Cross-Technology Wireless Experimentation: improving 802.11 and 802.15.4e coexistence

Peter Ruckebusch, Jan Bauwens, Bart Jooris, Spilios Giannoulis, Eli De Poorter, Ingrid Moerman
Department of Information Technology
IBCN - Ghent University – iMinds, Ghent, Belgium
Peter.ruckebusch@intec.ugent.be

Domenico Garlisi, Pierluigi Gallo, Ilenia Tinnirello
DEIM Department of Energy, Information Engineering and Mathematical Models
CNIT/University of Palermo, Italy
domenico.garlisi@cnit.it

***Abstract*—In this demo we demonstrate the functionalities of a novel experimentation framework, called WiSHFUL, that facilitates the prototyping and experimental validation of innovative solutions for heterogeneous wireless networks, including cross-technology coordination mechanisms. The framework supports a clean separation between the definition of the *logic* for optimizing the behavior of wireless devices and the definition of the device *capabilities*, by means of a unifying platform-independent control interface and programming model. The use of the framework is demonstrated through two representative use cases, where medium access is coordinated between IEEE 802.11 and IEEE 802.15.4 networks.**

Keywords—unified interfaces; control; wireless; API; cross technology communication (key words)

I. Introduction

Experimental validation of novel wireless solutions is becoming increasingly important in the emerging scenario of Internet-of-things (IoT) applications, due to the difficulty of modelling or simulating the interactions among a multitude of coexisting devices based on heterogeneous technologies (like IEEE 802.11, IEEE 802.15.4, Bluetooth low energy, etc.). However, setting-up experiments across heterogeneous technologies and hardware platforms requires a steep learning curve for experimenters, because they need to get accustomed to the overall software and hardware framework of the specific device and/or technology before prototyping their solutions.

The EU H2020 WiSHFUL project aims to lower the threshold for experimentation on heterogeneous wireless platforms, drive innovation and minimize time and resource investments for experimental validation, by providing a common programming model, for many popular experimentation platforms, such as SDR radios, IEE 802.15.4 based radios and IEEE 802.11 radios. Indeed, the WiSHFUL unified programming interface (UPI) allows experimenters to build platform-agnostic control programs that monitor and configure each wireless device in a uniform way. For example, the interface allows the tuning of the transmission channel, the selection of a modulation and coding scheme, the enabling of antenna diversity, and even the configuration of the parameters of the medium access protocol (e.g. the slotframe size of a TDMA protocol or the contention window of a CSMA protocol), among a set of available choices, without requiring a deep understanding of the device architecture.

To demonstrate the abilities of the WiSHFUL framework, especially in setting-up advanced experiments mixing different technologies, we show how two different coordination strategies between coexisting IEEE-802.11 and IEEE-802.15.4e networks can be easily implemented on top of the WiSHFUL UPIs. The first showcase is based on a common state-of-the-art solution, where interference is avoided by blacklisting IEEE-802.15.4 channels that overlap with highly loaded IEEE-802.11 channels. The second showcase exhibits how the WiSHFUL framework allows creating a beyond-state-of-the-art solution that mitigates cross-technology interference, by enabling a cross-technology synchronization scheme and TDMA access mechanism.

II. Background

The need for fine-grained control of communication networks is becoming increasingly important. This is well demonstrated by the interest of the scientific community in solutions that enable *software defined networking*, (SDN). *OpenFlow* [1], for instance, is a good example of an SDN-enabler because it allows researchers to control routers, without knowing the internals of vendor-specific implementations. OpenFlow, however, focuses on controlling the forwarding rules between devices (switches, routers and wireless access points) connected by means of pre-installed links (usually wired). The WiSHFUL control framework is complementary to OpenFlow, because it enables software defined networking in the wireless access area, where the concept of link is time varying and strongly dependent on interference and propagation conditions. For this reason, WiSHFUL focuses on the MAC and PHY layers, which strongly affect the link performance and availability, by taking into account the emerging solutions and standardization work concerning *reconfigurable radio systems* (ETSI-RRS) [2]. Within the WiSHFUL framework, three of such reconfigurable systems are currently supported: *Wireless MAC Processor* (WMP) for IEEE-802.11 radios [1], *Time-Annotated Instruction Set Computer* (TAISC) for IEEE-802.15.4 radios [4], and the *Implementing Radio in Software* (IRIS) for software defined radios (SDR) 0. The co-existence solution, demonstrated in this paper uses WMP and TAISC to coordinate medium access between IEEE-802.15.4 and IEEE-802.11 networks.

