IN-DEVICE COEXISTENCE SIMULATION FOR SMARTPHONES

Sami Kiminki and Vesa Hirvisalo
Department of Computer Science and Engineering
Aalto University School of Science
P.O.Box 15400, FI-00076 AALTO, Finland
Email: {sami.kiminki,vesa.hirvisalo}@aalto.fi

KEYWORDS

wireless communications, in-device coexistence, smartphones, WiFi, LTE, protocol modeling, discrete event simulation

ABSTRACT

Wireless communication platforms in small enclosures are susceptible to in-device RF interference. Difficult interference cases emerge in the next-generation 4G smartphones using certain LTE bands, and support in MAC and higher protocol layers is required for efficient interference avoidance. Interference and the avoidance mechanisms create coupling between the radios complicating the interaction between the smartphone radios and their respective networks. Therefore, the purely analytical approach for studying radio performance becomes difficult.

In this paper we present RCOEX, an in-device coexistence simulator for the MAC layer. RCOEX simulates simple multi-network setups focusing on various in-device coexistence mechanisms and their performance. The simulation uses the discrete event variable time-delta model. As the protocol models are complex entities with non-trivial interaction, extensive effort has been made to support rich programming environment, including transceiver emulation, threads, and complex event filtering. Within this framework, we study the design of the WiFi model in more detail.

INTRODUCTION

Modern smartphones pack a number of radios in a small enclosure. Due to the current spectrum allocation, cellular (2G/3G/4G) and other radios (*e.g.*, WiFi, Bluetooth) may operate on close frequencies, which makes them susceptible to in-device RF interference. The related study item is known as in-device coexistence (IDC), in which various frequency-domain (FDM) and time-domain (TDM) solutions and mechanisms are studied. As the interference creates coupling between the affected radios, there must be some degree of cooperation between the radios for interference avoidance. (3GPP TR 36.816, 2012; Baghel et al., 2011)

The main interference scenario is the transmitter (TX) to receiver (RX) interference (Fig. 1). Due to non-linearities in RF processing, transmitters inflict noise around the frequencies of the transmitted signal as well as harmonics. Harmonics are easy to filter out but cost, size, and power issues limit reducing noise on the adjacent frequencies. As the TX signal may be more than 100 dB stronger than the RX signal, even small noise leakages may completely overshadow the reception. (Kiminki et al., 2011)

Arguably the most pressing IDC cases are with the LTE bands 7 (uplink 2500–2570 MHz), 40 (2300–2400 MHz), and 41 (2496–2690 MHz). LTE band 7 is one of the main bands allocated for 4G networks around the world, band 40 is common in Asia, and band 41 has been planned for North American region. The ISM band, in which WiFi and Bluetooth commonly operate, is between 2400–2500 MHz, the details varying per region. With unfortunate but not unrealistic network conditions, it may be that, *e.g.*, Bluetooth headset cannot be reliably used for LTE voice calls without specific IDC mechanisms, or the concurrent WiFi connection becomes flaky. (3GPP TR 36.816, 2012)

Our focus is on the IDC TDM mechanisms. Generally, the TDM mechanisms attempt to avoid problematic TX/RX operations by traffic shaping. Due to non-trivial interaction between the smartphone and the networks, and between the different radios in the smartphone, accurate results are difficult to obtain with purely analytical methods.

The nature of RF devices sets multiple requirements for the simulation of their behavior and control. In addi-

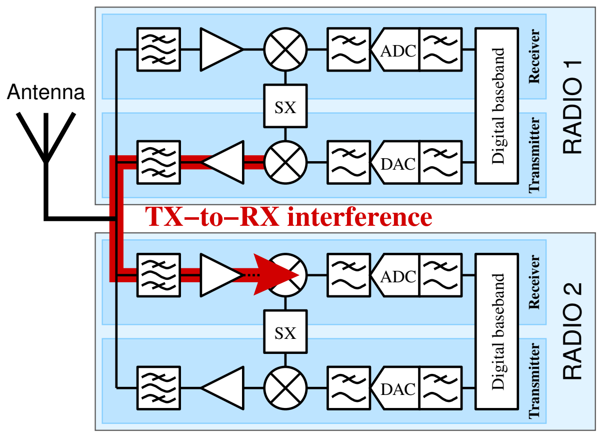

Figure 1: A common in-device interference case is that transmitter noise of one radio leaks to receiver of another

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

tion to being real-time systems controlling mixed signal electronics, the related radio systems and in-device radio implementations are complex. Therefore, flexibility for incorporating different aspects and programmability for tackling the complexity of communication protocols are essential in the simulation technology.

