Time Division Configuration Multiplexing of Wireless Interfaces

Hakan Coskun
Next Generation Networks
Technische Universität Berlin
Email: hakan.coskun@tu-berlin.de

Tom Pfeifer
Next Generation Networks
Technische Universität Berlin
Email: t.pfeifer@computer.org

***Abstract*—Virtual wireless interfaces are not as easy to manage as virtual fixed interfaces. Compared to Ethernet, 802.11 supports several operational modes for the MAC-layer. According to the way of operation, virtualizing the wireless interface means to switch between the available configurations and to track the states of each network connection and keep them consistent between the virtual interfaces. In this paper we shed some light on the technical requirements and methods used to achieve virtualization of 802.11 interfaces for stations and access points. We propose a mechanism that allows a virtual wireless interface in access point mode to run side by side with another virtual wireless interface, each configured to a different channel. An experimental performance evaluation of the proposed mechanism shows promising results.**

I. Introduction

Right from the start 802.11 supported different services, each of them enabling different scenarios [1]. The basic service set (BSS) and the extended service set (ESS) are usually called infrastructure mode and provided by an access point (AP) which has the control of the channel usage. On the contrary, the independent basic service set (IBSS), often called ad-hoc mode, does not require any configured infrastructure for operation. In this mode networks are setup spontaneously, i.e. wireless stations (STA) communicate directly with each other in a peer-to-peer manner.

Beside these different services, 802.11 networks support multiple channels for data transmission. Wireless network interfaces are tuned into one channel only and configured to operate either as STA or as AP. One possible approach to overcome this limitation is to equip the mobile node with multiple radio interfaces which are tuned into multiple channels at the same time [2] and operate in different modes. As this is a costly solution, virtualization techniques can be used on the 802.11 MAC layer that create the illusion of having multiple physical WiFi Network Interface Cards (NICs) inside the node, allowing the same functionality as in multi-radio scenarios [3].

Usually, 802.11 wireless LAN cards and drivers are designed to operate exclusively with one configuration which is composed of a set of different parameters, like network name, channel, mode, etc. Virtualization of the wireless interface means to break this limitation and to allow different configurations on top of the same physical NIC. The challenge is to switch between the available configurations and to track the states of each network connection and keep them consistent between the virtual interface and the corresponding attachment point. Each configuration comprises different aspects, like physical configuration (channel, mode, timers, etc.), operational configuration of the card (mode, encryption, mac address, etc.). One of the major problems occurs when interfaces require hardware setups, for instance different channels. But this would mean that interface activities are handled mutually exclusive. Thus, their operation has to be time-scheduled and coordinated accordingly, since depending on the mode of the virtual interface, its absence from the channel might break the communication at all.

According to the mode of operation, virtualization has to take different requirements into account. In our previous work [4] we covered the station mode. In this paper we shed some light on the technical requirements and methods used to achieve virtualization of 802.11 interfaces in infrastructure based networks and propose a mechanism that allows to run a virtual wireless interface in access point mode side by side with another virtual wireless interface, each configured to a different channel.

The remainder of this paper is organized as follows. Section 2 introduces related work on this area. Enabling virtualization on 802.11 interfaces and related challenges are covered in section 3. Our proposed virtualization mechanism is introduced in section 4 and evaluated in section 5. Finally, section 6 concludes the paper and gives an outlook of future research.

II. Related Work

The research community already presented some approaches [5], [6], [7] how one WLAN network interface can be virtualized to simultaneously connect to multiple wireless networks even on different channels [1].

Using the power save mode (PSM) feature available in 802.11 networks, a station is able to connect to more than one infrastructure network simultaneously, without having to repeat the association procedure with every network switch. Instead of entering the doze state, the station uses the time interval for sleep (Listen Interval) to switch its interface to another network. The latter method was introduced first by

[1]Microsoft introduced virtualization of wireless cards as a feature in Windows 7, allowing to run an access point interface and a station interface side by side on the same WiFi card (without channel switching)

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

Chandra and Bahl [8]. They proposed *MultiNet* (now known as *VirtualWiFi*), a software-based approach that virtualizes a wireless card by inserting an intermediate layer called *MultiNet Protocol Driver* between the IP and MAC stacks. This layer virtualizes the wireless card and switches it across multiple networks without the user having to know about it. Based on the same scheme, members of Telefonica Research in Spain proposed *WiSwitcher* [7]. Their approach enables a single radio wireless client to be simultaneously associated with several APs operating in different radio frequency channels. Therefore they can achieve a higher accumulated throughput over them. They studied the relation and the impact of the switching frequency in the packet loss under off-the-shelf APs.