III. WiSHFUL Control Framework

Figure 1 provides a conceptual overview of the WiSHFUL framework. The *control programs* (CP) are user-defined algorithms that use the UPIs (Unified Programming Interfaces) to monitor and configure a single or a group of devices. The *monitoring and configuration engine* (MCE) provides an implementation for the UPIs on each platform. It also facilitates remote UPI usage (brown arrows) and offers necessary support services such as time synchronization.

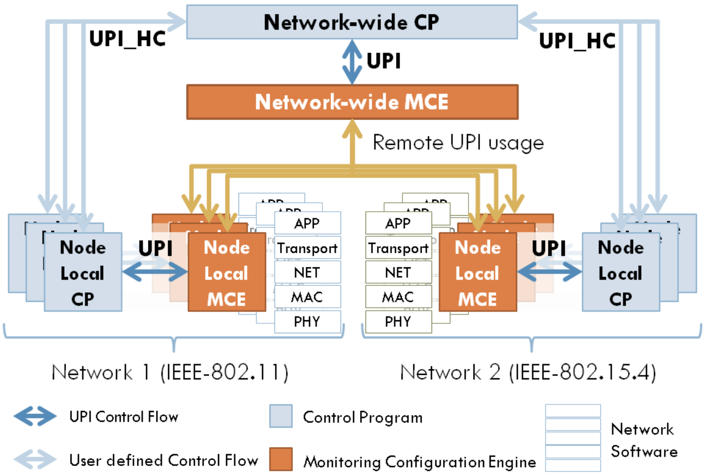

Figure 1. Conceptual overview of WiSHFUL control framework.

The framework operates both on the node-local and network-wide level. *Node local control programs* use the UPIs directly for node-local monitoring and configuration. They hence control a single device. *Network-wide control programs* use the UPIs indirectly for monitoring and configuring a group of nodes. The *network-wide MCE* enables remote UPI usage by dispatching UPI calls to the required *node-local MCEs*. The framework also allows to set-up additional, user-defined control flows (UPI_HC) between control programs.

IV. Cross-Technology Interference Mitigation

It is well-known that the simultaneous operation of IEEE-802.11(Wi-Fi) and IEEE-802.15.4e (TSCH) networks in close proximity will inevitably lead to performance degradation due to interference. The WiSHFUL framework, presented above, is used to implement two cross-technology interference mitigating algorithms that coordinate medium access between both technologies.

There are many possible co-existence schemes for Wi-Fi and TSCH. Some basic schemes can be implemented by only modifying the TSCH network. More advanced, and also more promising, schemes require cooperation between the networks. Figure 2 illustrates the schemes used in the demonstrated interference mitigation solutions for Wi-Fi (green/red) and TSCH (grey). The first basic strategy (A) uses the TSCH blacklisting features to exclude the spectrum used by the Wi-Fi network from being used in the hopping scheme applied by TSCH. The second, more advanced strategy (B) exploits the beyond-state-of-the-art runtime configuration features of the TAISC and WMP platforms to implement a cross technology TDMA schedule for a TSCH and a Wi-Fi network respectively. Both networks are synchronized using a specialized inter-technology synchronisation beacon send out by the Wi-Fi Access Point and detected by all Wi-Fi and TSCH nodes. The TSCH nodes can detect the Wi-Fi beacon using CCA (Clear Channel Assessment) because the beacon has a specific on-off pattern.