There are several simulators available for study of wireless communication, *e.g.*, LTESim (Piro et al., 2011) for LTE, Pamvotis for WiFi (Pamvotis, 2013), and ns-3 (ns-3, 2013) for networks using multiple access protocols. However, we are not aware of simulators implementing in-device coexistence mechanisms for LTE and WiFi with the related protocol extensions.

To study the efficiency and implementation feasibility of the IDC TDM mechanisms, we implemented RCOEX, a MAC-level radio coexistence simulator for LTE (3GPP TS 36.300, 2013) and WiFi (IEEE 802.11, 2012). With RCOEX, we can compare different proposed mechanisms and their effect on bitrates and retransmission rates. RCOEX models the communication between the user device and the network access point, in-device interference propagation, and the required protocol layers from MAC to IP level. To some extent, multi-user network simulations are also possible. At the MAC level and regarding transmission timing, RCOEX follows the standards in detail. We have published some of the research results produced with RCOEX (Kiminki and Hirvisalo, 2012).

Simulating the operation and understanding the properties of IDC TDM mechanisms calls for integrated methods. Hybrid simulators have been extensively studied and applied in practice. For embedded systems, integration to state-based simulators are very common (*e.g.*, Simulink/Statemate). For radios, simulation technology suitable for circuit design can be dominating for methodology, *e.g.*, (Nastov et al., 2007), but sometimes protocols form the dominating part, *e.g.*, (Piro et al., 2011).

Our contribution is in implementing a MAC-level radio coexistence simulator by integrating other mechanisms to a discrete-event variable time-delta simulator. The detailed modeling of the LTE and WiFi operation has been the dominating task for the work. We begin our description by explaining our problem model from the radio IDC point of view. The subsequent section describes our simulator core. The simulator basis is a full-fledged programming language (Java) embedded with an asynchronous simulator kernel by us. We also review how we implemented our WiFi simulation model that uses a bridging layer for incorporating synchronous behaviors. We also discuss our way of modeling related physics. To show the capabilities of the simulator, we explain its application to an LTE-WiFi coexistence scenario, and then, conclude our presentation.

PROBLEM MODEL

We simulate wireless communication of the user device in various IDC scenarios. The user device can be concurrently connected to multiple different networks. During an IDC scenario, there can be in-device interference, which prevents simultaneous operations of different radios such as transmission and reception. The simulation goal is not to study the overall network impact by devices with IDC issues, but instead, the goal is to study the IDC mechanisms in fine-grain manner from the user device perspective. The assumption is that if the mechanism is effective from the user device perspective (*i.e.*, spectrum-efficient and bitrate-efficient), the mechanism is effective also from the network perspective.

Simple network models are therefore sufficient. For LTE, this means modeling the communication between the user equipment (UE) and the base station (eNodeB). For WiFi, the model contains frame exchange between the station and the access point. Detailed temporal behavior models are crucial so that the interfered RX/TX operations affect the communication patterns and network scheduling correctly.

Wireless channels are unreliable by nature. This is modeled by dropping frames with a configurable probability. In-device interference of operations is modeled by projecting RX or TX operations of one radio as interference to RX or TX operations of another. In-device interference is assumed to be hard, *i.e.*, interfered operation always fails. Note that in realistic implementations, RX of more prioritized radio may prevent TX of less, *e.g.*, by forcing power-off of offending RF circuitry, which justifies the RX-to-TX in-device interference modeling.