Nicholson, Wolchock and Noble presented *Juggler* [6], an enhancement of the MultiNet virtualization approach, built into the Linux kernel. By employing an autonomous kernel module between the network and link layers called *Juggler* and a user space process called *jugglerd* they comprise advantages of using virtual interfaces, such as sharing a small percentage of time for AP discovery through scanning to make a soft handoff, the benefits through data striping especially in scenarios where wireless bandwidth exceeds the wired bandwidth at the end connection of the access point, and the creation of an ad-hoc network while being connected to an infrastructure network. Juggler improved especially the time it takes to switch from one virtual interface to another and uses power save mode in order to buffer all the packets sent to the virtual WiFi current inactive.

III. Challenges

The standard does not cope with virtualization at all. Once a mobile station is associated with an AP, the connection keeps up until the station management entity (SME) or MAC Sublayer Management Entity (MLME) infer that the connection is lost. 802.11-based systems monitor the state of the wireless link by several means. Only if the characteristics of the link degrade below specific thresholds, the connection is seen as lost and the association of the station is deleted, leading to a scan for new networks.

On the access point, the activity of the stations might be tracked by an **inactivity timer**. If the station is inactive for a specific amount of time, the association might be removed.

On the client-side, two monitors are started when an association is made. The **beacon monitor** tracks the availability of the access point. Once started it passively monitors the channel by counting the beacon losses. If too many beacons are lost consecutively, it signals the unavailability of the AP. The **connection monitor** is actively probing for the associated BSS, which means that probe requests are sent out and responses are observed. In the case of a non-responding AP, the association is discarded and the STAs starts to look for new networks. But this leads to several well-defined steps, which are time consuming.

A common feature of STAs and AP is to monitor the number of frame retransmissions. If the retransmission limits are crossed, the link quality is considered as too bad and the association is deleted.

Due to these facts, channel switching imposes new problems on the virtualization. If two virtual interfaces in access point mode should be tuned to different channels, it has to be assured that associated clients are not forced to disassociate because of an uncoordinated absence of the access point or station. To overcome this issue, the management of the card requires some modifications that can maintain the state of the card and the network interface, thus enabling the usage of different modes and a fast switching between them.

IV. Approach

In our previous work [4], we proposed to use the power save mode (PSM) for station interfaces and Point Coordination Function (PCF) for access point interfaces. In this section we just sketch the first and elaborate in detail on the latter, and propose a mechanism for a coordinated deferral of stations. This will allow the access point to switch channels without losing its clients, because of a unavailability for a specific time.

We will call the time period in which the AP serves the BSS an **activity period** and the dormant phase of the virtual interface an **inactivity period**.

A. Virtual Station Interfaces

As stated above, we already presented how an intermediate layer below IP can be realized that manages the state information of all network connections as virtual interfaces and has also the full control over them. Using the power save mode (PSM) feature available in 802.11 networks, a station is able to connect to more than one infrastructure network simultaneously, without having to repeat the association procedure with every network switch. By indicating its intention to sleep, the associated AP buffers all incoming frames. Thus, a PSM STA can sleep and miss a specific number of Beacons, without losing any data traffic or disconnecting from the network. Virtualization in this context can be achieved by using the time interval for sleep (Listen Interval) to switch its interface to another network instead of entering the doze state.

B. Virtual Access Point Interfaces

An AP can provide STAs with the illusion of multiple physical APs within the same enclosure by using different SSIDs inside a beacon or in subsequent beacons. Each of these APs is a virtual access point (VAP). A VAP is a logical entity that exists within a physical Access Point and is bound to a virtual network interface. When a single physical AP supports multiple VAPs, each Virtual AP appears to stations (STAs) to be an independent physical AP, even though only a single physical AP is present. Off-the-shelf wireless access point support VAPs with different configurations (MAC address, SSID, QoS Parameters, etc.). Nevertheless, the VAPs run side by side on the same channel, competing for channel access.

This is due to the fact that the a channel switch would to lead to an absence of the central coordinator. This is not considered by the standard. Even worse, since wireless nodes may be subject to mobility, stations and access points permanently monitor several parameters to be aware of a change of the environment. This poses some challenges which are explained above.

Controlled Deferral of Clients
In [4], we proposed to use of the Point Coordination Function (PCF) by the AP to force the associated STAs defer for a specific amount of time. The Point Coordination Function (PCF) is an optional access method which enables a contention-free transmission. It is built on top of the DCF, and is used only in infrastructure networks. The AP acts as a master called the *Point Coordinator* (PC) and the STAs as slaves. The PCF is used by the AP to start a *Contention Free Period* (CFP), in which all clients, even those not associated with the BSSID, but residing in the same channel, defer from the channel, in order to be polled by the AP. The CF period shall alternate with a Contention Period (CP), in which the DCF is working. The mechanism works by introducing a specific information element (IE) called CF Parameter Set into the Beacons. This IE contains information like: duration of the CFP (*CFPMaxDuration*), in which periods it is started (*CFPPeriod*) and the remaining time, once it is started (*CFPDurRemaining*). This information is used by the STAs to defer from the channel.

