International Journal on Advances in Telecommunications

2011 vol. 4 nr. 3&4

The *International Journal on Advances in Telecommunications* is published by IARIA.
ISSN: 1942-2601
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Telecommunications
Volume 4, Number 3 & 4, 2011

- Tulin Atmaca, IT/Telecom&Management SudParis, France
- Claus Bauer, Dolby Systems, USA
- Claude Chaudet, ENST, France
- Gerard Damm, Alcatel-Lucent, France
- Michael Grottke, Universitat Erlangen-Nurnberg, Germany
- Yuri Ivanov, Movidia Ltd. – Dublin, Ireland
- Ousmane Kone, UPPA - University of Bordeaux, France
- Wen-hsing Lai, National Kaohsiung First University of Science and Technology, Taiwan
- Pascal Lorenz, University of Haute Alsace, France
- Jan Lucenius, Helsinki University of Technology, Finland
- Dario Maggiorini, University of Milano, Italy
- Pubudu Pathirana, Deakin University, Australia
- Mei-Ling Shyu, University of Miami, USA

Communication Theory, QoS and Reliability

- Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
- Piotr Cholda, AGH University of Science and Technology - Krakow, Poland
- Michel Diaz, LAAS, France
- Ivan Gojmerac, Telecommunications Research Center Vienna (FTW), Austria
- Patrick Gratz, University of Luxembourg, Luxembourg
- Axel Kupper, Ludwig Maximilians University Munich, Germany
- Michael Menth, University of Wuerzburg, Germany
- Gianluca Reali, University of Perugia, Italy
- Joel Rodriques, University of Beira Interior, Portugal
- Zary Segall, University of Maryland, USA

Wireless and Mobile Communications

- Tommi Aihkisalo, VTT Technical Research Center of Finland - Oulu, Finland
- Zhiquan Bai, Shandong University - Jinan, P. R. China
- David Boyle, University of Limerick, Ireland
- Bezalel Gavish, Southern Methodist University - Dallas, USA
- Xiang Gui, Massey University-Palmerston North, New Zealand
- David Lozano, Telefonica Investigacion y Desarrollo (R&D), Spain
- D. Manivannan (Mani), University of Kentucky - Lexington, USA
- Himanshukumar Soni, G H Patel College of Engineering & Technology, India
- Radu Stoleru, Texas A&M University, USA
- Jose Villalon, University of Castilla La Mancha, Spain
- Natalija Vlajic, York University, Canada
- Xinbing Wang, Shanghai Jiaotong University, China
- Ossama Younis, Telcordia Technologies, USA

Systems and Network Communications

- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Anne-Marie Bosneag, Ericsson Ireland Research Centre, Ireland

- Huaqun Guo, Institute for Infocomm Research, A*STAR, Singapore
- Jong-Hyouk Lee, INRIA, France
- Elizabeth I. Leonard, Naval Research Laboratory – Washington DC, USA
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Reijo Savola, VTT, Finland

Multimedia

- Dumitru Dan Burdescu, University of Craiova, Romania
- Noel Crespi, Institut TELECOM SudParis-Evry, France
- Mislav Grgic, University of Zagreb, Croatia
- Christos Grecos, University of Central Lancashire, UK
- Atsushi Koike, Seikei University, Japan
- Polychronis Koutsakis, McMaster University, Canada
- Chung-Sheng Li, IBM Thomas J. Watson Research Center, USA
- Artur R. Lugmayr, Tampere University of Technology, Finland
- Parag S. Mogre, Technische Universitat Darmstadt, Germany
- Chong Wah Ngo, University of Hong Kong, Hong Kong
- Justin Zhan, Carnegie Mellon University, USA
- Yu Zheng, Microsoft Research Asia - Beijing, China

Space Communications

- Emmanuel Chaput, IRIT-CNRS, France
- Alban Duverdier, CNES (French Space Agency) Paris, France
- Istvan Frigyes, Budapest University of Technology and Economics, Hungary
- Michael Hadjitheodosiou ITT AES & University of Maryland, USA
- Mark A Johnson, The Aerospace Corporation, USA
- Massimiliano Laddomada, Texas A&M University-Texarkana, USA
- Haibin Liu, Aerospace Engineering Consultation Center-Beijing, China
- Elena-Simona Lohan, Tampere University of Technology, Finland
- Gerard Parr, University of Ulster-Coleraine, UK
- Cathryn Peoples, University of Ulster-Coleraine, UK
- Michael Sauer, Corning Incorporated/Corning R&D division, USA

Additional reviewers

- Vassilis Stylianakis, ECE, University of Patras, Greece

International Journal on Advances in Telecommunications
Volume 4, Numbers 3 & 4, 2011

CONTENTS

System Architecture for High-speed Close-proximity Low-power RF Memory Tags and Wireless Internet Access

Iiro Jantunen
Department of Applied Physics
University of Eastern Finland
Kuopio, Finland
iiro.jantunen@uef.fi

Joni Jantunen[a], Harald Kaaja[a], Sergey Boldyrev[b]
[a]Nokia Research Center
[b]Nokia Location & Commerce
Helsinki, Finland
{joni.jantunen, harald.kaaja, sergey.boldyrev}@nokia.com

Le Wang, Jyri Hämäläinen
Department of Communications and Networking
Aalto University
Espoo, Finland
{le.wang, jyri.hamalainen}@aalto.fi

Abstract — **We have developed an open architecture platform for implementing passive radio-frequency identification (RFID) tags with a mass memory for close proximity environment. Purposes for such mass memory tags are, e.g., multimedia files embedded in advertisements or logged sensor data on a low-power sensor node. In the proposed architecture, a mobile phone acts as the reader that can read or write the memory of these RFID tags. The architecture also enables creation of a new type of wireless internet access suitable for, e.g., internet kiosks. The architecture is designed so that development path to a full Network on Terminal Architecture (NoTA) is feasible. The wireless reading speed of the mass memory tags, demonstrated to be 112 Mbit/s, is in the range that a 3-minute 640×320-pixel video can be loaded from the tag to the phone in less than 10 s. Our solution supports Nokia's *Explore and Share* concept.**

Keywords — memory architecture; multimedia systems; RFID; telephone sets; RF memory tags; Internet connection

I. Introduction

Today's mobile phones contain music and video players, which make it possible for consumers to enjoy entertainment while on the move. Acquiring new multimedia content by downloading or streaming, however, is hampered by the high cost and slow speed of Internet connections, as well as by the fact that commonly used physical multimedia formats, such as optical disks, cannot be read with a mobile phone. Thus, to make acquiring new content easier, cheaper and less power-consuming, we propose a new technology based on radio frequency (RF) memory tags readable and writable by mobile phones [1].

RFID tags are increasingly a part of our life; transport, traceability, and secure access are some of the main uses of this close proximity technology today. Conventional machine-readable wireless tags, e.g., Near Field Communication (NFC) tags, normally have a very small memory in the range of hundreds of bytes or kilobytes. Some RFID standards include an option to have a flexible-use memory, but the capacity is low compared to factory-set fixed-content memory. Tag selection is based on reading the content in a selected tag memory address (e.g., tag or manufacturer ID). As the memory capacity of these tags is small, the amount of data to be transferred is also small and power consumption of RF communication is, thus, not a critical issue.

Various research groups have developed improvements to the commercially available RFID technologies. To overcome the storage capacity limitation of passive tags, Wu et al. increase effective tag storage sizes with proposed distributed RFID tag storage infrastructure (D-RFID stores) [2]. Tags would be distributed in space and time in this architecture. Ahmed et al. focus on RFID system unreliability and improvements in middleware for object tracking and object location with moving readers or tags [3]. As a result of their research, a virtual reader system architecture was introduced. Ying described a verification platform for RFID reader that utilized Ultra High Frequency (UHF) frequency [4]. This platform is applicable for customization with different RFID standards. Pillin et al. have developed a passive far-field RFID tag using the 2.45 GHz Industrial, Scientific and Medical (ISM) band, with a data rate of 4 Mbit/s on the range of 5.5 cm [5]. As an example of a proprietary solution, HP's *Memory Spot* tag also works on the 2.45 GHz ISM band and has demonstrated 4 MB memory and 10 Mbit/s data rate but only allowing a touch range [6].

The problems of low data reading rate and small memory size provided by contemporary RFID tags become emphasized if one considers mobile users reading multimedia files from tags embedded on, e.g., paper media. The attention span of a mobile user is about 10 seconds [7]. Within this period, the user could get a single multimedia content file from a memory tag. Considering a movie trailer, the file size for a 2-minute 640×320-pixel 30-fps (3 Mbit/s), encoded with H.264, would be around 50 MB [8]. The required minimum data transfer rate from the user point-of-view is thus 50 Mbit/s. This exceeds the maximum data rate available by the 13.56 MHz NFC technology, 848 kbit/s, by a factor of 60. Even the maximum data rate for NFC demonstrated on a laboratory set-up, 6.78 Mbit/s [9], is not enough. When users are getting used to NFC, the speed and storage capacity becomes quite easily a limiting factor. Thus, there is a need for a new high-speed touch-range RFID radio interface.

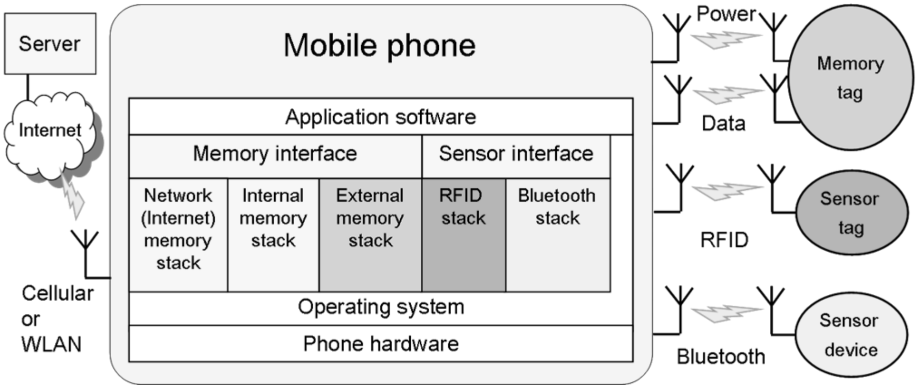


Figure 1. MINAmI Architecture [11]

The aim of our research has been to develop a high-capacity memory tag, which is wirelessly readable with a mobile phone and suitable for consumer markets in ubimedia applications [10][11]. The mobile phone acts as the user interface for reading and writing passive RF memory tags that contain a high-capacity memory (0.1–1 GB). The reasoning for the proposed technology was justified by modern trends in non-volatile memory technologies, according to which the power consumption, physical size, and price of memory are continuously decreasing. The technology presented in this paper supports Nokia's *Explore and Share* concept, a new way of transferring content (e.g., multimedia, maps, and applications) to a mobile phone [12].