Detailed temporal behavior at PHY and MAC-level is of importance so that the impact of interfered transmissions and receptions reflects accurately to the changed communication patterns. The TDM-based IDC mechanisms also operate at the PHY/MAC level. The load is IP packets, which allows simulating application-level bitrates.

The general setup in LTE/WiFi IDC scenarios is that the LTE has higher priority. This means that the offending operations of the WiFi radio are averted. For successful IDC, this means that the LTE radio must provide interference-free intervals for the WiFi radio and the WiFi radio must be able to adjust into them.

The IDC mechanisms in a smartphone require internal communication between radios, which is also modeled. For simple mechanisms, the WiFi radio senses the interference in the channel access procedure (CSMA/CA). For more complex mechanisms, the LTE protocol produces conservative predictions on its forthcoming RX and TX gaps. The predictions are maintained in a prediction vector with known RX and TX gaps.

The simulation runs forward in time without clairvoyance. The desired simulation results are usually: application level bitrate, retransmission rate, and power efficiency. This allows comparing different IDC mechanisms. Bitrates and power efficiency measure the impact on user experience. Retransmission rate is a measure of spectrum efficiency, which is important for networks.

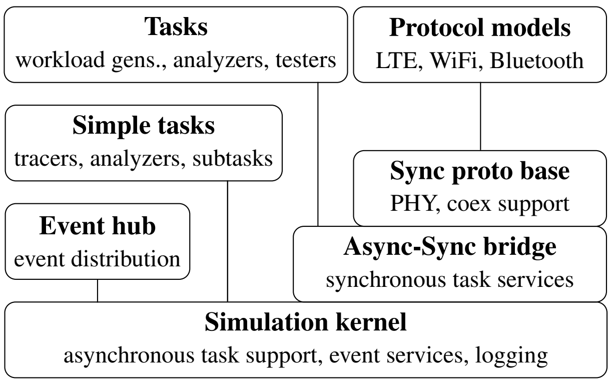

Figure 2: Architecture overview

COEXISTENCE SIMULATION

RCOEX is an event-based simulator with variable time deltas. The simulation consists of different simulation tasks such as protocols, load generators, and tracers. Many of them are further separated into subtasks. Most of the inter-task communication is by events with some of the communication additionally using shared objects such as queues. Simulation of physical transmissions and interference propagation is also implemented with events: start events signaling start of transmission or interference and end events the end of those. An overview of the simulator architecture is presented in Fig. 2.

The tasks are asynchronous at the low level and they are triggered by events and timers. To support synchronous programming model for complex protocol models (*i.e.*, to enable simulated sleeping and event waiting), an asynchronous-to-synchronous bridge layer is implemented. The tracer tasks analyze the event flow, producing bitrate and interfered frame rate measurements, for instance.

RCOEX simulates the timing-wise behavior of protocols in network setups. The ensemble of procedures related to different aspects of communication and protocols is fairly complex with many procedures containing intentionally stochastic behavior. The interaction between the user devices and the network base stations, in-device interaction between the user-device protocols, and interaction with workloads add more to the complexity of the big picture. This requires great flexibility in programming the simulation models and is the reason why considerable effort is paid to make the programming feasible.

The simulator and the simulation models are written in Java with the code base of approximately 20000 lines of code. The main rationale for using Java for the models (instead of dedicated simulation languages) was the existing libraries, debugging, and refactoring tools, which are essential for developing models with thousands of lines of non-trivial code. There is a performance-related trade-off in asynchronous-to-synchronous bridging performance, as synchronous tasks must be run in separate threads. This is because Java does not support continuations or co-operative multitasking, which means that a task that pre-empts cannot simply jump to the next task but has to switch threads. However, as a simulation run finishes usually in a matter of seconds, this trade-off between flexibility and performance is acceptable.

Simulation Kernel

The simulation kernel implements the time-delta simulation loop, event mediation, and provides a framework for implementing various simulation tasks.