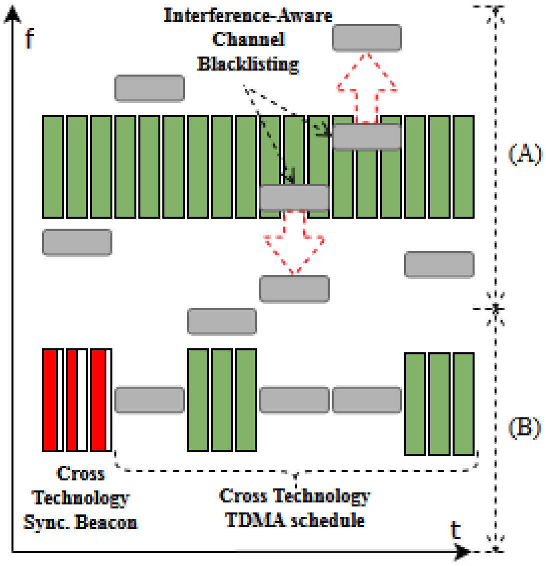

Figure 2. A basic (A) and advanced (B) cross-technology interference mitigation solution implemented using UPIs.

V. Implementation Details and Demonstration Set-up

The WiSHFUL control framework is provided as an open source solution and is fully supported on the following federated testbeds (iMinds w.iLab.t, TUB TWIST, TCD IRIS, Orbit). The core of the framework is currently implemented in Python and runs on both Linux and Windows. Table 1 lists the communication technologies that are currently supported and summarizes, for each technology, the available operating systems, hardware platforms and drivers.

Technology	**Supported platforms, operating systems and drivers**		
	Operating System	***Platform***	***Driver***
IEEE-802.11	Linux, Windows	Atheros, Broadcom	Ath9k, NDIS driver, WMP
IEEE-802.15.4	Contiki, TinyOS	MSP430, CC2x20, CC283x	Contiki/TinyOS drivers, TAISC
SDR	Linux, Windows	USRP, Xylink ZebBoard	Iris, LabView, GNU radio

Table 1 Supported platforms, OSs and drivers.

The demonstration set-up presented in this paper is deployed in the iMinds w.iLab.t testbed and comprises of 32 Contiki sensor nodes with an IEEE-802.15.4 radio and 14 Linux nodes with two IEEE-802.11 radios. Both showcases are executed simultaneously and demonstrated remotely. During execution, live measurements are shown in two formats: 1) live graphs displaying performance statistics and b) real-time spectrum scanning plots using a USRP device. Attendees can interact at run-time by changing configuration parameters of the example showcase control programs. For this purpose a simple UI is available. The following configuration options for both showcases are possible:

A. Configuration options for the basic showcase

The attendees can configure the Wi-Fi channel (2.4 GHz ISM band) and select the bandwidth (20/40 MHz) used for sending Wi-Fi frames. To mitigate interference, attendees can choose the TSCH channels that must be blacklisted. It is also possible to add an extra external Wi-Fi interference stream on a different channel and investigate the impact of uncontrolled cross technology interference.

B. Configuration options for the advanced showcase

The attendees can dynamically change the cross-technology TDMA schedule, e.g. allocation of slots between Wi-Fi and TSCH networks. Moreover, they can also specify a different synchronization pattern on-the-fly and add multiple concurrent streams in the TSCH network.

VI. Results

An example of the live performance statistics monitored during execution of the first, basic showcase is given in Figure 3. The graph shows the overall average network throughput measured over time. From the results, it can be clearly seen that there is a substantial loss of throughput when there is Wi-Fi interference. After blacklisting the affected TSCH channels, the throughput normalizes again. By changing the configuration parameters described in Section V.A, the attendees will see an immediate impact on the performance. Note that other statistics such as packet loss, jitter, TX throughput can be shown as well.