Another use case for the technology presented in this paper is an Internet Kiosk, i.e., a short-range hotspot providing access to the Internet. Traditionally, an Internet kiosk is nearly always a computer connected to core network via a backhaul link that can be wired or wireless broadband. These Internet Kiosks are typically pay-as-you-go (credit card or pre-paid) or funded by advertisements presented within the kiosk screen.

Nowadays, accessing the Internet freely and securely has become critical for business and recreational needs. Personal wireless terminals (such as mobile phones or laptop computers) are, thus, associated with modern Internet hotspots based on wireless local area network (Wi-Fi) technology. These Wi-Fi hotspots are either free or paid by credit card when logging in and are widely deployed. Another kind of wireless Internet hotspot is femtocell-based [14], providing a local connectivity through cellular technology such as wideband code division multiple access (WCDMA). Conventionally, femtocells are designed for use in a home or small business to improve indoor wireless coverage. As Bluetooth is a competitive wireless solution especially from energy saving point-of-view, it has also been considered as one technology to enable wireless kiosks [15]. Moreover, NFC featured kiosks allow users to be connected and download multimedia content via NFC-enabled mobile phones [16].

All the wireless internet kiosk or hotspot types mentioned above share the demand of internal power within the phone, the power usage of the connection depending on the technology and status of the hotspot (distance, amount of devices connected etc.). If draining the battery of the phone is not acceptable (e.g., while travelling), a wired connection to a power outlet is needed.

In this paper, we describe and specify a network architecture, which enables mobile phones to read and write passive RF memory tags and use a RF memory tag based Internet connection. The architecture has been developed and demonstrated in the EU's 6th Framework Programme (FP) "Micro-Nano integrated platform for transverse Ambient Intelligence applications" (MINAmI) project [17], and thus the architecture is referred to as MINAmI Architecture. Important architecture requirements include openness, modularity, scalability, and energy efficiency. Openness and modularity are needed to support creation of novel applications and services by 3rd parties. Scalability of data rate is needed to enable evolution of the technology along with evolution of multimedia services. Energy efficiency is essential to enable passive operation of the tags as well as to save the phone's battery.

The paper is organized as follows. In Section II, we introduce the system architecture, along with a key component of the architecture, RF memory tags. In Section III, we introduce a novel dual-band radio subsystem and its hardware and software implementation. In Section IV, we introduce a new type of power-saving short-range wireless internet access for mobile devices and compare its power use to other available wireless internet access technologies. In Section V, we present the current status of implementation of the architecture, discuss possibilities for future development, and draw some conclusions.

II. MINAmI Architecture

The proposed MINAmI architecture makes use of the mobile phone's capability of running software and providing several radio interfaces (Fig. 1). The architecture is modular, enabling simpler and faster development of new technical extensions (e.g., RF memory tags). Our architecture focuses on utilization of modularity on component level (e.g., where to plug memory tag functionality) and on communication level (e.g., how the available memory tags are utilized). At close proximity domain (range < 1 m), different tags are communicating locally with a mobile phone. In the present work we have concentrated on the RF memory tags. The sensor parts of the architecture (RFID sensor tags and Blue-

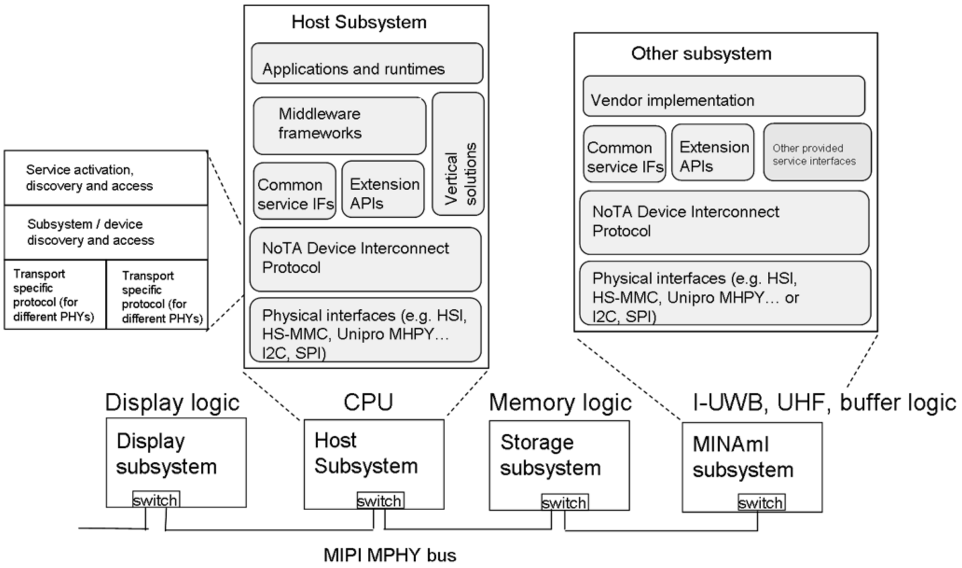

Service activation, discovery and access
Subsystem / device discovery and access
Transport specific protocol (for different PHYs)
Transport specific protocol (for different PHYs)
Host Subsystem
Applications and runtimes
Middleware frameworks
Vertical solutions
Common service IFs
Extension APIs
NoTA Device Interconnect Protocol
Physical interfaces (e.g. HSI, HS-MMC, Unipro MHPY… I2C, SPI)
Other subsystem
Vendor implementation
Common service IFs
Extension APIs
Other provided service interfaces
NoTA Device Interconnect Protocol
Physical interfaces (e.g. HSI, HS-MMC, Unipro MHPY… or I2C, SPI)
Display logic
CPU
Memory logic
I-UWB, UHF, buffer logic
Display subsystem
Host Subsystem
Storage subsystem
MINAmI subsystem
switch
switch
switch
switch
MIPI MPHY bus

Figure 2. NoTA extension architecture for MINAmI subsystem, where HSI = High Speed Serial Interface; HS-MMC = High Speed MultiMediaCard: SPI = Serial Peripheral Interface bus.

tooth sensor devices) have been studied in an earlier project [18][19].

The main RF memory tag architectural design challenges include target platform performance obstacles, such as available bus operations (read/write) and power requirements, especially when drawing the line for autonomous operations in the described MINAmI architecture. The other challenge is minimizing changes to the existing system communication layering, only to the external memory stack block. The technological choices in the MINAmI system architecture were able to support both existing standard radios for low-rate sensors, and the high-rate high-capacity memory tags.

A. Network-on-Terminal Architecture (NoTA)

NoTA is a modular service-based system architecture for mobile and embedded devices offering services and applications to each other [20]. The concept is being defined in an open initiative. NoTA is also known as an open device distributed architecture, which allows direct connections between different nodes, within subsystem or between subsystems, supporting several physical interfaces within the device or between devices [21]. This architecture also supports both messaging and streaming services. The beauty in the architecture resides in modularity and transport independency. Direct connection between subsystems improves the efficiency as they do not necessarily require any processor involvement, when subsystems have all the needed functionalities available for their independent operations. Transport-specific portion is hidden underneath NoTA communication layering.

NoTA communication layering is built around transport-independent parts and it provides interfaces towards transport-specific parts (Fig. 2). Extension architecture with Device Interconnect Protocol (DIP) enables flexible open-source architecture for different hardware platforms. DIP provides logical links between a requesting subsystem and other subsystems or within a subsystem [21]. DIP is a device-level communication protocol that can be implemented for various physical interfaces ranging from MIPI (Mobile Industry Processor Interface) high speed serial interfaces and Universal Serial Bus (USB) to wireless interfaces, such as Bluetooth [22][23]. Another example of utilization of DIP is the Open Modem Interface Protocol [24].

NoTA host subsystem and neighboring subsystems are connected via the high speed physical interface. DIP adapts physical interfaces to the upper layers. It is the lowest layer that is common for all subsystems (i.e., also for MINAmI subsystem) and hides the physical dependencies underneath. Above DIP there is a common service interface used for resource management, file systems, and system boot-ups. Middleware frameworks, e.g., for multimedia, USB, and other applications, use a common service interface or extension Application Programming Interface (API). The architecture also takes into account vertical solutions, which may require an optimized protocol design for certain requirements that are tied to HW-specific applications.

NoTA subsystem structure takes into account possibility to add different types of independent service or application subsystems to the architecture, and the MINAmI subsystem forms one high data rate high capacity subsystem. When properly designed, the modular NoTA-based subsystem specification involves clear distinction of the system designer/integrator and vendor views of subsystem description and scenarios. Based on a provided subsystem specification the vendors test and validate their subsystem implementation and deliver them to the product designers for integration [25].

The MINAmI subsystem offers memory tag read/write, storage and local connectivity services to other subsystems within mobile device, and its architecture is compatible with NoTA communication layering. The MINAmI subsystem includes both the mobile phone (Mobile Reader/Writer) and the tag and all the relating hardware and software resources. Mobile Reader/Writer sees the contents of the memory of a

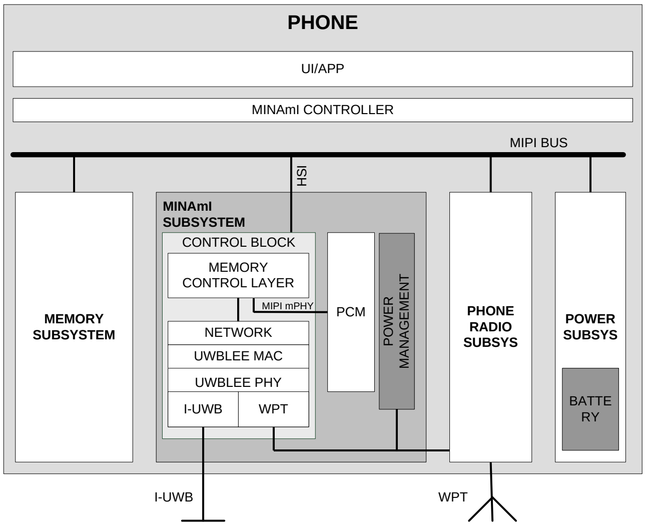


Figure 3. MINAmI Architecture on phone

passive RF memory tag only when there is an established connection, i.e., power field and data connection exists.

B. RF Memory Tags

The focus of our research has been on mobile-phone-operable memory tags suitable for consumer markets and ubimedia applications. The tag is developed as a part of our mobile-phone-centric architecture. Our memory tag development targets improving both transfer speed and storage capacity. These improvements give direct benefit for ubimedia users.

The target memory capacity of our memory tag has been in the range of gigabits and mobile reader/writer transfer speed to and from memory tag in excess of 10 Mbit/s. The same design platform is usable for both ends, for mobile phone platform reader/writer and for tag implementation. When designing the platform, various important design parameters, such as the selection of the used radio technology, were considered to provide an efficient and low-power solution for mobile reader/writer and tags.

