. Unlike real environments where physical resources are dedicated to a single machine, virtual environments share physical resources such as memory, disk space, and network devices of the host machine with several virtual machines. By isolation, applications running on one virtual machine cannot see, access, and use resources on other virtual machines [6].

Furthermore, security is enhanced by separating services on multiple virtual machines. If one service is compromised, other services remain unaffected. A vailability is improved by migrating virtual machine to another physical machines is host machine should fail. Scalability is improved because additional physical and virtual machines can be added or removed easily without the need to shutdown running virtual machines [6].

Virtualization provides a software abstraction layer between the hardware and the operating system. This layer is called Virtual Machine Monitor (VMM) or hypervisor. The main task of the VMM is to manage hardware resource allocation for virtual machines and to provide interfaces for additional administration and monitoring tools [6].

There are various approaches with respect to virtualization. In the so called *full virtualization* approach, the VMM runs on top of the host operating system acting as a user space program. As a result, virtual machines and guest operating systems run on top of virtual hardware provided by the VMM. However, the VMM has to provide sufficient virtual devices to allow guest operating system to run without modification. This architecture is depicted in Fig. 1 [6]. KVM (Kernel-based Virtual Machine) [7] is a full virtualization solution which adds VMM capability to Linux operating systems. Using KVM, multiple virtual machines can be created and operated with unmodified operating systems, since KVM benefits from CPU hard-

ware virtualization extensions such as Intel VT and AMD-V [8].

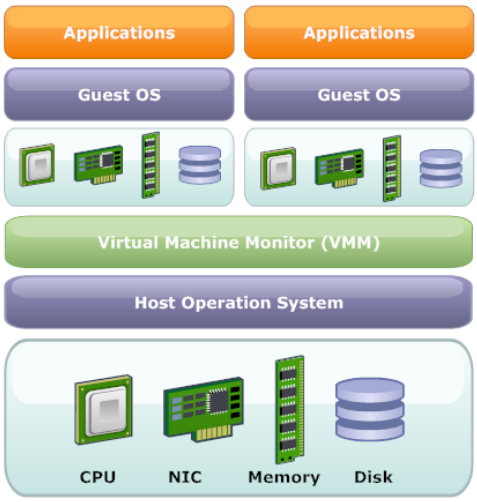

Figure. 1. Full virtualization

In the *paravirtualization* approach, guest operating systems have to be modified (adapted) to be able to operate in the virtual environment. The VMM provides a software interface (API) connecting the underlying hardware and the guest operating system. The function of this interface is to improve performance and efficiency behavior. However, virtual machines rely on physical device drivers of the host machine. Paravirtualization is used by the Xen open source hypervisor to run multiple virtual machines with modified operating systems [6]. More recently, Xen also supports full virtualization based on hardware virtualization extensions.

B. Network Virtualization

Network virtualization (also denoted as overlay network) allows multiple heterogeneous architectures to run concurrently in a shared network environment [4]. Network virtualization often combines hardware and software resources to deploy virtual networks for different architectures.

By means of virtual networking, virtual machines can be connected to virtual networks in the same way as to physical machines. However, the way of deploying and managing virtual networks is different from physical (real) networks.

For the time being, virtualization techniques can realize virtual Ethernet interfaces, virtual switches and virtual routers, as shown in Fig. 2.

Today's virtualization solutions typically realize virtual Ethernet interfaces by emulating legacy Ethernet adapters. The virtual Ethernet interface has its own MAC (Layer 2) and IP address (Layer 3). As a result, a virtual machine has the same networking properties as a physical machine [1], [4].