The time-delta simulation operates on instances of class `TimeAdvanceAware`, which is the common base class for all tasks including the event hub. For each time-step, the simulation loop (Algorithm 1) polls each `TimeAdvanceAware` for their next wake-up times, possibly advances the simulation time, and invokes the wake up routines. The event hub ties the other tasks together. The tasks may send new events in a broadcast manner and receive events using various filters. The event hub buffers all new events and distributes them when the event hub task is run. The event filters are Boolean functions, which can match event types, event attributes, and they are composable for complex expressions. Custom event filters written in Java are also possible.

Algorithm 1 Simulation loop

```
while true do
    $T \leftarrow \min_{\theta \in \text{Tasks}}(\theta.\text{NEXTWAKEUP}())$
    if $T \geq T_{\text{end}}$ then
        return                                  ▷ End of simulation
    end if
    for all $\theta \in$ Tasks do               ▷ Inform tasks the new time
        $\theta.\text{SETTIME}(T)$
    end for
    for all $\theta \in$ Tasks do               ▷ Run tasks
        if $\theta.\text{NEXTWAKEUP}() = T$ then
            $\theta.\text{WAKEUP}()$
        end if
    end for
end while
```

The asynchronous task model is sufficient for simple tasks but becomes cumbersome for many complex tasks such as protocol models. Because of this, RCOEX simulator provides an asynchronous-to synchronous bridge. Synchronous tasks are executed in their own threads and can use higher-level constructs such as DELAY function for sleeping and event queuing mechanisms, *e.g.*, WAITFOREVENT. Essentially, synchronous tasks can be run in a loop.

For simulation purposes, there exists two general event categories shared by all protocol models:

PhyEvent Physical layer event, such as transmission start and end and interference events

L2Event Protocol layer 2 events such as workload frame transmissions and control info events

The PhyEvent category enables modeling the propagation of transmissions between transmitters and receivers of protocol models. The general PhyEvent and L2Event categories can then be traced by analyzer tasks which provide statistics such as bitrates and interfered frame rates. Protocol models use also custom event types for communication between internal tasks.

Protocol Models and Other Tasks

The protocol models are based on synchronous tasks. The base protocol class provides a virtual PHY for transmission and reception, transceiver constraints vectors for IDC purposes, and workload interface.

The virtual PHY communicates with virtual PHYs of other protocol models by PhyEvents. The communication can be beneficial, *e.g.*, exchanging frames between the user device and the network access point, or harmful in case of interference. If interference overlaps with an operation in the virtual PHY, the operation becomes interfered and cannot successfully transmit data.

Transceiver constraint vectors are exchanged between protocols. There are separate constraint vectors for the receiver and the transmitter. The constraint vectors are combined from transmit/receive prediction vectors of higher-priority protocol models in an IDC scenario. The constraint vectors tell when transmissions and receptions are safe. For example, in the LTE/WiFi IDC case where the LTE receiver is interfered by the WiFi transmitter, the WiFi transmitter may be forced to be powered off whenever the LTE receiver is on. The prediction vector for the LTE receiver is then used as the constraint vector for the WiFi transmitter, thus, allowing planned use of the transmitter for interference avoidance.

For complex layered protocols, the protocol models are usually modeled using subtasks. For example, in the LTE protocol model, the uplink and downlink are separate subtasks with shared state. In WiFi protocol model, the frame scheduler, MAC, transmitter and receiver control are separate subtasks. For network simulation, there are separate model variants for protocols running in base stations (LTE eNodeB, WiFi access points) and protocols in user devices (LTE UE, WiFi station).

The protocol models are driven by workloads. For example, a WiFi station can only send or receive data frames if there is data to be sent or there is data in the access point to be received. The workload is IP packets and is provided by workload generators by the workload interface. The mainly used workload generator injects n new IP packets periodically to a protocol model.

Inter-protocol Interaction

In user devices with multiple active radios, in-device signaling between protocols should be simple and generic for practical reasons. Our approach has been to organize radios in priority order, the less flexible and the more important radios with higher priority, and the more flexible with lower priority. The priority order is then used in deciding which radio gets access during conflict. (Kiminki and Hirvisalo, 2012)

In practice, the cellular radios (2G/3G/4G) have highest priority in our scheme. The cellular network organizes scheduling, which usually prevents the user device to autonomously schedule reception or transmission. This is because the cellular spectrum is expensive in a sense that unused slots are away from other users. It is also expensive because many operators have spent considerable amount of money for the licenses.