It was important to make sure that connectivity technology is simple enough for the user, e.g., it should facilitate easy content selection (see Section III.D). Memory tag content selections should be based on metadata (e.g., filenames, file content types, file content keywords). Due to the large memory size, power consumption for memory access is a critical design issue, both for reading and writing the memory tag. To be successful on the market, RF memory tags for ubimedia must be passive to make them as small (size) and cheap as possible, and to achieve autonomous usage with minimum maintenance (e.g., usage without charging of battery). This severely limits the power budget. On the other hand, a short communication range (even touch) is sometimes preferable to make it easier for the user to physically select the tag. An RF memory tag will be read many times by different users, but written more rarely – in some cases, only once. The memory unit must work reliably even with several consecutive read cycles. A limited write throughput due to power constraint is not an issue, as data is rarely written by the users.

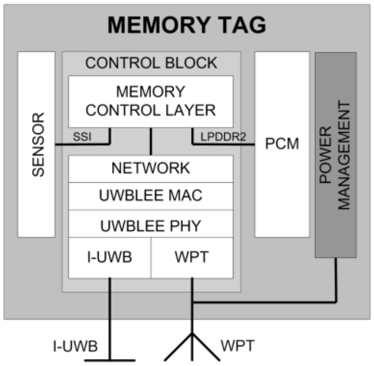


Figure 4. MINAmI Architecture on a RF memory tag

III. UWB LOW END EXTENSION

As memory tags have high data storage capacity, a high-speed radio is needed for communication to enable reading even all the contents of the tag in an acceptable time. Currently available mobile phones contain several radio transceivers, such as cellular, Bluetooth, and Wi-Fi, along with NFC. Most of the technologies are made for well-established communication between active devices, consuming a relatively large amount of power. These technologies are also not inherently designed for ad-hoc, possibly one-time, connections between devices that have not communicated with each other before, resulting in long latency in establishing the communications. For example, in an environment with many unknown Bluetooth devices, the Bluetooth connection setup latency can be over 10 seconds [26]. NFC enables communications between an active and a passive battery-less device and is physically more selective; its communication range is almost in touch. However, it has severe limitations in data transfer speed.

To provide higher data rates, a wider frequency band available on higher frequencies needs to be used. On the other hand, the efficiency of wireless power transfer (WPT) decreases as a function of center frequency. To solve the problem of providing high-speed communication (high frequency needed) while simultaneously providing power wirelessly to the tag, a dual-band radio interface has been proposed [27]. One narrowband signal on RFID frequencies (e.g., RFID frequency bands globally available between 860–960 MHz) is used to power the tag and to provide a mutual clock reference for both ends of the communication link, whereas the communication link itself is based on impulse Ultra-Wideband (UWB) technique to provide a high communication bandwidth and scalability for even higher data rates.

As the selected RFID frequencies are approximately in the same frequency range as Global System for Mobile Communications (GSM) or WCDMA 900 MHz, in the reader there is a possibility of integrating the WPT function to the existing Phone Radio Subsystem, as presented in Fig. 3. In that case, Phone Radio Subsystem is designed so that the WPT Physical (PHY) Layer function may request a direct access to control the activation of the narrowband transmitter. Especially, the time-domain interleaving of different functions is important to support co-existence of GSM/WCDMA and WPT signaling.

The architecture of the RF memory tag (Fig. 4) is similar to the MINAmI subsystem on the mobile phone. For simple RF memory tags, no network layer implementation is needed to take care of the point-to-point communication between the reader and the tag, and therefore is handled on Medium Access Control (MAC) layer.

As an option for use-cases like data-logging sensor devices, the memory control layer provides a sensor interface. During the sensing, the sensor data is stored to the Phase-Change Memory (PCM) block and the low data-rate data capturing is powered from a battery or with energy harvested from the environment. For fast downloading of the logged data, the reader powers the sensor tag wirelessly.

A. Hardware architecture

The hardware, both the radio front-end and the memory, of the RF memory tag needs to run on the energy scavenged from the UHF transmission of the mobile phone. This subsection describes the enabling technologies: low-power high-data-rate radio front-end and low-power high-capacity high-speed non-volatile memory.

1) Radio Front-end

As presented in [27], a very simple super-regenerative transceiver architecture can be used in impulse UWB communication to achieve required data-rates over short distances. In contrast to conventional impulse UWB transceivers [28], there is no need for multipath recovery over the distances below 30 cm. This decreases the requirements set for the UWB transceivers. This is used to minimize complexity and power consumption of the transceivers. In the aforementioned super-regenerative transceiver one super-regenerative oscillator is used alternately both to generate transmitted pulses and to amplify received pulses, and no linear amplifiers are needed. Thus, the architecture utilizes the inherently low duty cycle of the transmitted impulse UWB signal also in reception the receiver being fully active only exactly during the detection of incoming pulses.

Synchronization is often problematic in impulse UWB systems because of the low duty cycle and pseudo-random timing of pulsed signal, and due to frequency drift and differences of reference clocks between the transceivers. In the proposed system the frequency synchronization between the reader and tag is achieved thanks to the mutual narrowband WPT signal, which is also used as the reference clock. The phase synchronization of impulse UWB transceivers is also easier to achieve due to decreased need for pseudo-random time-coding of pulse patterns.

The transceiver structure supports simple On-Off-Keying (OOK) modulation. The data-rate and power consumption is also scalable depending on the power level available for the wirelessly powered tag. Due to the simplified transceiver structure, targeted ultra-low power consumption and partial exploitation (500 MHz) of full UWB band (3.1–10.6 GHz) authorized by Federal Communications Commission (FCC) for unlicensed use, the impulse UWB system referred here is called UWBLEE (UWB Low End Extension).

Altogether, the optimized transceiver architecture makes it possible to achieve required high data-rates with a low power consumption performance (a few mW) suitable for WPT. As a proof-of-concept, a complete wirelessly powered RF front-end implementation of the super-regenerative transceiver is presented in [29] and [30] by using a single super-regenerative oscillator for transmission and reception. The front-end implementation supports data-rates up to 112 Mbit/s with the energy consumption of 48 pJ/bit in reception and 58 pJ/bit in transmission. The feasibility of the ultra low power consumption in high data-rate two-way communication is verified with an integrated RF front-end implementation based on the symmetrical transceiver architecture proposed earlier [27]. A 900 MHz WPT signal is used as a mutual clock reference and the communication is done over an impulse UWB link at 7.9 GHz center frequency. The scalable data-rate of UWB link up to 112 Mbit/s has been demonstrated as well as robustness against narrowband interference.

2) Non-Volatile Memory (NVM) technology

The main reason to pick up PCM in favor of any other memory technology [31] were the benefits of PCM technology, e.g., the estimated high number of read/write cycles as 1×10^6, which consequently results in need of no or just a lightweight wear leveling algorithm, and the bit alterability – lack of need of block erase cycles (as with flash memory) when data should be stored. From the perspective of technology lifecycle PCM stands now between a pure innovative technology and early adopters' stage. There are several 90 nm products [32] on the market already and more to come.

Aggregating main memory characteristics in comparison with NAND/NOR flash technology and dynamic random-access memory (DRAM) execution memory, PCM stands between those two in terms of cost per die. It is characterized as 5.5 F^2 factor in cell size having the same wafer complexity as DRAM technology. Currently only Single Level Cell (SLC) PCM is available, though Multi-Level Cell (MLC) PCM is on the way out, which can substantially extend the density and, justify the cost structure [31]. Thus, the application range can be quite wide from external usage (cards, keys) and wireless applications (RF memory tags) to high performance computing applications (caches, code execution, data storage). Considering reliability characteristics it is important to note that PCM technology gives more than 10 years retention ratio that can be extended even further, if necessary, by proper bit error management.

PCM has performance characteristics such as read & write latency and read & write endurance almost as good as DRAM, while giving clear benefits through the non-volatile nature of PCM technology. PCM has a low system-wise energy consumption (~0.2 mW/pF read, <1.25 mW write) ~<1 mW/GB of idle power, access time comparable to DRAM (~85 ns), with read latency 50–100 ns, write bandwidth from 10 to 100+ Mbit/s/die, write latency 500 ns – 1 μs, various packaging and die stacking solutions, high-speed low-pin-count low-power interface solutions, and maturity of the technology as such.

The PCM technology highlights provide clear reasoning for the selection of such technology for the RF memory tag application, preserving the opportunity to justify it even further when some other application should be designed.

B. *Software Architecture (protocol stack)*

The MINAmI software architecture (protocol stack) is designed to be modular and scalable. The protocol stack is based on three layers: Network Layer, MAC Layer, and PHY Layer. The APIs of the layers are open for 3rd parties. These layers will be presented in the following sections. The protocol stack has been developed taking into account future compatibility with NoTA architecture. Care should be taken to have a clear implementation path towards the final architectural (NoTA) solution.

1) *Network Layer*

Network Layer will first only provide point-to-point connections regardless of state. In future, also applications using multiple targets could become feasible when MINAmI Subsystem is in active mode. If a point-to-multipoint network protocol is needed, nanoIP is easily implementable [18][33]. However, to get full internet support classical IP protocol may be valid, and more common in networking devices. In the final architecture (NoTA) solution, the network layer will consist of Device Interconnect Protocol (DIP), as a middleware, which guarantees the compatibility with NoTA. In DIP protocol, it is possible to select, which transport mode and network is used. For example DIP TCP L_IN (transport selected) is ready to be used within one device and between several devices in a sub-network as such. Multicasting must be enabled in IP interface in order for device discovery to work. Nodes, which are in different sub-network, cannot be detected [21].

Packet size is an important parameter and depends on what is feasible for MAC and PHY layers. Upper layer packets are segmented and reassembled and this is dependent on what kind of packet sizes the system supports.

2) *MAC Layer*

The MAC of the novel dual-band radio interface has three different operational modes: the passive mode, where no internal power source is available or used; and the active and semi-passive modes, where internal power source is available. Tags on battery-less objects without power wire connection (e.g., implanted on paper) are passive.

In the active mode, the mobile phone actively searches and selects the target tags, sends the targets the WPT signal for powering and for frequency synchronization of the communication link, reads/writes data on the tags, and closes the connection to the target when active connection is no longer required. This operation can be an automatic feature, or enabled by the user (initiating the application for reading and writing the tag). In the semi-passive mode the phone receives data sent by an outside device, but powers itself, allowing a longer communication range, which would otherwise be limited by the WPT link. In semi-passive mode, however, the initiator device takes care of the synchronization of the I-UWB communication link.

Active mode states are used by battery-powered mobile devices, whereas passive mode states are applied for passive devices and tags. In passive mode, possible connections are powered by an outside device with WPT. In the passive operating mode, the default state (when powered by an outside device) is P-IDLE, i.e., ready to receive any data, after the boot-up sequence.

