Demo: Runtime MAC Reconfiguration Using a Meta-compiler Assisted Toolchain

Xi Zhang, Junaid Ansari and Petri Mähönen
RWTH Aachen University, Institute for Networked Systems,
Kackertstrasse 9, D-52072 Aachen, Germany
{xzh, jan, pma}@inets.rwth-aachen.de

ABSTRACT

A rapid reconfiguration of medium access scheme is required in order to achieve runtime performance optimization for dynamic spectrum access and fulfilling varying Quality of Service (QoS) demands. We have developed TRUMP, a toolchain which allows composing MAC solutions at runtime [1] [2]. In this demonstration, we will show how MAC reconfiguration can be achieved efficiently using TRUMP. Inspired by the optimum route calculation method used in car navigation systems, the compiler toolchain in TRUMP realizes an appropriate MAC solution at runtime. TRUMP allows expressing various types of constraints and options such as speed, energy consumption and packet delivery rate which leads to different MAC compositions. The live demonstration of MAC reconfiguration will be carried out on WARP SDR platform [3].

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Design, Experimentation, Performance

Keywords

MAC, Reconfiguration, Compiler assisted, SDR platform

1. INTRODUCTION

With a growing number and complexity of wireless communication applications, new challenges have arisen in terms of providing the required stable and high QoS due to the ever crowding spectrum, changing application requirements and the need to coexist with other wireless applications. Since the spectral environment and the performance expectation can be highly dynamic, a runtime reconfigurable medium access scheme is needed in order to optimize the performance characteristics. For instance, a resource constrained device typically gives energy efficiency high priority while for video streaming applications, maximizing the data rate would become more important. Therefore, in order to achieve the desired performance characteristics, the application should be allowed to specify the preferences. This in turn will enable the medium access control protocol to reconfigure at runtime.

Classically, protocols are implemented in a monolithic fashion which restricts reconfigurability as required in a dynamic spectrum access paradigm. A common approach for reconfiguration is through appropriately setting the parameters of a particular protocol such as the selection of the modulation and coding schemes as in LTE [4]. MultiMAC [5] switches among a few pre-defined standalone protocols depending on the QoS requirements. Although these approaches aim at providing flexible and adaptable configurations at runtime, the design choice is limited to the subset of the tunable parameter values and the pre-selected options. Tinnirello *et al.* [6] have realized runtime composition of various IEEE 802.11 WLAN MAC behaviours on commodity hardware. However, this approach is limited to 802.11 group of MAC protocols implemented on WLAN cards. Our demonstration uses TRUMP [1] which allows composing a wide range of MAC protocols for various types of wireless networks at runtime by simply binding reusable MAC functional components [7].

In order to enable runtime protocol realization tailored to a wide range of user-specified application preferences, with the support of a MAC meta-compiler [2] we have adopted an approach similar to the car navigation system. The car navigation systems plan optimal route based on the position of a car in real time. The driver can specify a wide range of preferences and constraints such as minimum cost, minimum time, minimum distance, etc. The optimum solution is concluded based on the prior knowledge of the length of all the route, the toll costs and the traffic condition. It allows runtime update, route recalculation as the car position and driver preference changes. In our demonstration, a user can specify multiple criteria for MAC realization such as maximum energy efficiency, minimum latency and maximum data reliability through the meta-compiler. The meta-compiler requires pre-fed knowledge such as the execution time of each individual functional component, energy consumption at different protocol and hardware states, etc. Using this information, the meta-compiler selects the optimum combination of components and parameters.

2. SYSTEM DESIGN AND IMPLEMENTATION

Using a meta-compiler, TRUMP supports protocol realization based on user preferences both at design stage and during runtime as shown in Figure 1. Only two criteria are shown in the figure for simplicity. At the design stage, the meta-compiler can provide the MAC protocol designer with the predicted protocol performance in terms of execution speed and power consumption based on the prior knowledge of the MAC components without having to deploy the MAC code on the target platform. The compiler also selects MAC parameters based on designer specified preferences at the design stage. If there are multiple implementations for the same functionality available and the user does not specify a choice, the compiler selects the implementation which is optimized for user preferences.

SIGCOMM'12, August 13–17, 2012, Helsinki, Finland.
ACM 978-1-4503-1419-0/12/08.

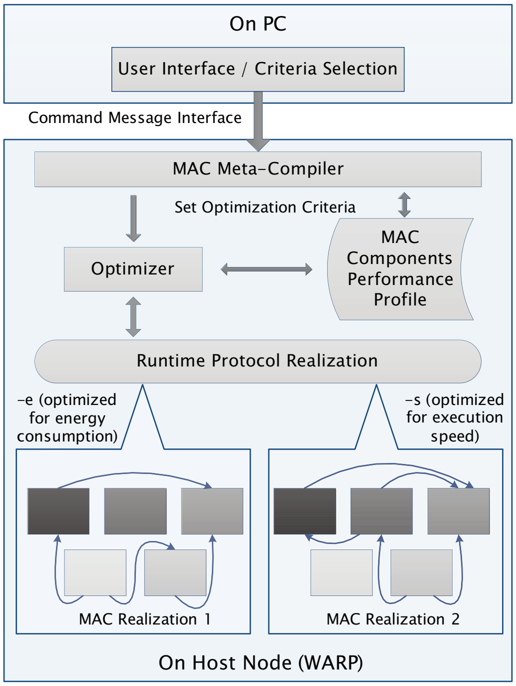

Figure 1: System architecture.