On the other hand, WiFi and Bluetooth operate on the free bands. By design, the radio access by user devices is flexible in WiFi networks. In power save mode, the only timing-wise fixed operation by user devices is the reception of beacons at regular intervals. The user device can freely decide when it starts to send frames or when it initiates delivery of received frames from access point buffers. In terms of flexibility, Bluetooth is somewhere between WiFi and LTE. Therefore, the WiFi radio is at the lowest priority and the Bluetooth radio is in the middle.

After the prioritization is done, the general idea is that the lower-priority radios adapt to interference-free gaps left by the higher-priority ones. To prevent starvation, the higher-priority radios should be throttled in order to provide the gaps.

Throttling and traffic shaping may be done at higher protocol layers simply by throttling workload. It may also be supported by the MAC layer. In LTE, MAC layer traffic shaping mechanisms include measurement gaps and the DRX mechanism (3GPP TS 36.321, 2012), but the DRX mechanism further requires either workload throttling or extensions.

In the RCOEX simulator, the general model for intermediating the interference-free gap information is prediction vectors. These vectors contain conservative prediction of RX and TX gaps. If a gap is indicated for certain time interval, it is a guaranteed gap. However, a non-gap indicates only uncertainty, which may later change to a gap. Usually as the simulation time advances and the closer the prediction becomes, the more accurate it is. Note that in LTE, accurate gap prediction for the uplink (TX) is possible approximately 2–3 ms in advance by using UL HARQ grant information. During measurement gaps and DRX off-duration, accurate predictions for both uplink and downlink are possible further in advance. In the RCOEX LTE model, the prediction vectors are updated every subframe (1 ms).

The produced prediction vectors can be used for action planning. The WiFi model uses the vectors to decide whether or not to initiate transmission or to trigger delivery of buffered frames. For some delivery mechanisms, the gap prediction information is conveyed to the access point. One such mechanism is the proposed CXA-Poll (Kiminki and Hirvisalo, 2012), which extends the standard U-APSD delivery (Takeuchi et al., 2006) with a delivery deadline. In the RCOEX WiFi model, the prediction vectors are consulted during the CSMA/CA channel

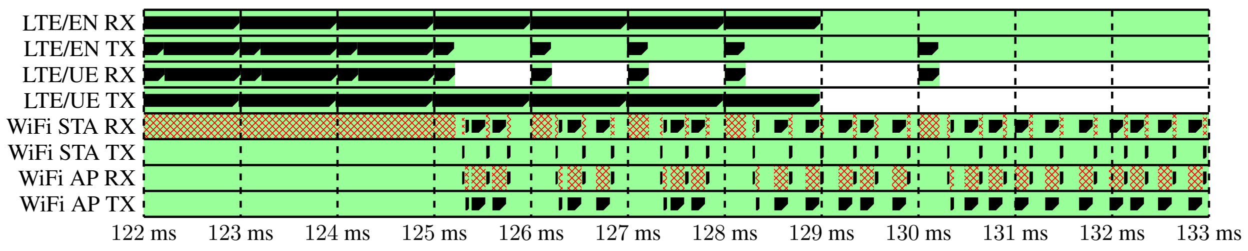

Figure 3: Visualized trace of LTE/WiFi IDC simulation where the use of LTE RX (UE) prevents simultaneous WiFi RX (STA). This setup is used to simulate the use of shared SDR receivers. Note that WiFi TX casts self-interference on WiFi RX. WiFi uses CXA-Poll delivery. Black bars denote transmission or reception, green areas powered-on state, and red crossing interference. UE and STA represent the user device. EN and AP are the LTE and WiFi network access points.