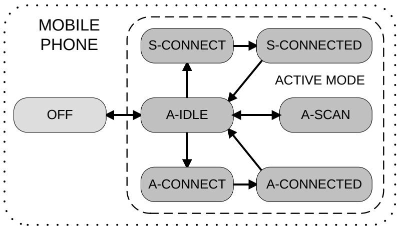


Figure 5. Active (and semi-passive) UWBLEE MAC states on a mobile phone. Active states denoted with A, semi-passive with S.

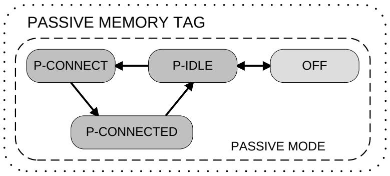


Figure 6. Passive UWBLEE MAC states on a RF memory tag.

TABLE I: UWBLEE PHY IN DIFFERENT MAC STATES

	MAC mode		
	Passive	***Semi-passive***	***Active***
I-UWB	Transmit / receive		
WPT synch	Receive	Receive	Transmit
WPT power	Receive		Transmit
Power source	WPT reception	Battery	Battery
Remarks	Being read / written		Reading / writing other devices

The main operational states of UWBLEE MAC are shown in Figures 5 and 6. In addition to the shown directions of movement from state to state, there need to be possibility of built-in error recovery operation from any operational state to the corresponding idle state (A-IDLE or P-IDLE). For the applications requiring higher security, a suitable security protocol can be applied for the ongoing data transmission.

3) *Physical Layer*

UWBLEE PHY layer controls both the I-UWB communications and Wireless Power Transfer (WPT) transmission. Depending on the operational mode (active or passive) WPT link is used to send (or receive) power and/or to provide the clock reference signal.

UWBLEE PHY is divided to two sub-blocks: I-UWB PHY and WPT PHY. I-UWB PHY controls the Impulse-UWB radio interface and WPT PHY controls the Wireless Power Transfer interface. I-UWB PHY and WPT PHY are coordinated by UWBLEE PHY so that I-UWB transmission is synchronized with the WPT transmission.

The function performed by UWBLEE PHY is defined by UWBLEE MAC, as shown in Table 1.

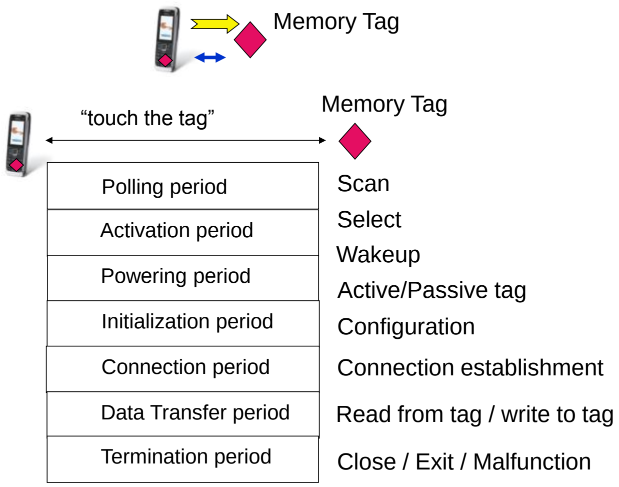


Figure 7. Mobile reader/writer to RF memory tag communication sequence

C. *Packet-level Communication*

The MINAmI subsystem communication between active mobile reader/writer and passive RF memory tag consists of periods shown in Fig. 7. In the beginning, there are no tags within the mobile reader/writer local connectivity coverage. If the mobile reader/writer detects a tag during the powering period, it tries to scan all tags available (in the polling period) and – based on the current selection criteria – choose one with whom to communicate (in the activation period). The right tag is found by scanning the coverage area, synchronizing communications with the tags, and selecting the right tag. After this selection, connection and device configuration is executed in the initialization period to set communication parameters, to specify packet level parameters (e.g., length, memory allocation). The connection period is initiated when connection between mobile reader/writer and selected tag is established. This is followed by the data transmission period, reading and/or writing selected content from/to tag. After successful data transmissions, in the termination period, connection is closed or continued with another read/write operation to the tag.

Basic connection procedure between a mobile reader and a tag is described in Fig. 8, which also identifies affected internal entities, e.g., MINAmI server, memory management, and communication entity (MAC and PHY layers). For the air interface, the data from/to the non-volatile storage memory (PCM) is buffered into a DPRAM buffer memory equal to the maximum packet size transferred over the air.

D. *File System Design*

The mobile phone can read tags and with writeable tags the phone can also write all or parts of their contents. The communication capacity between the mobile terminal and the RF memory tag is targeted to exceed 50 Mbit/s (as discussed in Section I). Plug-in software (External memory stack in Fig. 1) is required to facilitate seamless use of the tag memory for mobile phone applications.

The memory tag can be used as an extension to the local file system of the reader (e.g., mobile phone). The memory tag can be either a passive and cheap one (Fig. 9) or an active

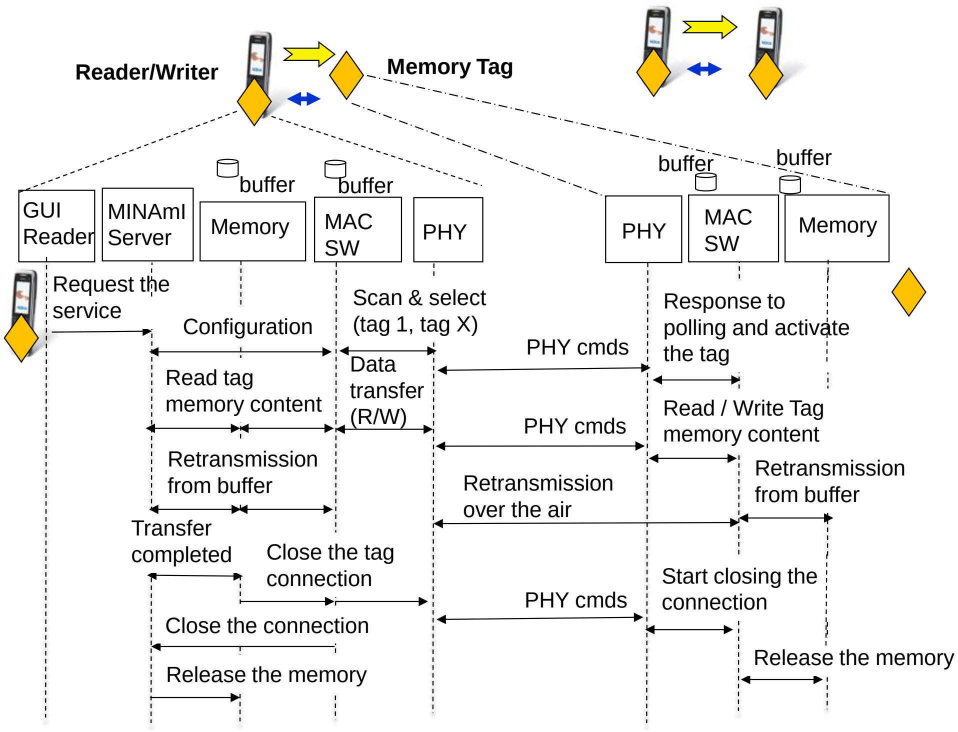


Figure 8. Basic MINAmI subsystem communication setup sequence

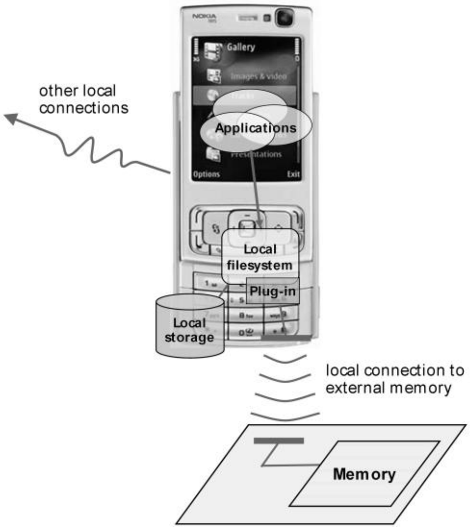


Figure 9. File system view of a mobile phone reading a passive memory tag: a cheap tag without its own processor [11].

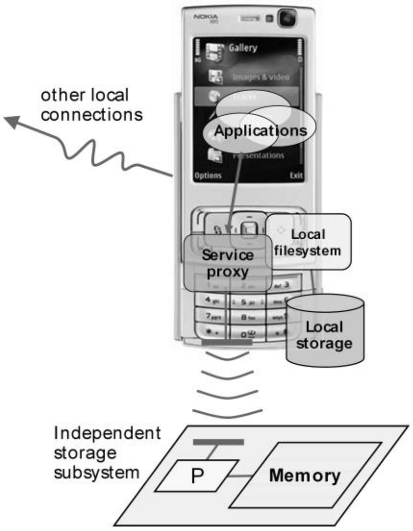


Figure 10. File system view of a mobile phone reading a passive memory: a more expensive tag with its own processor [11].

one, including an own power source and thus being more expensive (Fig. 10) [11]. Plug-in software in the file system of the reading device handles the connection to the memory tag. Storage space on the memory is mounted on the local file system in the same way as any detachable storage. The volatile nature of the connection causes overhead in maintaining the file system view in the reader/writer device. This kind of RF memory tag would be suitable for e.g., a concert ticket containing implanted multimedia available to be read with a mobile device.

Adding a processing element to the memory simplifies the connection. An ultra low-power processing element can process the access requests independently and even provide some more advanced services like metadata-based queries [34]. A service proxy relays the service interface of the memory directly to the applications running on the accessing device. The volatile nature of the connection is not a problem if the server is made stateless and transactions atomic. This type of RF memory tag will be able to support more complex use cases.

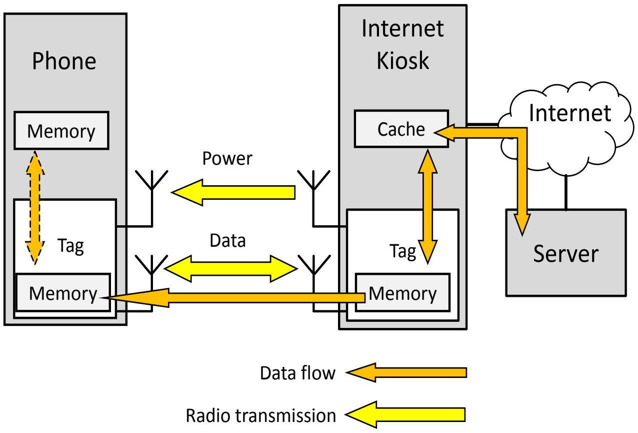


Figure 11. Mobile phone operating in passive or semi-passive mode to download data through an Internet Kiosk.