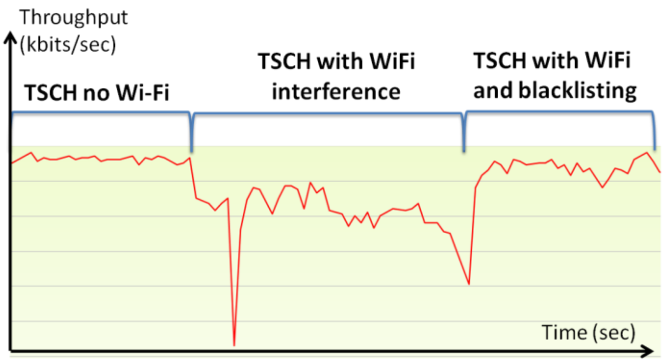

Figure 3. Live performance statistics showing the average network throughput (kbits/sec) over time.

While executing the more advanced showcase it is also possible to monitor performance statistics in combination with real-time spectrum scanning using USRP devices. Figure 4 illustrates the cross-technology synchronization beacon and TDMA schedule in real-time using an energy detection plot (y-axes is RSSI in dBm). When configuring this showcase, attendees will have an immediate feedback on the USRP plot.

The results from both showcases demonstrate the effectiveness of cross-technology interference mitigation and the ability to quickly set-up, investigate and fine-tune an interference scenario using the WiSHFUL control framework.

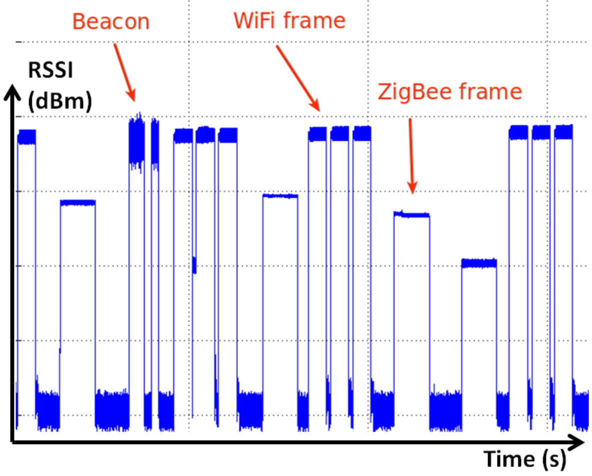

Figure 4. Live capture of RSSI (dBm) measured by the USRP over time.

VII. Conclusions

Representative showcases have been implemented and presented that clearly exhibit the abilities to create and evaluate cross-technology interference mitigation techniques on top of the unified programming interfaces (UPIs) provided by the WiSHFUL framework. Using the UPIs, experimenters can hence easily create techniques that allow minimizing the problems resulting from coexistence of IEEE 802.15.4 and IEEE 802.11 networks in close vicinity. The presented demos will give the ability to attendees to experience these problems and tweak specific control knobs of the presented solutions in order to solve them in real-time. This will give a better understanding of the WiSHFUL framework and the presented solutions.

Acknowledgment

This work was supported by the European Commission Horizon 2020 Programme under grant agreement n645274 (WiSHFUL) and the "SAMURAI: Software Architecture and Modules for Unified RAdIo control'" project.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. 2008. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun*

[2] ETSI TR 102 682", Reconfigurable Radio Systems (RRS); Functional Architecture for Management and Control of Reconfigurable Radio Systems, 2009

[3] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, "Wireless MAC Processors: Programming MAC Protocols on Commodity Hardware" IEEE INFOCOM, March 2012.

[4] B. Jooris; E. De Poorter; P. Ruckebusch; P. De Valck; C. Van Praet; I. Moerman; TAISC: a cross-platform MAC protocol compiler and execution engine; Under submission for Computer Networks.

[5] Sutton, Paul, et al. "Iris: an architecture for cognitive radio networking testbeds." Communications Magazine, IEEE (2010)