Only very few off-the-shelf access point devices implement PCF. Nevertheless, the STAs have to understand at least the CFP start and stop mechanism, in order to coexists with potential PCF-enabled BSS.

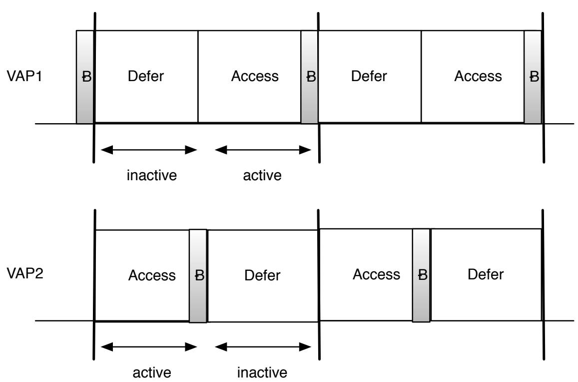

Figure 1. Controlled deferral of clients using the CFP mechanisms

We propose to use the information element described above to generate a CFP, and to make the STAs to defer from the channel. Figure 1 depicts our approach. The duration of the announced contention-free period can be used to create a second BSS event on another channel. The only required elements are the CF Parameter Set element in the Beacons to the start the CFP and a CF-End Frame to stop the CFP. The following example illustrates the mechanism:

A node serves two BSS through two virtual access points **VAP1** and **VAP2**. Both VAPs are alternating. When VAP1 starts its CFP to make the STAs defer from the channel, the interface is put on hold for the duration of a *CFPMaxDuration*. This time interval is the inactivity period of VAP1 but at the same time it is the activity period of VAP2. Announced through the CF Parameter Set Information Element in the Beacon, the duration of both periods can be controlled accurately.

V. Performance Evaluation

We performed some experiments to evaluate the performance of the PCF based approach, as described above.

A. Experiment Setup

a) Access Point: An IBM T42p Laptop equipped with an Intel M processor 2 GHz and an internal Atheros Communications AR922X 802.11abgn NIC running Ubuntu 12.04 and Kernel 3.8.10.

b) Station: A Dell Precision T3600 workstation (4 x 3.0 GHz Intel Xeon) with Ubuntu 12.04 and Kernel 3.8.10. We used a D-Link DWA-160 USB WiFi Card.

Performance measurements were performed on channel 48 (5 GHz), where no other AP were operating. We measured the TCP throughput, by running the iperf[2] tool for 100 seconds, respectively, between AP (server) and STA (client).

B. Measurement Scenarios

We start our investigation with a common use of an access point as reference scenario. The access point provides one network (one BSSID, one SSID). An STA is associated to the AP and transfers packets constantly.

Two scenarios will be discussed here to show the effect of coordinated deferral on the throughput of the stations.

The first scenario does not use the PCF. By using the off-channel command of the cfg80211 API, the wireless interface is sent to another channel for a specific time. This illustrates the operation of an virtual access point interface with channel switching constraints.

In the second scenario, the AP uses the proposed mechanism, to make the associated STAs defer from the channel.

In both scenarios, we performed two measurements. First the activity/inactivity period was set to 50 ms. In the second measurement, we increased the time to 100ms.

Table I
Throughput of virtual access point interfaces

	Active (ms)	Inactive (ms)	Throughput (Mbit/s)
reference (A)	100	off	21.9
off-channel (B)	50	50	n/a
off-channel (C)	100	100	0,62
PCF-based (D)	50	50	12.6
PCF-based (E)	100	100	14.0

The results (see table) show that PCF-based VAP outperforms uncoordinated VAP by far. In the first scenario, an activity of 50ms did not even work. The STA deleted the association shortly after association. The reason was too many missed beacons. This led to repeated association, authentication cycles. The network layer did not have the time to issue