Device internal modules need to support NoTA to get full benefit of subsystem independency and still give a fast connection between subsystems. This interconnect architecture allows future extensions for modules within one device.

IV. Wireless Internet Access

Service providers and device manufacturers are continually challenged to deliver value and convenience to users by, for example, providing compelling network services. These services can include selling and distributing content. More effective and efficient way is needed to distribute the content. As a complementary solution to address the issue, RF memory tag based internet kiosk can provide the ultra-fast and power-efficient connectivity for downloading multimedia content, which is beneficial especially when ad-hoc downloading large amount of content.

The multimedia content such as audio and /or visual content can be ordered by users and/or pre-downloaded by service providers. The content is preloaded into a radio frequency memory tag installed in the Internet kiosk as shown in Fig. 11. When a user stays in the range of UWB, a request is generated for the content stored in the memory tag. The Internet kiosk initializes wireless transmission to push the content from the memory tag to the user's terminal in response to the request via UWB [35]. The Internet kiosk can be deployed in public spaces for users to access ubimedia applications, e.g. downloading magazines, newspapers or audio/movie multimedia for recreation in airports.

As discussed in Section III, most of the technologies, namely Wi-Fi, cellular, Bluetooth and NFC are designed for well-established communication between active devices, which result in relatively high power consumption and a long connection establishment time. In our RF memory tag based solution, MINAmI subsystem is within mobile devices to provide necessary connectivity to the Internet kiosk, memory tag and storage for downloading ubimedia content as illustrated in Figs. 3 and 4. The subsystem can be standalone and operational without maintaining from main application bus in mobile devices and consuming extra system resources.

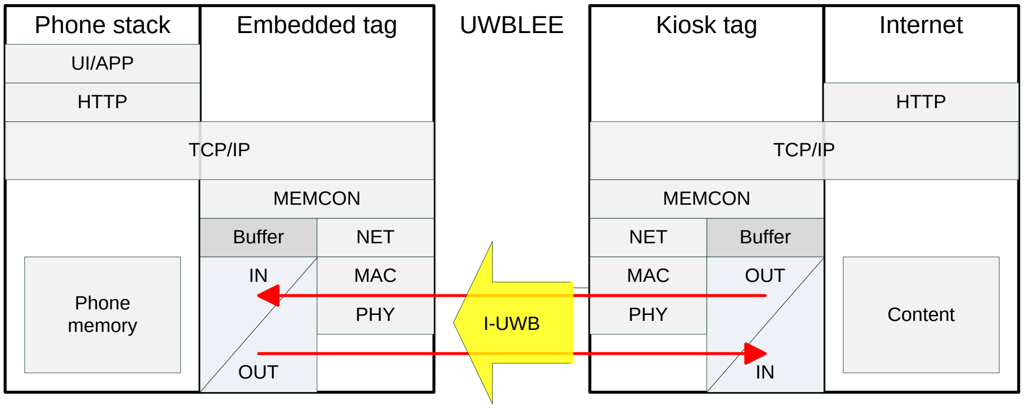


Figure 12. Schematic layout of the communication system when a mobile phone connects to the Internet via a memory-tag-based Internet kiosk.

Besides, the subsystem can be powered off until a new request arrives and it would not suffer from long latency in establishing a new connection.

In this architecture, the mobile devices could operate in either semi-passive or passive mode instead of active mode. The downloading is powered by phone battery in semi-passive mode and it is powered by the Internet kiosk in passive mode. By running in passive mode, the memory tag is powered by the Internet kiosk, and the mobile devices only consume energy to receive content and write these data into memory. This yields a great potential of energy saving on mobile devices especially when downloading large amount of content.

From the communications point-of-view, under the end-to-end Transmission Control Protocol – Internet Protocol TCP/IP layer there is a memory connection (MEMCON) layer connecting the content of the phone tag and the kiosk tag (see Fig. 12). The memory within the tags is divided into at least two parts, of which one is outgoing and one incoming data area. The master device (in the case of the kiosk, the kiosk tag) reads automatically the data in the outgoing data area of the slave device (phone tag), copying it to the incoming data area of its own memory. The data is then read by the controlling software and possible application level commands (e.g., fetch content from web address) found are then carried out. The reply data (e.g., data fetched from the web address) is then written to the outgoing data area of the kiosk tag, which is automatically copied to the incoming data area of the phone tag. Thus the memory connection layer automatically reads and copies the outgoing data from each device to the incoming data area of the other device. The upper layer control software of each device then moves the data further to the Internet services (Kiosk) or phone memory (phone) to be used by the application requesting the data. From the point-of-view of an application on the phone requesting data (e.g., Internet message access protocol (IMAP) email download) from a service provider in the Internet, there is a TCP/IP connection available.

The solution also makes possible providing an instant content download possibility by preloading data from an Internet service to the phone, e.g., a web page, digital magazine, or email account contents. In that case, the kiosk can be labeled with the logo of the corresponding service, telling the user what he/she would get by touching the logo with the phone. As the data is pre-loaded to the tag, the speed of delivery is only affected by the speed of the UWBLEE connection and data handling and displaying within the phone.

A. Comparisons

Maximized throughput and minimized power consumption are critical requirements in order to extend battery life of mobile devices. Given the scenario of Internet Kiosks, various radio technologies could be utilized to provide Internet connectivity from mobile phones to the Internet Kiosks. Normally, Wi-Fi and femtocell focus on local coverage. They are widely deployed and typically used by mobile devices. However, they are not always the most viable solutions. When traveling abroad, data roaming over cellular network, such as 3G (WCDMA) may be very expensive.

3G and Wi-Fi typically drain battery quickly on mobile devices. Since the fixed overhead of transmission is significantly high when the radio interfaces are in communication state. Once the radio interface is on and operates in active state, most of the power is consumed on circuits and does not matter how many data are sent or received over the interface. Especially in 3G networks, the radio switches to the higher power states, DCH (Dedicated Channel) or FACH (Forward Access Channel) from IDLE state, when the network is ac-

TABLE II. COMPARISON OF POWER CONSUMPTION OF DIFFERENT DATA COMMUNICATION TECHNOLOGIES

Wireless interface	**Max data rate**	**Maximum application throughput**	**Power consumption**	**Energy consumption**
	Mbit/s		***mW***	***nJ/bit***
3G	7.2	~5	~850	~170
IEEE 802.11g	54	~20	~500	~25
Bluetooth 2.0+ EDR	3	~2.1	~60	~28
NFC	6.78	848 kbit/s	~30	~35
UWBLEE	112	~50	~5.4	50-60 pJ/bit (*
*) The value 50-60 pJ/bit is only for the RF front-end of UWBLEE				

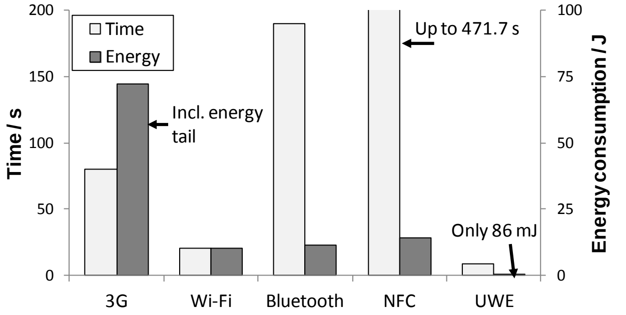


Figure 13. Time and energy consumed of using different radio technologies when downloading 50MB movie trailer.

tive [36]. Based on our measurements on a Nokia N900 phone, IDLE state is considered as low power states, which consume only around 30 mW. The state of Cell FACH consumes around 400 mW and the state of Cell DCH consumes around 800 mW. According to 3GPP standard, there are so called inactivity timers managed by the radio network controller (RNC). The transitions between the different states are controlled by inactivity timers. Transitioning from the high to the low power state immediately after a packet is transmitted, the device transitions only when the network has been inactive for the length of the inactivity timer. This mechanism serves two benefits: 1) it alleviates the delay incurred in moving to the high power state from the idle state, and 2) it reduces the signaling overhead incurred due to channel allocation and release during state transitions. Since lingering in the high power state also consumes more energy, network operators set the value of the inactivity timer based on this performance/energy trade-off, with typical values being several seconds long. However, these timers result in extra energy consumption even if there is no data to be sent or received since the radio has to wait for the timers to expire. The energy consumption is defined as tail energy [36].

To overcome energy consumption constraint in 3G and WLAN networks, short-range radio communication could be used. There are several radio technologies that can be considered for the use of Internet Kiosks. For instance, Bluetooth, NFC and UWBLEE where the mobile phone could operate in semi-passive mode in which communications are powered by the Internet Kiosk.

In order to reduce power consumption and extend battery life of mobile phones, battery-operated devices require being equipped with a radio technology with high bandwidth and low power consumption. Therefore, the mentioned radio technologies are taken into consideration of comparison. The transmission rate and energy consumption of receiving data over various radios are benchmarked in Table II.

Considering the scenario shown in Section I, which assumes a mobile user downloads 50 MB movie trailer, Fig. 13 demonstrates the time and energy consumption of using various radio technologies. In the figure, the value of time is shown in the left y-axis and the value of energy consumption in the right. In the results of the case in 3G, data rate follows High-Speed Downlink Packet Access (HSDPA) Category 10 in 3rd Generation Partnership Project (3GPP) Release 5 and we assumed inactivity timer lasts 5 seconds. For the UWBLEE technology the estimated total power consumption for the complete integrated transceiver with digital parts in passive and semi-passive operating modes is multiplied with the factor of 2 in comparison to the power consumption of RF front-end implementation [30]. However, the power consumption of digital parts is highly dependent on the total complexity allowed in passive and semi-passive operating modes, and on the optimized design of integrated circuit. The energy consumption does not include the part of writing data into memory storage in all the cases and only shows the energy consumption of receiving data via different radios. The power consumption of writing in our NoTA-based solution is around 2 nJ/bit for NAND flash and approximate 1–2 nJ/bit for PCM. Both of memories are considered competitive from energy efficiency point-of-view [37].

Based on the calculation, the time spent on downloading the movie trailer is only 8 s and the energy consumption of RF front-end is 0.043 J when using UWBLEE. As mentioned, in the total power consumption for the complete integrated transceiver the power consumption of digital parts must be taken into account. In addition, there is a great difference in the global power consumption of the system in the passive and semi-passive modes although the power consumption of the functions in the mobile phone is equal in the two modes. The reason for this is that in the passive mode the energy for communication is transferred wirelessly, whereas in the semi-passive mode the energy is taken from the battery. The efficiency of the WPT link, mandatory in the passive mode, is highly dependent on the factors such as transfer frequency, size of antennas, and distance, and it is obviously lower than in battery-powered case. Nevertheless, the estimated total energy consumption of downloading the trailer remains below 0.1 J (and below 1 J with the memory access) for UWBLEE in the mobile phone. Comparing other technologies with UWBLEE, the energy consumption of using 3G is up to 72.25 J, where the tail energy accounts for 6% of total energy consumption. Moreover, time consumed on downloading the movie trailer by using Bluetooth and NFC is up to 190.5 s and 417.7 s respectively. No matter from speed and energy saving point of view, our RF memory tag based solution would enable shorter time of downloading ubimedia content, better user experience, as well as smaller energy consumption.