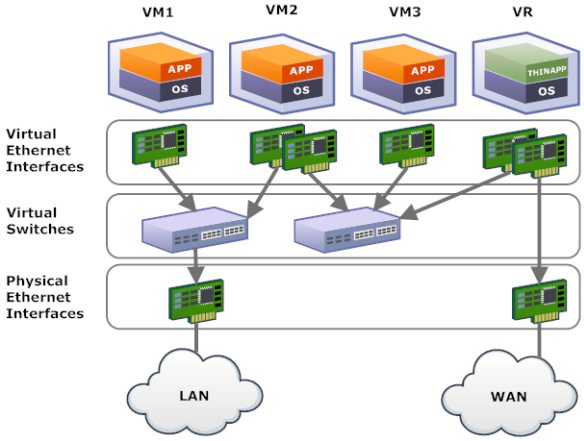

Figure. 2. Virtual networking components

By means of virtual switches, virtual machines on the host machine can communicate with each other. In the open source domain two solutions are viable: VDE (Virtual Distributed Ethernet) [9] and Open vSwitch [10]. A VDE switch operates at Layer 2, while Open vSwitch operates at Layer 3 with advanced features. Both provide switch functionality and support standard Virtual LAN (VLAN) [11], [12].

Virtual machines can be configured with one or more virtual Ethernet interfaces to offer different virtual appliances for virtualization and cloud computing environments. Besides, open source routing and security solutions such as Vyatta [13] complement virtual routing, firewalling, and VPN functionality, if needed.

Virtual routers (VR) are essential components in virtual networking infrastructures. They operate in much the same way as physical routers, forwarding and routing packets based on standard routing protocols such as RIP, OSPF, etc..

C. Wireless LAN Virtualization

With the introduction of IEEE 802.11n and the increase in bandwidth needs, wireless LAN virtualization is required as a viable and low cost alternative for deploying multiple wireless networks with different authentication mechanisms [14]. It is a form of resource virtualization where logical resources are created by partitioning hardware resources into virtual interfaces or ports [1], [6].

All virtual interfaces operate concurrently without considering the physical nature of the wireless medium as well as physical management tasks. Each virtual interface abstracts a single wireless device and has its own wireless network and its own unique MAC address. From the application's perspective, the virtual wireless network behaves like wired Ethernet, but is wireless [2], [14].

In the wireless medium, radio resources can be shared and thus virtualized in different ways such as in time, space, and frequency. By splitting the wireless medium

into different channels, to each channel a specific time slot (Time Division Multiplexing), space (Space Division Multiplexing), frequency (Frequency Division Multiplexing) or combinations can be allocated. To conserve frequency channels, virtualization of the wireless medium uses the same radio frequency for multiple virtual interfaces, each with its own Service Set Identifier (SSID) or network name. Efforts also have been made in splitting the wireless medium by assigning different radio frequency channels to the virtual interfaces or operators [15], [16].

Using wireless LAN virtualization, a virtual interface (VIF) can be configured to operate as an access point (AP) and also as station (STA) device. In this way, several virtual APs can be configured on top of solely one physical wireless device. Each virtual AP independently keeps the configuration and service of the wireless network [2]. Also, by virtualizing the WLAN interface, a wireless device can be connected to several networks simultaneously; one virtual interface can be connected to an AP, while another virtual interface operate as an AP.

When a single physical AP supports multiple virtual APs, each virtual AP appears to stations as an independent physical one. Since each virtual AP is logically separated, wireless LAN providers may use virtual AP to offer multiple services on the same physical infrastructure. Alternatively, virtual APs can be shared by multiple providers allowing each provider to offer separate services for their subscribers [3].

A virtual AP acts as a master device in a managed wireless network and allows client devices to communicate with each other by managing and maintaining a list of associated stations or clients. It also supports different security mechanisms (authentication and encryption) [17]. One example in the open source software domain is hostapd for controlling wireless authentication and association [17], [18].

A virtual STA functions as a managed device in a managed wireless network and is associated to an access point after successful authentication [14]. wpa_supplicant is a well-known example in the open source software domain [17], [19].

III. Proposed Solution

By combining wireless LAN virtualization with virtualization software (hypervisor), wireless LAN interfaces can be shared among several VMs. Each VM can be assigned to one or more virtual wireless interfaces.

Our proposed solution shown in Fig. 3 follows a combined approach of software and hardware. It is based on a new technology emerged in the wireless market recently offering a viable solution for wireless LAN virtualization. It supports concurrent wireless connections sharing the same physical layer (PHY) of the wireless LAN device. We extended this capability to operate in the virtualization environment, where VIFs have been configured to operate in one of the wireless operating modes, specifically the AP mode, and then can be assigned to various virtual networking components.