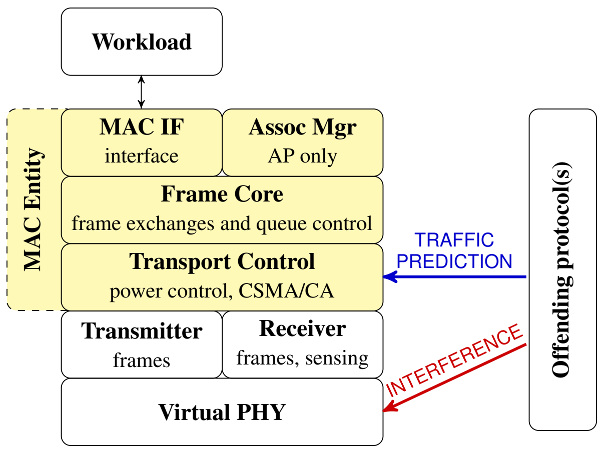

Figure 4: Overview of the WiFi model

access procedure when the CXA-Poll delivery is initiated. For more rudimentary inter-protocol IDC mechanisms, prediction vectors are not used. Instead, virtual sensing of interference may be used to extend the channel access procedure.

Preventing transmitter use because of higher-priority receiver use in RCOEX simulator is modeled by casting interference to the transmitter. This emulates cutting the transmitter power with the identical effect in communication patterns.

CASE: WIFI PROTOCOL MODEL

In this section, we examine the design and implementation of the WiFi simulation model in more detail. The WiFi model is layered with much of the same MAC-level functionality than what a real implementation would have. The WiFi model consists of virtual PHY, separate receiver and transmitter tasks, MAC entity task with multiple subcomponents, and an interface for workload generators. The overview is presented in Fig. 4. Technical details on the WiFi protocol can be found in, *e.g.*, (Gast, 2005), and authoritatively, in (IEEE 802.11, 2012).

The receiver task listens to PHY events from the transmitters of other protocol models. When transmission from an other WiFi node ends and the transmission does not contain interference, an event is sent to the MAC entity. The receiver also performs carrier sensing, which is used in the CSMA/CA procedure. The receiver task is asynchronous.

The MAC entity is also an asynchronous task reacting to events such as received frames and various timeouts (*e.g.*, no ACK frame received for sent data frame). The MAC entity commands transmissions and whether the virtual PHY should be powered or not. The MAC entity is further divided into subcomponents, such as Frame Core handling the lower level frame exchanges (*e.g.*, ACK handling and delivery mechanisms) with segmentation and concatenation of frames, Association Manager for managing associations in access point mode, and Transport Control for low-level frame queuing and CSMA/CA procedure. Transport Control is also responsible for transmission rate control, which uses the Minstrel algorithm in our case.

The transmitter is a synchronous task commanded by the MAC entity. It listens to the MAC entity command events and sends WiFi frames accordingly via the virtual PHY layer.

The main difference between a regular WiFi station and an access point is at the management level. For example, the basic network access for stations and access points uses the same CSMA/CA procedure. The PHY-level access capabilities of devices may be asymmetric, as wireless capabilities of stations are often biased to reception while the bias is on the better transmitter capabilities in the access points. The similarity allows the access point and station models share most of the code in the RCOEX simulator.

The low-level station-side IDC mechanisms are implemented in Transport Control. In WiFi, this means adaptation to interference-free gaps. The most effective mechanisms take advantage of the transceiver prediction vectors of the offending protocol. Frame transmission is postponed in CSMA/CA procedure if there is not enough interference-free time left for transmission and its corresponding acknowledgment. When triggering delivery of buffered frames by proposed CXA-Poll, a relative deadline for the delivery is added. For rudimentary best effort mechanisms, virtual sensing of interference is supported. In the access point and in case of CXA-Poll delivery, the deadline is used as an additional condition for triggering the end of delivery.

To simplify simulation setups, there is WLAN Configurator, a small start-up time task. It handles setting up the network configuration, and most importantly, the initial associations between the WiFi devices.

The simulation runs produce traces, which can be further processed with tools such as bitrate calculator, spectrum efficiency analyzer, and various correctness analyzers. An example trace is visualized in Fig. 3 using the LaTeX/TikZ-based trace visualizer.