V. Discussion and Conclusions

The RF memory tag (i.e., mobile reader/writer and tag) solution was developed and tested in the MINAmI project. Implementation is shown in Fig. 14. The development of a RF memory tag sub-system of MINAmI project is based on a flexible, field-programmable gate array (FPGA) based hardware platform. The sub-system takes benefit from the ultra-low power UWBLEE transceiver architecture, which is suitable for data rates required in RF memory tag applications. The technical results are promising and useful for the concept of mobile-phone-readable RF memory tags. The data-rate of 112 Mbit/s has been achieved over the novel radio interface in technical demonstrations [30]. This leaves

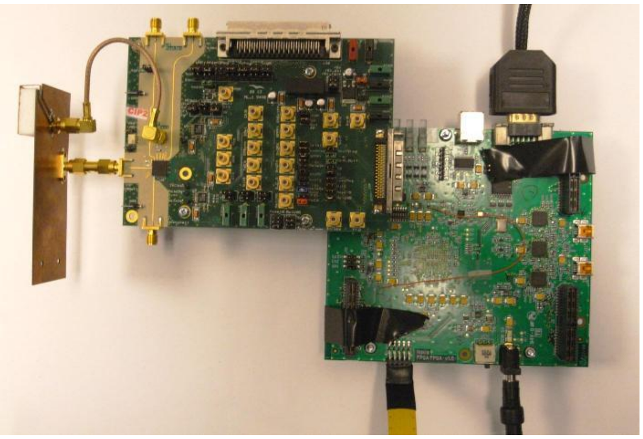


Figure 14. Our UWBLEE implementation

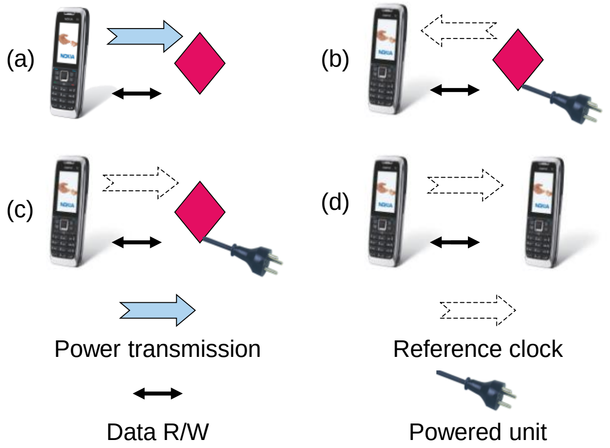


Figure 15. Some possible operational combinations of mobile phones interacting with RF memory tags.

room for up to 50% protocol and memory access overhead when targeting 50 Mbit/s end-to-end communications. On the PHY and MAC layers short target distance and point-to-point communication efficiently minimize the protocol overhead on packet level. However, efficient pipelining in buffering of the data is in crucial role in optimization of the end-to-end system. The third important factor is the memory access speed. This is relevant when reading data from the source memory and when writing the data to the target storage memory. As shown in Section III, the continuous development of NVM memories will provide power-efficient and fast solutions for the target applications. Altogether, the listed factors and the results achieved with the demonstration platform show that mobile reader/writer and the high capacity memory tag is implementable.

A. *Future development*

The UWBLEE wireless connection technology presented in this paper provides data rates significantly exceeding the existing NFC technology already in the market. From technology ecosystem point-of-view there is little sense in developing UWBLEE as an independent technology. UWBLEE can thus be seen as a possible future high-speed extension to existing RFID or NFC technologies.

As the range of this wireless interface is fairly short, in the range of 10 cm, there exist use-cases similar to the NFC use cases (range touch to 3 cm). Physical selection [38] by touching of a service-providing tag is thus possible. In such use, the tag would be marked with a logo of the corresponding service, such as title or picture of a movie or a magazine, making selection of the service intuitive and easy.

The possibility of using a mobile phone to read a passive tag is, naturally, not the only operational combination of these devices, as shown in Fig. 15, where (a) refers to a phone reading a RF memory tag, (b) to Internet kiosk based on a RF memory tag, (c) is a variant of (b), and (d) refers to data transfer between mobile phones. In a multi-device environment one device can work as a proxy for the memory tag and provide other devices with access to its services [11]. There are also possibilities to have memory tags with their own power sources, which eliminate the need of wireless powering. In that case, the reading range can be extended or power use within the mobile phone can be reduced. The phone can also communicate directly with other similarly equipped phones.

Our RF memory tag solution supports Nokia’s *Explore and Share* concept, a new way of transferring content (e.g., multimedia, maps, and applications) to a mobile phone [12]. RF memory tags feed users appetite for ever increasing local bandwidth and capacity requirements. Users would, naturally, invent new use cases and ways of utilizing these tags in the local content delivery domain. These *Express Tags* can explore new large content shared by others [39]. NoTA is well positioned in the transport agnostic technology. It fits to the many inter-device use cases, such as in ubiquitous world.

Our vision is that there is an ever-increasing need to move content from the Internet to mobile devices and vice versa, as well as between devices. The amount of available energy to support all this wireless traffic is not increasing correspondingly, however. Thus, possibility to distribute the energy consumption of wireless connections so that either of the endpoints takes care of most of the power usage is an interesting enabler to future applications.

B. *Conclusions*

The evolution of non-volatile memory technologies gives the basis for the vision about RF memory tags. However, the large memory creates a need for a high-speed data connection that can be used to transfer the contents of the tags in a timeframe acceptable for the user. The dual-band radio interface, UWBLEE introduced in this paper provides the required data rate and possibility for future scalability as memory sizes become larger.

Modular architecture is mandatory in the RF memory tag system to optimize performance. For example, latencies common in memory access of centralized systems are not acceptable. Power consumption of the mobile reader/writer is efficiently minimized with an independent sub-system keeping the involvement of the main processor at the minimum. In contrast to conventional radio systems, the main processor only triggers the communication and the independent subsystem handles the transfer and storage of the data. Thus, the main processor does not have to be involved in the low level communication processes.

In addition to the RF memory tag reader/writer capability in mobile devices, mobile devices can be also equipped with embedded RF memory tags. This enables a new usage scenario called internet kiosk which can be further used to enable internet connection seemingly with zero power consumption in the mobile device.

ACKNOWLEDGMENT

We thank all the partners of MINAmI project, who have contributed to development of this technology. This study is a part of MINAmI project under EU FP6 contract IST-034690. This research was also supported by the Finnish Academy under grant no 129446.

REFERENCES

[1] I. Jantunen, J. Hämäläinen, T. Korhonen, H. Kaaja, J. Jantunen, and S. Boldyrev, "System architecture for mobile-phone-readable RF memory tags". Proc. UBICOMM 2010. IARIA, 2010, pp. 310–316.

[2] V. Wu, M. Montanari, N. Vaidya, and R. Campbell, “Distributed RFID tag storage infrastructure”. University of Illinois at Urbana-Champaign, IL, USA. Tech. Rep. 2009.

[3] N. Ahmed, R. Kumar, R.S. French, and U. Ramachandran, ”RF2ID: a reliable middleware framework for RFID deployment”. Proc. IPDPS 2007. IEEE, 2007, pp 1–10.

[4] C. Ying, “A verification development platform for UHF RFID reader”. Proc. CMC'09. IEEE, 2009, pp. 358–361.

[5] N. Pillin, N. Joehl, C. Dehollain, and M.J. Declercq, “High data rate RFID tag/reader architecture using wireless voltage regulation”. Proc. RFID 2008. IEEE, 2008, pp. 141–149.

[6] J.T.E. McDonnell, J. Waters, W.W. Loh, R. Castle, F. Dickin, H. Balinsky, and K. Shepherd, “Memory Spot: A Labeling Technology”. IEEE Pervas. Comput., vol. 9, 2010, pp. 11–17.

[7] J. Nielsen, “Usability engineering”. Morgan Kaufmann, 1993.

[8] W. Cui, P. Ranta, T.A. Brown, and C. Reed, “Wireless video streaming over UWB”. Proc. ICUWB 2007. IEEE, 2007, pp.933–936.

[9] H. Witschnig, C. Patauner, A. Maier, E. Leitgeb, and D. Rinner, “High speed RFID lab-scaled prototype at the frequency of 13.56 MHz”. Elektrotechnik & Informationstechnik, vol. 124, 2007, pp. 376–383.

[10] J. Jantunen, I. Oliver, S. Boldyrev, and J. Honkola, ”Agent/space-based computing and RF memory tag interaction”. Proc. IWRT 2009. INSTICC PRESS, 2009, pp. 27–38.

[11] E. Kaasinen, M. Niemelä, T. Tuomisto, P. Välkkynen, I. Jantunen, J. Sierra, M. Santiago, and H. Kaaja, “Ubimedia based on readable and writable memory tags”. Multimedia Syst., vol. 16, 2010, pp. 57–74.

[12] M. Cooper, “Explore and Share – Nokia shows ultra-fast wireless data transfer concept”. Nokia Conversations, 23 Feb 2010.

[13] “The PayKiosks Opportunity”. PayKiosks Internet Terminals, 2011.

[14] A. R. Brisebois and R.Klein. “Enterprise femto based kiosk”. US Pat. App. 20100318417. 16 Dec. 2010.

[15] Bluetooth Kiosk System Project. webee.technion.ac.il /labs/comnet/Info/projects/winter08/cn02w08 12.1.2012

[16] K. Ok. V. Coskun, M.N. Aydin, and B. Ozdenizci, “Current Benefits and Future Directions of NFC Services”. Proc. ICEMT 2010. IEEE, 2010, pp. 334–338.

[17] www.fp6-minami.org 12.1.2012

[18] I. Jantunen, H. Laine, P. Huuskonen, D. Trossen, and V. Ermolov, “Smart sensor architecture for mobile-terminal-centric ambient intelligence”. Sens. Actuators A, vol. 142, 2008, pp. 352–360.

[19] Y. Têtu, I. Jantunen, B. Gomez, and S. Robinet, “Mobile-phone-readable 2.45GHz passive digital sensor tag”. Proc. RFID 2009. IEEE, 2009, pp. 88–94.

[20] K. Kronlöf, S. Kontinen, I. Oliver, and T. Eriksson, “A Method for Mobile Terminal Platform Architecture Development”. Advances in Design and Specification Languages for Embedded Systems, S.A. Huss, ed. Springer, 2007, pp. 285–300.

