Demo: Towards a MAC Protocol App Store

Jan Bauwens, Bart Jooris, Eli De Poorter, Peter Ruckebusch and Ingrid Moerman
Ghent University - iMinds,
Department of Information Technology,
Internet Based Communication Networks and Services (IBCN)
Gaston Crommenlaan 8 - bus 201, B-9050 Gent, Belgium

1 Introduction

In recent years there has been a growing interest in the Internet-of-Things (IoT), leading to an expanding number of wireless environment and application domains in which IoT deployments are realized. Over the course of its lifetime, an IoT device will be exposed to different environments and varying demands in terms of network throughput, latency, power consumption, etc. For instance, consider the case of transportation and logistics where sensors will be attached to goods, pallets and containers, monitoring their respective state. These nodes need to work optimally every step of the supply chain: from frequent monitoring in densely packed warehouses to infrequent monitoring during overseas transit. The Medium Access Control (MAC) plays a major role in how a device handles the (changing) application requirements and environment. Unfortunately today there is no one-size-fits-all MAC protocol that works well in all cases. This causes an ever increasing number of MAC protocols that are each optimized for a certain application [1]. There have also been some hybrid or adaptive protocols developed to cope with different environments [2], but even they have limitations in reconfiguring the parameters or changing the behaviour of MAC protocols.

For the sensor devices in the example to work optimally over the entire trading lane, on-the-fly MAC reconfiguration or even replacement is required. This would also be beneficial for bug fixing or adapting to newer standards. Currently, on commercially available wireless radio chips, it is very hard to adapt or change MAC protocols on-the-fly after deployment. Moreover, also at development time it is cumbersome to port MAC protocols to new hardware platforms. This limits the reuse of the developed protocols between devices with a different radio installed.

We propose an architecture that enables us to (i) make on-the-fly changes to MAC protocol parameters, (ii) change the entire MAC protocol and (iii) upgrade the protocols over the air. This will enable users to better incorporate differing application needs. The end goal is to create an App Store where protocol developers can make their custom MAC protocols available to the public, as shown in Fig. 1. That way a user can select the protocol(s) that most suit his needs without any knowledge about the protocol implementation.

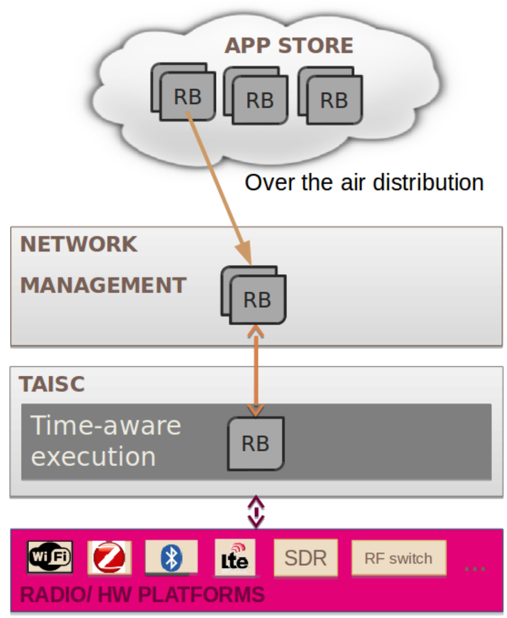

Figure 1: Concept of MAC protocol App Store as part of TAISC (RB = Radio Binary)

2 Framework

To achieve the goals set in Section 1, the Time Annotated Instruction Set Computer (TAISC) was introduced [3], a framework for hardware independent MAC protocol development and management. The framework gives us advantages in three key areas.

Firstly, TAISC introduces a cross-compilation phase allowing the developer to compile the same protocol code for different devices. This enables users to rapidly create their own custom MAC protocol that can be deployed on all their different IoT devices. The App Store can quickly be filled with a wide array of different MAC protocols. Secondly, the code efficiency for MAC protocols is a factor three smaller compared to their Contiki or TinyOS counterparts, albeit with a larger core (+- 10kB ROM). The core enables installing multiple MAC protocols on a single device. Due to the small protocol code size, it can be distributed over the air using only a few packets. Our version of CSMA/CA for example, with a size of 470 bytes, can be distributed with only four packets. Lastly, a MAC protocol and its parameters can be switched on the fly: the execution of the protocol stack doesn't need to be stopped. This ensures that the MAC layer is highly adaptable at runtime.

TAISC is an essential building block in the WiSHFUL architecture[1]. It enables to build unified programming interfaces (UPIs) for local and remote protocol configuration and monitoring according to the WiSHFUL vision[4]. WiSHFUL aims to incorporate cognitive adaptations of radio operation and automated run-time network intelligence, by means of flexible and unified radio and network control. The WiSHFUL UPIs can be used by local and global intelligent engines to gather node-local or network-wide information and to dynamically select the most optimal radio and network configurations. Over the air MAC replacement is provided the Generic extensions for Internet-of-Things ARchitectures (GITAR)[5].