As a simple example, for sending a packet the user may specify the modulation scheme used on the packet payload. If the designer wishes to be able to send packets as fast as possible without having an insight on how each and every modulation option affects the MAC performance, one can set the optimization criteria for minimum execution speed and leave the modulation parameter for the `SendPacket()` interface to be `ANY`. The compiler queries the performance database for the `SendPacket()` component and replaces `ANY` with `QAM16` on our platform or any other modulation scheme which results in the highest data rate. If the user specifies maximum data reliability as the top priority, the compiler will return `BPSK` for the modulation parameter instead.

At runtime, an optimizer monitors the performance of the currently executing MAC. With feedbacks from the meta-compiler, it adapts the protocol according to the preference set by changing protocol functionalities and parameters. At the same time, it also monitors the spectral conditions by sampling received signal strength indicator periodically. For example, when a particular wireless channel is used by other networks, chances for the medium to be free are low. The optimizer will be aware of the increasing latency of delivering a packet due to the lack of opportunity to send packet in a free channel. If the application gives minimizing latency high priority, the protocol can deploy functionalities for multiple-channel support. It essentially provides a higher possibility within a time frame to locate a non-occupied medium for the packet to be delivered [8]. However, using multiple channels leads to additional control overhead and message exchanges which results in more energy consumption. Therefore, if the application prefers to be highly energy saving, the protocol can choose not to adopt the multichannel scheme. In conclusion, based on the selected criteria and the spectral condition, the MAC protocol adapts itself to the best suited behaviour by modifying composition of MAC components at runtime.

We have implemented our solution on WARP, a SDR platform with Virtex Pro II FPGA [3] (cf. Figure 2) based on the v16 OFDM reference design. The host compiler is developed using Lex&Yacc [9] and modified for our target platform.

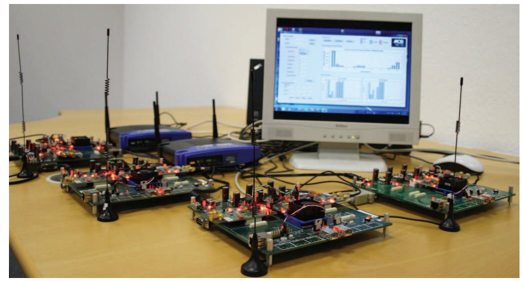

Figure 2: Demonstration setup.

3. DEMONSTRATION DESCRIPTION

This demonstration will provide opportunity to the audience to interactively modify a protocol realization criteria on-the-fly and correspondingly observe the triggered reconfiguration and the resulting MAC performances. The demonstration will also show an efficient runtime adaptation of the MAC protocol realization depending on to the channel condition. A controlled external interferer will be used to influence the spectral environment and thereby initiate auto MAC reconfiguration. In this demonstration, two WARP boards will be connected to a PC. An interactive Graphical User Interface (GUI) is used to display live MAC performance statistics. The demonstration will use two channels at 2.4 GHz ISM band.

Acknowledgment

We thank partial financial support from RWTH Aachen University, DFG through UMIC research centre (Nucleus project) and EU through 2PARMA project. We also thank Mr. Guangwei Yang for his technical support.

4. REFERENCES

[1] X. Zhang, J. Ansari, G. Yang and P. Mähönen. "TRUMP: Efficient Realization of Protocols for Cognitive Radio Networks" In *Proc. of IEEE DySPAN*, 2011.

[2] J. Ansari, X. Zhang and P. Mähönen. "A compiler assisted approach for component based reconfigurable MAC design" In *Proc. of Med-Hoc-Net (invited paper)*, 2011.

[3] A. Khattab, J. Camp, C. Hunter, P. Murphy, A. Sabharwal, and E. Knightly. "WARP: a flexible platform for clean-slate wireless medium access protocol design" In SIGMOBILE Mob. Comput. Commun. Rev, 2008.

[4] 3rd Generation Partnership Project (3GPP). The LTE Protocol Specification, 3GPP Rel8 In http://www.3gpp.org/Release-8.[Last visited: 10.06.2012]

[5] C. Coerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald. "MultiMAC - an adaptive MAC framework for dynamic radio networking" In *Proc. of IEEE DySPAN*,2005.

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

[7] J. Ansari, X. Zhang and P. Mähönen. "A Flexible MAC Development Framework for Cognitive Radio Systems" In *Proc. of IEEE WCNC*, 2011

[8] J. Ansari, X. Zhang and P. Mähönen. "A Decentralized MAC for Opportunistic Spectrum Access in Cognitive Wireless Networks," In Proceedings of the ACM SIGMOBILE Workshop on Cognitive Wireless Networking, 2010

[9] The Lex & Yacc Page In http://dinosaur.compilertools.net. [Last visited: 10.06.2012]