[21] projects.forum.nokia.com/NoTA/wiki 12.1.2012

[22] K. Keinänen, J. Leino, and J. Suomalainen, “Developing keyboard service for NoTA”. VTT, Espoo, Finland. Tech. Rep. 2008.

[23] www.mipi.org 12.1.2012

[24] “Open Modem Interface Proposal Based on Device Interconnect Protocol Version 1.0”. Nokia, Espoo, Finland. White paper, 2010.

[25] D. Truscan, J. Lindqvist, J. Lilius, I. Porres, T. Eriksson, J. Rakkola, and A. Latva-Aho, “Testable Specifications of NoTA-based Modular Embedded Systems”. Proc. ECBS 2008. IEEE, 2008, pp. 375–383.

[26] S. Asthana, and D.N. Kalofonos, “The problem of Bluetooth pollution and accelerating connectivity in Bluetooth ad-hoc networks”. Proc. PerCom 2005. IEEE, 2005, pp. 200–207.

[27] J. Jantunen, A. Lappeteläinen, J. Arponen, A. Pärssinen, M. Pelissier, B. Gomez, and J.A. Keignart, “New symmetric transceiver architecture for pulsed short-range communication”. Proc. GLOBECOM 2008. IEEE, 2008, pp. 1–5.

[28] S.R. Aedudodla, S. Vijayakumaran, and T.F. Wong, “Timing acquisition in ultra-wideband communication systems”. IEEE T. Veh. Technol., vol. 54, 2005, pp. 1570–1583.

[29] M. Pelissier, B. Gomez, G. Masson, S. Dia, M. Gary, J. Jantunen, J. Arponen, and J. Varteva, “112Mb/s full duplex remotely-powered impulse-UWB RFID transceiver for wireless NV-memory applications”. Proc. VLSIC 2010. IEEE, 2010, pp. 25–26.

[30] M. Pelissier, J. Jantunen, B. Gomez, J. Arponen, G. Masson, S. Dia, J. Varteva, and M. Gary, "A 112 Mb/s Full Duplex Remotely-Powered Impulse-UWB RFID Transceiver for Wireless NV-Memory Applications," IEEE J. Solid-St. Circ., vol.46, 2011, pp. 916–927.

[31] G.W. Burr, B.N. Kurdi, J.C. Scott, C.H. Lam, K. Gopalakrishnan, and R.S. Shenoy, “Overview of candidate device technologies for storage-class memory”. IBM J. Res. Dev., vol. 52, 2008, pp. 449–464.

[32] “Intel, STMicroelectronics deliver industry's first phase change memory prototypes”. Physorg.com, 6 Feb 2008.

[33] Z. Shelby, P. Mahonen, J. Riihijärvi, O. Raivio, and P. Huuskonen, “NanoIP: the zen of embedded networking”. Proc. ICC’03. IEEE, 2003, vol. 2, pp. 1218–1222.

[34] J. Arponen, A. Lappeteläinen, J. Jantunen, and O. Tyrkkö, ”Content storing device query”. US Pat. App. 20080281787. 13 Nov 2008.

[35] S. Boldyrev, I. Oliver, J. Jantunen, J. Arponen, and S. Balandin. “Method and apparatus for retrieving content via a service endpoint”. US Pat. App. 20110055351. 3 Mar 2011.

[36] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications”. Proc. IMC’09. ACM, 2009, pp. 280–293.

[37] S. Eilert, M. Leinwander, and G. Crisenza, “Phase Change Memory: A New Memory Enables New Memory Usage Models”. Proc. IMW’09. IEEE, 2009, pp. 1–2.

[38] P. Välkkynen, M. Niemelä, and T. Tuomisto, “Evaluating touching and pointing with a mobile terminal for physical browsing”. Proc. NordiCHI’06. ACM, 2006, pp. 28–37.

[39] S. Constantinescu, “Nokia Research Center working on high speed NFC enabled file transfers”. Intomobile.com, 9 June 2011.

Towards Statistical Analysis of the Impact of Playout Buffer on Quality of Experience in VoIP Applications

Tibor Gyires Yongning Tang Aishwarya Mishra Olusegun Obafemi
School of Information Technology
Illinois State University
Normal IL 61790 USA
tbgyires,ytang,amishra,oeobafe@ilstu.edu

***Abstract*—The speech quality of Voice over IP (VoIP) applications can be assessed subjectively as Quality of Experience (QoE) and objectively as Quality of Service (QoS). QoE is multifaceted, which ties together user perception and expectations to application, network performance, and various voice data processing (e.g., codec) and streaming (playout buffering) methods. Most of prior work focuses on understanding the impact of network performance on QoE, but not explicitly describing how playout buffer affects user satisfaction or QoE assessment. Towards this goal, this paper presents a statistical analysis of the correlation among QoE assessment, QoS measurement, and the impact of playout buffer on QoE assessment. In this paper, we first identify QoE as a function along two dimensions of network loss and delay to understand how different network factors as well as playout buffer affect QoE assessment. Then, we propose a new performance metric called playout buffer QoE impact factor (IF^{QoE}) to explicitly evaluate the effectiveness of playout buffer in terms of its contribution to QoE improvement. Finally, we validate IF^{QoE} to statistically show its accuracy in terms of its strong correlation with the results of QoE assessment. All our study is based on extensive simulations using various emulated or real network scenarios. Our simulation results show that IF^{QoE} can accurately evaluate the impact of playout buffer on QoE assessment using directly measurable network performance metrics.**

Keywords- Quality of Service, Quality of Experience, Playout Buffer, Impact Factor, Statistical Analysis, VoIP.

I. Introduction

In recent years, *Voice over IP* (VoIP) along with other multimedia networking applications has become one of the most important IP network services to end users. Correspondingly, a major paradigm shift on the quality assessment methods of multimedia networking applications has occurred from network-centric to user-centric. User perceived *Quality of Experience* (QoE) is given special attention by network operators and service providers to assess the overall level of users satisfaction and maintain acceptable quality of service for VoIP communications.

User perceived QoE in VoIP is generally described in terms of *Mean Opinion Score* (MOS) [4], the formal subjective measure of user satisfaction on received voice quality. QoE is multifaceted, which ties together user perception and expectations to application, network performance, and various voice processing (e.g., codec) and streaming (e.g., playout buffering) methods.

Most of prior work focuses on understanding the impact of network performance on QoE, but not explicitly describing how playout buffer affects user satisfaction or QoE assessment.

A typical VoIP application buffers incoming packets and delays their playout in order to compensate for variable network delays (i.e., jitter). Such an application buffer is commonly referred to as *Playout Buffer*. A playout buffer can allow late arrival packets to be smoothly played out. However, the fluctuating end-to-end network delays may enforce the size of a playout buffer to increase to a level to trigger user unsatisfactory delay. In addition, if the size of playout buffer is too small, some late arrival packets will still be dropped in playout buffer because their arrival time exceeds required presentation deadlines. The two conflicting goals of minimizing buffering delay and minimizing late packet loss have motivated various playout algorithms.

Our objective is to understand the impact of playout buffer on QoE in VoIP applications. In the paper, we study the correlations among network delay, network loss, buffering delay, buffering loss, and QoE. Our study aims at providing an easy-to-measure performance metric to accurately evaluate the effectiveness of a playout buffer on improving QoE assessment.

In this paper, we use a simple but representative evaluation model to study the correlation among QoE assessment, QoS measurement, and the effectiveness of playout buffer in terms of its contribution to QoE improvement. In this process, we first identify QoE as a function along two dimensions of network loss and delay to understand how different network factors as well as playout buffer affect QoE assessment. Then, we propose a new performance metric called playout buffer QoE impact factor (IF^{QoE}) to explicitly evaluate the effectiveness of playout buffer in terms of its contribution to QoE improvement. Finally, we validate IF^{QoE} to statistically show its accuracy in terms of its strong correlation with the results of QoE assessment. Our extensive simulations show that IF^{QoE} can accurately evaluate the impact of playout buffer on QoE assessment using measurable network performance metrics.

Our contribution is twofold: (1) we present an experimental study on measuring the dimensions of QoE assessment, and (2) we propose a new playout buffer performance metric called playout buffer QoE impact factor (IF^{QoE}), and provide a statistical analysis on the validation and accuracy of IF^{QoE} on evaluating the impact of playout buffer on QoE.

Though various approaches on showing QoS-QoE correlation have been proposed in the literature as described in Section II, to the best of our knowledge, none of them focuses on explicitly describing the impact of playout buffer on QoE assessment. After reviewing two basic QoE assessment methods in Section III, we elaborate our analytical methodology and propose IF^{QoE} in Section IV. We continue our study by first showing QoE dimensioning results in Section V, and then present a statistical analysis on the validation of IF^{QoE} in Section VI. Finally, we conclude our work in Section VII.

II. Related Work

There are numerous approaches proposed to objectively measure speech quality in VoIP. Robinson and Yedwab [10], [25] proposed a Voice Performance Management system to monitor call quality in real-time by proactively monitoring, alerting, troubleshooting and reporting network performance problems. Robinson and Yedwab [10] concluded that only packet loss, jitter and latency show the correlations between QoS and QoE.

Gierlich and Kettler [13] provided insight into the impact of different network conditions and the acoustical environment on speech quality. Testing techniques for evaluating speech quality under different conversational aspects were also described. Gierlich and Kettler [13] argued that there is no single number that can objectively indicate speech quality; and pointed out that overall speech quality is a combination of different single values from different speech quality parameters. Wang et. al., [14] designed and implemented a QoS-provisioning system that can be seamlessly integrated into current Cisco VoIP systems. Wang et. al., [14] also described Call Admission Control (CAC) mechanisms (Site-Utilization-based CAC and Link-Utilization-based CAC) to prevent packet loss and over-queuing in VoIP systems.

Myakotnykh and Thompson [15] described an algorithm for adaptive speech quality management in VoIP communications, which can show a real-time change in speech encoding parameters by varying voice packet sizes or compression (encoding) schemes. The algorithm involves the receiver making control decisions based on computational instantaneous quality level (which is calculated per talkspurt using the E-Model) and perceptual metric (which estimates the integral speech quality based on latency, packet loss and the position of quality degradation period in the call). Myakotnykh and Thompson [15] calculated the maximum achievable quality level for a given codec under specific network conditions, packet playout time, packet delay before jitter buffer and degradation in quality caused by traffic burstiness and high network utilization. The algorithm however results in an increase in average quality without increasing individual call quality.