[2]http://iperf.sourceforge.net

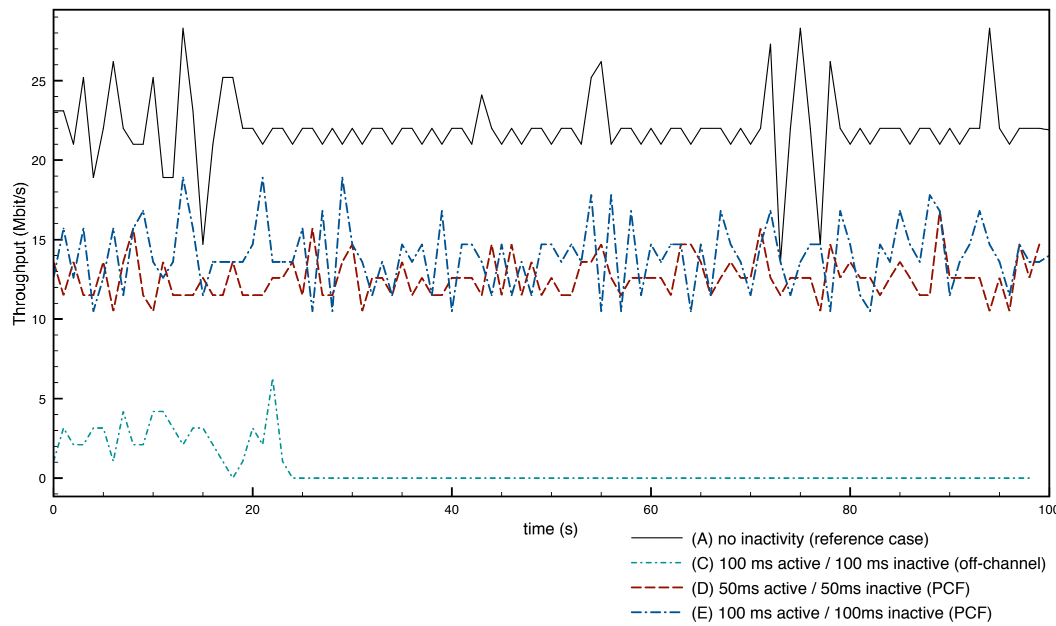

Figure 2. Deferral based on PCF

DHCP request. As soon as the association was created, it was removed immediately.

Increasing the activity and inactivity period solved the problem for a while, but the uncoordinated absence of the AP led to massive retries on the MAC layer and to TCP retransmission accordingly. The figure 2C shows that after 25 seconds the throughput dropped to zero, without recovering afterwards.

With PCF-based deferral of STAs, the throughput drops by approximately 40-50% (figure 2D,E). This is as expected, since the transmission gains only 50% of available channel capacity compared to our reference scenario (figure 2A). Most notably is the fact that TCP retransmission do not occur, because of the controlled absence of the AP.

VI. Conclusion

To support virtual wireless interface with different channel configurations on a single wireless card, different challenges on the operation of the MAC layer have to be taken into account. The fact that only one virtual interface can be active at the same time requires to prepare the communication partners in the wireless cell for the upcoming inactivity. In this work, we presented how parts of the PCF can be used to support virtualization of access points to allow wireless interfaces to serve more than one BSS, even on different channels. Through experimental evaluation of our proposed mechanism, we showed how a coordinated deferral of the STAs from the channel for a specific time makes virtual AP interfaces feasible.

For further research, more scenarios, where the combination of different virtual interface setups, like STAs and APs, will be studied. In the future, we will investigate which effects PCF-based deferral has on overlapping BSSs, as well as switching delays being tied with virtual interfaces have on the communication aspects. We plan to come up with an opportunistic MAC scheme combining multiple access principles to allow multi-channel relays. Therefore we will study different combinations of activity and inactivity cycles, even try to figure out how to adapt the time intervals during runtime according to characteristics of the wireless cell (number of STAs, performance, QoS, etc.)

References

[1] "IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, 12 2007.

[2] A. Raniwala and T.-c. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 2223–2234 vol. 3, March 2005.

[3] V. Bahl, R. Chandra, P. Lee, V. Misra, J. Padhye, D. Rubenstein, and Y. Yu, "Opportunistic Use of Client Repeaters to Improve Performance of WLANs."

[4] H. Coskun, I. Schieferdecker, and Y. Al-Hazmi, "Virtual WLAN: Going beyond Virtual Access Points," *Electronic Communications of the EASST*, vol. 17, 2009.

[5] Microsoft Research, "Virtual Wifi," URL: http://research.microsoft.com/en-us/um/redmond/projects/virtualwifi/, 2005.

[6] A. J. Nicholson, S. Wolchok, and B. D. Noble, "Juggler: Virtual Networks for Fun and Profit," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, 2010.

[7] D. Giustiniano, E. Goma, A. Lopez, and P. Rodriguez, "WiSwitcher: an efficient client for managing multiple APs," in *Proceedings of the 2nd ACM SIGCOMM workshop on Programmable routers for extensible services of tomorrow*. ACM, 2009, pp. 43–48.

[8] R. Chandra and P. Bahl, "MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, 2004.