CONCLUSION

We presented the design of RCOEX simulator, an in-device coexistence simulator for smartphones and other space-constrained devices with multiple concurrently active radios. RCOEX is an asynchronous event-based simulator for studying IDC TDM mechanisms and their efficiency. Considerable effort has been made to support the implementation of protocol models focusing on the MAC level. Radio protocols are complex entities with multiple layers and subsystems resulting in non-trivial behavior under different network and workload conditions. Additional complexity comes from inter-protocol interaction due to in-device interference and its avoidance. To exemplify, we discussed the design of the WiFi protocol model.

By using a full-fledged programming language embedded with a simulation kernel we have been able to ensure sufficient programmability for our simulator. This basis also gives us enough flexibility to integrate other behaviors than asynchronous ones to our simulation. Most importantly, we have integrated synchronous behaviors by using a bridging layer. The selected simulation approach is generic, thus, also other radios, *e.g.*, Bluetooth, can be simulated in this way.

The simulators for embedded systems have a long tradition towards hybrid simulators that target control systems. However, embedded systems are becoming more and more software intensive systems. Also, the importance of networking and communication is increasing. This calls for new approaches and methods for simulating such systems. We see this direction the most important considering future research on the area.

REFERENCES

3GPP TR 36.816 (2012). Study on signalling and procedure for interference avoidance for in-device coexistence (V11.2.0).

3GPP TS 36.300 (2013). Overall description, v9.10.0.

3GPP TS 36.321 (2012). Medium access control (MAC) protocol specification, v9.6.0.

Baghel, S. K., Ingale, M. A., and Goyal, G. (2011). Coexistence possibilities of LTE with ISM technologies and GNSS. In *National Conference on Communications*, pages 1–5.

Gast, M. S. (2005). *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2nd edition.

IEEE 802.11 (2012). Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.

Kiminki, S. and Hirvisalo, V. (2012). Coexistence-aware scheduling for LTE and WLAN during hard in-device interference. In *ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 1–6.

Kiminki, S., Saari, V., Pärssinen, A., Hirvisalo, V., Immonen, A., Ryynänen, J., and Zetterman, T. (2011). Design and performance trade-offs in parallelized RF SDR architecture. In *ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pages 156–160.

Nastov, O., Telichevesky, R., Kundert, K., and White, J. (2007). Fundamentals of fast simulation algorithms for RF circuits. *Proceedings of the IEEE*, 95(3):600–621. Invited paper.

ns-3 (2013). The `ns-3` network simulator. http://www.nsnam.org/ (Accessed Mar 25th 2013).

Pamvotis (2013). Pamvotis — IEEE 802.11 WLAN simulator. http://pamvotis.org/ (Accessed Mar 25th 2013).

Piro, G., Grieco, L. A., Boggia, G., Capozzi, F., and Camarda, P. (2011). Simulating LTE cellular systems: An open-source framework. *IEEE Transactions on Vehicular Technology*, 60(2):498–513.

Takeuchi, S., Sezaki, K., and Yasuda, Y. (2006). Quick data-retrieving for U-APSD in IEEE802.11e WLAN networks. In *IEEE Wireless Communications and Networking Conference*, pages 1421–1427.

AUTHOR BIOGRAPHIES

SAMI KIMINKI received the M.Sc. degree in computer science and engineering from the Helsinki University of Technology (TKK), Finland, in 2007. He is currently working towards the D.Sc. (Tech) degree in Department of Computer Science and Engineering at Aalto University, School of Science, Finland. His primary research interests include various real-time control issues for reconfigurable multi-radio platforms covering hardware resource scheduling algorithms and coexistence mechanisms.

VESA HIRVISALO is senior scientist at the Aalto University Department of Computer Science and Engineering. He received M.Sc., Lic.Sc., and D.Sc. degrees in computer science and engineering from the Helsinki University of Technology, in 1994, 1998, and 2004, respectively. During his career, he has worked on various aspects of embedded systems together with the industry. His expertise area is in compiler technology and simulation mechanisms. His research interests are focused on on parallelism and energy-efficiency in mobile systems.