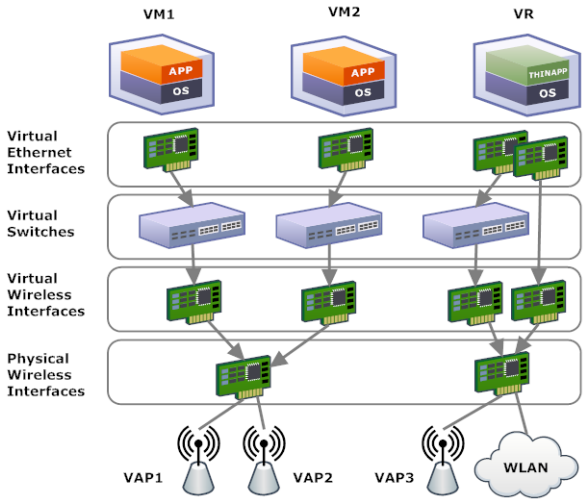

Figure. 3. Virtual Wireless LANs

This approach is suitable for virtualizing wireless LAN infrastructures, where VMs run on a server or network appliance. In mobile client environments, the wireless LAN virtualization approach is much more complex, since the VM runs on the client device and has to be aware of the wireless interface to be able to establish its own wireless connection with an AP. This requires the inclusion of virtualization extensions to wireless device drivers as well as wireless LAN management functions to the virtualization software. Virtual WiFi [20] proposes an approach to support wireless LAN client functions inside VMs and implements a prototype in KVM with Intel WiFi devices.

Our approach is intended to deploy multiple wireless networks on a single shared physical infrastructure with different security standards. At the same time, these wireless networks should be isolated from each other at a satisfactory performance level comparable to native hardware environments.

Since this approach adds wireless LAN infrastructure functionality to virtual environments, it can be deployed for different wireless LAN systems on the same host machine such as authentication services and intrusion detection, providing secure wireless LAN on a single box.

To emulate a physical AP, it is necessary to provide the emulation at different layers such as layer 2 (MAC), layer 3 (IP), and above. At the MAC layer, the behavior of a physical AP is being emulated by allocating a distinct MAC address and SSID to each virtual AP. At the IP layer, it is emulated by allocating a distinct IP address and potentially a Fully Qualified Domain Name (FQDN) to each virtual AP. At higher layers, the emulation can be carried out by providing each virtual AP with a unique authentication and accounting configuration (such as shared

key, or EAP methods with RADIUS authentication, or SNMP communities.

A virtual AP is constructed by configuring the VIF to operate in AP mode. This sets the main functionality of the wireless AP such as IEEE 802.11 operation mode and SSID. Once configured, the wireless interface is attached to a virtual switch to enable MAC forwarding similar to a physical AP. Then, the virtual AP interface is connected to virtual machines the same way as the virtual Ethernet interface.

IV. Implementation

The virtual interface capability given by the Atheros chipset [21] allows implementing multiple IEEE 802.11 networks on a single physical wireless card with Linux (Linux kernel version 2.6.33 and higher), since it includes a wireless driver supporting multiple VIF configurations.

The wireless driver for Atheros WLAN devices was initially developed by the madwifi project [22], then became part of the the Linux kernel. The implementation model of Linux kernel WLAN driver is currently based on SoftMAC wireless devices, where most of the MAC layer functionality is managed by means of software. For the time being, Linux kernel supports all wireless modes with PCI/PCI-Express Atheros WLAN devices only [17].

In our testbed, we used a conventional PC with a wireless card based on the Atheros IEEE 802.11n chipset. It had an Intel Core 2 processor with VT support, Fast Ethernet interface and 3 GB RAM.

Ubuntu Linux has been chosen to host the virtualization environment of the testbed. We used KVM as backend for virtualization and libvirt as frontend for managing virtual machines. With libvirt, there comes two management tools: virt-manager as graphical user interface (GUI) and virtsh as command line interface (CLI) [23].

The virtual wireless interfaces have been created using a CLI configuration utility in Linux named “iw” [17]. Once created, the interfaces have been configured to function as virtual AP or virtual STA interfaces (supplicants). It is essential for all VIFs to have a unique MAC address, which can be assigned with “macchanger” utility [23].