Our vision of a MAC protocol App Store has partly been fulfilled. The on the fly adaptations of the MAC protocols is fully working, over the air replacement is still a work in progress. Future work will concentrate on finalizing GITAR, as well as supporting large scale networks. The number of supported platforms will be increased along with the development of a wide range of MAC protocols.

3 Demo: MAC protocol switching

To showcase the architecture, MAC protocol switching will be shown during the demo. The sensor nodes being used, running Contiki, have a series of protocols pre-installed. Currently the available MAC protocols are limited to CSMA/CA, TDMA and TSCH. These protocols each have some parameters that can be adapted: CSMA/CA has a changeable contention window, for TDMA the channel can be changed as well as changes to the slot layout and slot size and for TSCH the hopping sequence can be altered. The demo will start with turning on 16 nodes that are fixed on a board. Each of them will be sending packets with a size of 128 bytes at a fixed data rate. The initial global throughput will be quite low, but later on in the experiment the data rate will be enlarged via the user interface. The current setup can be seen in figure 2. The cables connected to the device are only present for monitoring purposes.

The attendees can make a choice of protocol that, in their opinion, is the best for the current situation. The initial protocol could be CSMA/CA, a basic contention-based access protocol. When the protocol has properly started, the behavior can be assessed: the attendees will be able to monitor network parameters (network throughput, packet loss, etc.) on a screen. The main question will be if this protocol is optimal for this situation. To simulate changing application requirements the general traffic flow will be increased until the global throughput stagnates. The protocol parameters will be changed in an attempt to better handle the traffic. At some point the attendees will decide that the current protocol might be less suitable. The nodes need to switch to a different protocol, to a time slotted protocol (TDMA) for example. The behavior of new protocol will be compared to the old one: have there been any noticeable improvements? Finally, to showcase the cross-compilation, it will be demonstrated that the same MAC protocols can be installed on a sensor node as well as on a Software Defined Radio (SDR) platform.

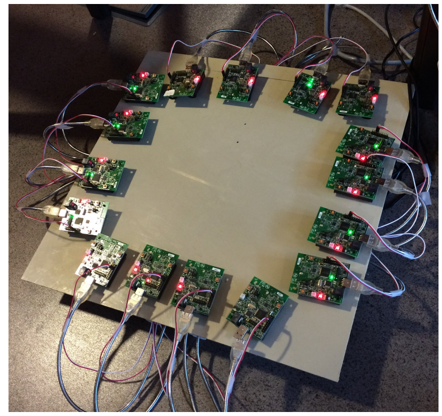

Figure 2: Example node set-up. Visual graphs are also shown were users can see the current performance and can decide to switch to a different MAC protocol at run-time.

4 Acknowledgments

This work was partially supported by project SAMURAI: Software Architecture and Modules for Unified RAdIo control, and European Commission Horizon 2020 Programme under grant agreement no. 645274 (WiSHFUL).

5 References

[1] Pei Huang, Li Xiao, Sima Soltani, Matt W Mutka, and Ning Xi. The evolution of mac protocols in wireless sensor networks: A survey. *Communications Surveys & Tutorials, IEEE*, 15(1):101–120, 2013.

[2] Mohammad Hossein Sedighi Gilani, Iman Sarrafi, and Maghsoud Abbaspour. An adaptive csma/tdma hybrid mac for energy and throughput improvement of wireless sensor networks. *Ad Hoc Networks*, 11(4):1297–1304, 2013.

[3] Bart Jooris, Eli De Poorter, Peter Ruckebusch, Peter De Valck, Christophe Van Praet, and Ingrid Moerman. Taisc: a cross-platform mac protocol compiler and execution engine. Computer Networks, 2015, under review.

[4] Carolina Fortuna, Peter Ruckebusch, Christophe Van Praet, Ingrid Moerman, Nicholas Kaminski, Luiz DaSilva, Ilenia Tinirello, Giuseppe Bianchi, Francesco Gringoli, Anatolij Zubow, et al. Wireless software and hardware platforms for flexible and unified radio and network control. *European Conference on Networks and Communications (EuCNC)*, June 2015.

[5] Peter Ruckebusch, Eli De Poorter, Carolina Fortuna, and Ingrid Moerman. Gitar: Generic extension for internet-of-things architectures enabling dynamic updates of network and application modules. *Ad Hoc Networks*, 36:127–151, 2016.

[1] `http://www.wishful-project.eu`