A virtual AP has been implemented using the hostapd daemon or background service. hostapd handles all aspects of IEEE 802.11 functionality and authentication configuration [18]. The virtual AP interface has been connected to a VDE switch interface to enable MAC forwarding, similar to a physical AP.

For testing purposes, several virtual wireless routers have been hosted on the PC with a shared Internet connection.

As illustrated in Fig. 4, we created three virtual APs (802.11g) and three virtual routers running Vyatta. Each virtual router had two virtual Ethernet interfaces. One of them was connected to the virtual AP interface and the other to the physical Ethernet interface using Linux interface bridging. Each virtual router acted as DHCP server

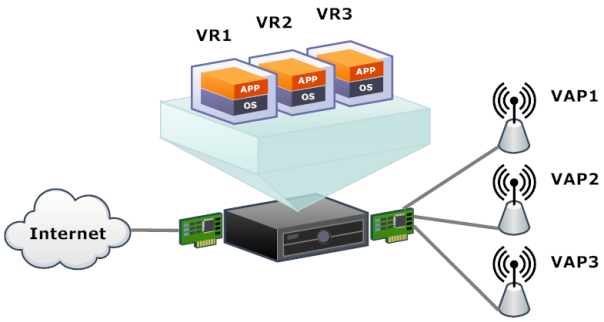

Figure. 4. Multiple virtual wireless routers

and DNS forwarder for the virtual wireless network and each virtual AP broadcasted different SSIDs to distinguish the wireless networks. NAT functionality was also added to the virtual routers to maintain public IP addresses and to enhance wireless network security.

Using these virtual routers, different wireless LAN clients could access the Internet with different wireless LAN security mechanisms.

V. Performance and Results

We have conducted some tests to understand the impact of the virtual software layer on wireless LAN networks. The objective of the tests was to compare and quantify the performance of both conventional and virtualized wireless networks. Testing WLAN performance primarily included two test metrics: throughput and response time. These performance metrics were used to evaluate the applicability of our approach for WLAN infrastructure virtualization since the virtual networks had to handle the same kind of traffic as conventional networks.

The throughput of WLAN is defined as the speed with which a user can send and receive data between the client and the AP [24]. Throughput varies across the WLAN’s coverage area. For this reason, we placed the test machines at close range to operate on the maximum available channel bandwidth.

Theoretically, the maximum TCP rate of 802.11g network is 24.4 Mbps and the maximum UDP rate is 30.5 Mbps. UDP throughput is higher than TCP throughput because there is less protocol overhead associated with UDP [24]. Therefore, TCP throughput is the most relevant metric in our performance measurements.

To measure the throughput, we used IPerf and JPerf as graphical interface [25]. IPerf tool was used to measure TCP and UDP throughput in two directions: uplink direction (from the client to the virtual AP) and downlink direction (from the virutal AP to the client). To measure response times or latencies, we used ping. Ping is used to measure the round-trip time between the client and the virutal AP.

In our test setup, IPerf was installed on two machines; the machine which hostes the virtual wireless routers

functioned as IPerf server and the wireless client machine as IPerf client. IPerf was configured on the wireless client to run test for 60 seconds in both directions and provided values in Mbps.

We performed the same test in both native and virtual environment. In the native hardware environment, the tests were performed between a remote client and host machine running three virtual APs without virtualization. In the virtual environment, the tests are performed between a remote client and a VM directly attached to the virtual routers. In this case, the wireless traffic passing through the virtual routers.

Fig. 5 depicts the throughput test results where all throughput results have been averaged over three measurements. The average downlink/uplink TCP throughput is 21.8/18.6 Mbps in native hardware environment and 21.4/18.2 Mbps in virtual environment. Latency test results show that the average round-trip time in native hardware environment is 1.1 ms and 2.1 ms in the virtual case. This latency overhead comes form the virtualization layer.

The results show that our proposed solution achieves performance metrics comparable to native hardware environment.

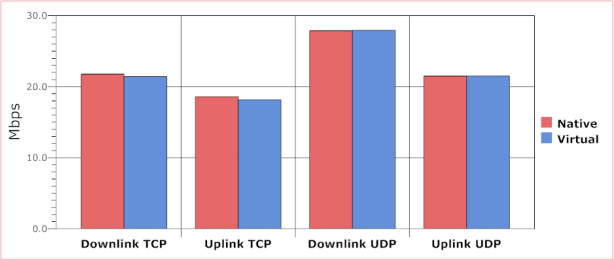

Figure. 5. Throughput test results

VI. Conclusion and Future Work

In this paper, we introduced a virtual networking infrastructure using conventional virtualization techniques. Also, we proposed a practical solution to realize virtual wireless networks by combining wireless LAN virtualization with open source virtualization techniques.

Our approach adds wireless LAN functionally to virtualization environments. Summarizing some of the benefits, we can conclude that our proposed solution:

- enables virtualized wireless LAN architectures.
- builds wired and wireless networks without deploying physical infrastructure.
- adds wireless LAN management and control functions to virtualization environments.

For the future, it is planned to investigate performance measuring and optimization with Xen open source hypervisor. Also, we will design a testbed for virtualization of wireless LANs with different security infrastructures.

References

[1] J. Lee and Y. Moon, "Research on virtual network for virtual mobile network," in *Second International Conference on Computer and Network Technology (ICCNT)*, 2010, pp. 98–101.

[2] T. Hamaguchi, T. Komata, T. Nagai, and H. Shigeno, "A framework of better deployment for wlan access point using virtualization technique," in *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2010, pp. 968–973.

[3] (2010, 12) http://aboba.drizzlehosting.com/ieee/11040238000wng-definitionvirtualaccesspoint.doc.

[4] N. M. N. K. Chowdhury and R. Boutaba, "Network virtualization: State of the art and research challenges," *IEEE Communications Magazine*, pp. 20–26, 2009.

[5] M. Anwar and N. Feamster, "Building a fast, virtualized data plane with programmable hardware," in *ACM SIGCOMM Workshop on Virtualized Infastructure Systems and Architectures*, 2009.

[6] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues," in *Second International Conference on Computer and Network Technology (ICCNT)*, 2010, pp. 222–226.

[7] (2010, 12) http://www.linux-kvm.org.

[8] A. Kivity, "Kvm: The linux virtual machine monitor," in *Ottawa Linux Symposium (OLS)*, 2007, pp. 225–230.

[9] (2010, 12) http://wiki.virtualsquare.org.

[10] (2010, 12) http://openvswitch.org.

[11] R. Davoli, "Vde: Virtual distributed ethernet," in *Proceedings of the first International Conference on TRIDENTCOM*, 2005.

[12] J. Pettit, J. Gross, B. Pfaff, M. Casado, and S. Crosby, "Virtual switching in an era of advanced edges," in *2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC-CAVES)*, 2010.

[13] (2010, 12) http://www.vyatta.org.

[14] "Wireless lan viertualization: Twice the network at half the cost," White Paper, Meru Networks, 2010.

[15] S. Singhal, G. Hadjichristofi, I. Seskar, and D. Raychaudhuri, "Evaluation of uml based wireless network virtualization," in *IEEE conference on Next Generation Internet Networks (NGI)*, 2008, pp. 223–230.

[16] S. Perez, J. M. Cabero, and E. Miguel, "Virtualization of the wireless medium: A simulation-based study," in *IEEE 69th conference on Vehicular Technology (VT)*, 2009, pp. 1–5.

[17] (2010, 12) http://wireless.kernel.org.

[18] (2010, 12) http://w1.fi/hostapd.

[19] (2010, 12) http://hostap.epitest.fi/wpa_supplicant.

[20] L. X. et al., "Virtual wifi: Bring virtualization from wired to wireless," in *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE)*, 2011.

[21] (2010, 12) http://www.atheros.com.

[22] (2010, 12) http://madwifi-project.org.

[23] (2010, 12) http://www.techotopia.com/index.php/ubuntu_10.x_essentials.

[24] "Methodology for testing wireless lan performance with chariot," White Paper, Atheros, 2003.

[25] (2010, 12) http://iperf.sourceforge.net.