Raja, Azad and Flanagan [16] designed generalized models to predict degradation in speech quality with high accuracy, in which genetic programming is used to perform symbolic regressions to determine Narrow-Band (NB) and Wide-Band (WB) equipment impairment factors for a mixed NB/WB context. Zha and Chan [17] described two algorithms for objective measurement of speech quality: single-ended (needing only to input the degraded speech signal) and double-ended (needing both the original and degraded speech signals). The algorithm developed by Zha and Chan [17] can objectively measure in real-time speech quality using statistical data mining methods.

Several algorithms have also been proposed to optimize some of the existing ITU-T models. The goal of optimization is to enhance existing models by correcting weaknesses that are identified in the models. Gardner, Frost and Petr [18] proposed an algorithm to optimize the E-Model by considering coder selection, packet loss, and link utilization. The authors however stated that the algorithm would have to be enhanced if used in a wide area network involving multiple user session. Mazurczyk and Kotulski [19] proposed an audio watermarking method based on the E-Model and the MOS, which provides speech quality control by adjusting speech codec configuration, playout buffer size and amount of Forward Error Correction (FEC) mechanism in VoIP under varying network conditions.

One of the limitations of the E-model is the fact that the model does not consider the dynamic nature of underlying networks that support VoIP. This limitation is addressed by several authors designing adaptive playout buffering to improve voice quality in VoIP. Most of these studies either optimize the E-Model, the PESQ [5] or combine the PESQ and the E-Model to propose a more holistic solution. Mazurczyk and Kotulski [19] highlighted two problems that are associated with adaptive playout buffering: how to estimate current network status and how to transfer network status data to the sending or receiving side. Wu et. al. [20] admitted that VoIP playout buffer size has long been a challenging optimization problem, as buffer size must balance the dynamics of conversational interactivity and VoIP speech quality. They stated that the optimal playout buffer size yields the highest satisfaction in a VoIP call. They further investigated the playout buffering dimensions in Skype, Google Talk and MSN Messenger, and concluded that MSN Messenger produces the best performance in terms of adaptive playout buffering, while Skype does not adjust its playout buffering at all.

$$MOS = \begin{cases} 1, & if\ \mathcal{R} \leq 0 \\ 4.5, & if\ \mathcal{R} \geq 100 \\ 7 \times 10^{-6}\mathcal{R}(\mathcal{R}-60)(100-\mathcal{R}) + 0.035\mathcal{R} + 1, & otherwise \end{cases} \tag{1}$$

Narbutt and Davis [21] stated that the management of playout buffering is not regulated by any standard and is therefore vendor specific. They proposed a scheme that extends the E-Model and provides a direct link to perceived speech quality, and evaluated various playout algorithms in order to estimate user satisfaction from time varying transmission impairments including delay, echo, packet loss and encoding scheme.

III. Quality of Experience Assessment

In this section we discuss two commonly used and well accepted quality of experience assessment methods: mean opinion score (MOS) and E-Model.

A. Mean Opinion Score

Mean Opinion Score or MOS has been endorsed by ITU-T as a subjective method to evaluate voice transmission quality. The MOS test involves using a group of testers (listeners) to assign a rating to a voice call. The quality is rated on a scale of 1 to 5, with $1 = bad$, $2 = poor$, $3 = fair$, $4 = good$ and $5 = excellent$ [2]. The arithmetic mean of the scores provided by all listeners becomes the final MOS value of the voice call. Assessment ratings can also be obtained by clustering the test results as “Good or Better” or as “Poor or Worse”, and further calculating the relative ratio or percentage of each type of results. For a given voice call, these results are expressed as “Percentage Good or Better” (%GoB) and “Percentage Poor or Worse” (%PoW) [3]. Table I shows the MOS rating, %GoB, %PoW and the correlation between each rating [4].

Table I: Subjective Ratings for Measuring QoE

User Satisfaction	**MOS** (5)	**%GoB** (100)	**%PoW** (0)
Very Satisfied	4.3-4.4	97.0-98.4	0.2-0.1
Satisfied	4.0-4.29	89.5-96.9	1.4-0.19
Some Dissatisfied	3.6-3.9	73.6-89.5	5.9-1.39
Many Dissatisfied	3.1-3.59	50.1-73.59	17.4-5.89
Nearly All Dissatisfied	2.6-3.09	26.59-50.1	37.7-17.39
Not Recommended	1.0-2.59	0-26.59	99.8-37.69

The advantage of the MOS is that it can provide an off-line analysis of end-user opinions. However, MOS tests cannot provide an absolute reference for the evaluations; that is, MOS ratings are dependent on the expertise of listeners [1]. Furthermore, MOS tests cannot be used in large scale experiments that involve a large number of users because of the involved overhead (e.g., test setup). Moreover, MOS tests are unrepeatable by nature.

B. E-Model

The E-Model, standardized by the ITU in 1998 as Recommendation $G.107$, provides a method for calculating a single metric representing voice quality, referred to as the *R-factor*, which can then be converted to estimate MOS values as shown in Eq. 1.

The E-Model is designed to measure the instant user perceived quality instead of the cumulative effect during an entire conversation. The E-Model assumes that individual impairment factors are additive on a psychological scale and combines the cumulative effects of these factors into the R-factor. The R-rating is on a scale of 0 to 100, with high values of R between 90 and 100 interpreted as excellent quality, while lower values of R indicate a lower quality. Values of R below 50 are considered unacceptable and values above 94.15 are assumed to be unobtainable in narrowband telephony. The E-Model measures individual impairment factors at different points in time to compute the R-rating. The value of the R-rating is consequently associated with measurements taken at a given time point and does not reflect the dynamic nature of quality during the entire length of a conversation.

The R-factor is expressed as the sum of five terms:

$$\mathcal{R} = \mathcal{R}_0 - I_s - I_d - I_e + A \tag{2}$$

R_O represents the basic signal-to-noise ratio, including noise sources such as circuit noise and room noise. The factor I_s is a combination of all impairments which occur simultaneously with the voice signal. The factor I_d represents the impairments caused by delay, and the effective equipment impairment factor I_e represents impairments caused by low bit-rate codecs and packet-losses of random distribution. The advantage factor A corresponds to the user allowance due to the convenience when using a given technology.

The E-Model not only takes in account the transmission statistics (transport delay and network packet loss), but it also considers the voice application characteristics, like the codec quality, codec robustness against packet loss and the late packets discard. However, the impact of playout buffer is simply converted into the impact of buffering delay and buffering loss, and thus not explicitly represented in E-Model.

In this paper, we are interested in finding the correlations of network performance (delay and packet loss) and user satisfaction assessment (MOS), and further relate these factors to the impact of playout buffer on QoE. Thus, we will adopt the recommended default values by the ITU-T

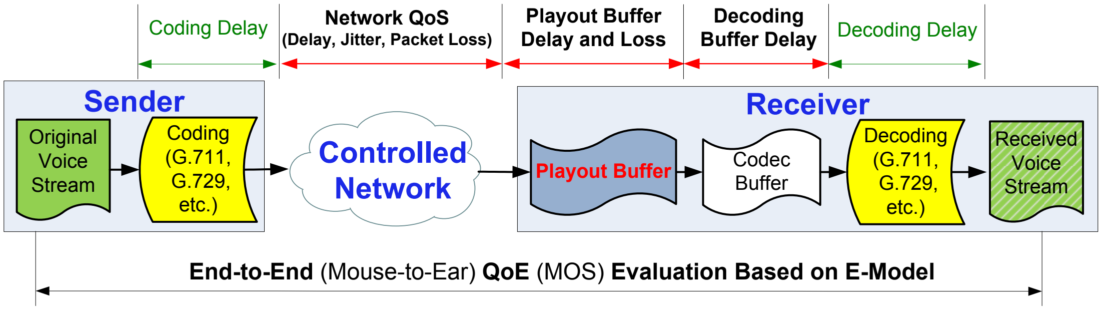


Figure 1: The Design of VoIP Speech Quality Assessment in Controlled Network Experiments.

Rec. G.107 for those intangible quantities (i.e., $\mathcal{R}_0$, A, I_s) and reduce the expression for the R-factor to:

$$\mathcal{R} = 94.2 - I_d - I_e \tag{3}$$

In the context of this work, delay impairment I_d comes from three sources: codec delay, network delay and playout buffering delay; and loss impairment I_e results from network packet loss and playout buffering packet loss.

IV. Analytical Methodology

In this section, we elaborate the network model and analytical methodology used in our study.

A. The Network Model

We generalize a typical VoIP application as a network system depicted in Fig. 1, which consists of a sender (caller), a receiver (callee), and a fully controlled network. On the sender, a voice stream is digitalized via a coding process, and then packetized to voice packets to send out. On the receiver, the received voice packets are first buffered in a playout buffer to compensate for network delay variation (jitter), and then further buffered in a codec buffer required by a decoding process.

It is worth noting that the playout and codec buffers are completely different from both their design objectives and their impacts on QoE assessment. A playout buffer is designed to allow the incoming voice packets with variant intervals (due to network jitter) can be played out as smooth as possible. Thus, a fixed or varying playout buffer delay is unavoidable depending on different buffering modes (fixed or adaptive); and moreover, some incoming voice packets may be dropped by the playout buffer if their arrival time later than required presentation deadlines. On the other side, codec buffer is required by decoding algorithm such that a minimum number of voice packets can be accumulated necessary for a decoding process being conducted. A codec buffer will cause a fixed buffering delay, but no packet loss.

Our study is performed in a well-known credible network simulation platform OPNET [11], which allows us (1) to choose a variety of codec schemes, (2) to create realistic networks supporting measurable performance metrics, (3) to flexibly control playout buffer; and (4) to estimate MOS (the result of QoE assessment) using E-Model.

In our study model, the sender can continuously send voice stream using a selected codec to the receiver over the network. The network can be fully controlled with specified network delay and loss rate to simulate various network conditions. A fully configurable playout buffer is presented on the receiver, which can operate in either fixed or adaptive mode with different parameters, including maximum buffer size, resizing interval, sliding mean coefficient.

According to Eq. 3, the impact of various components (the network, codec components, and playout buffer) on QoE assessment results from the total end-to-end accumulated delay (d_{tot}) and packet loss (e_{tot}). Since we consider the impact of coding and decoding delays into I_s, $d_{tot} = d_{net} + d_{buff} + d_{cbuff}$, and $e_{tot} = e_{net} + e_{buff}$. Here, d_{net}, d_{buff} and d_{cbuff} are delays caused by the network, playout buffer, and codec buffer, respectively. e_{net} and e_{buff} are packet loss rates caused by the network and playout buffer, respectively.

We proceed our experimental study in the following steps:

- To detect *Minimum Codec Buffer* (MCB): We remove the playout buffer and set the network to an ideal condition with a minimum constant network delay and no packet loss. We then gradually increase the size of codec buffer from the lowest value ($1ms$) to a more than enough large value (e.g., $250ms$), and use the measured MOS values of a continuous voice steam from the sender to the receiver to analyze the required minimum codec buffer for a specific codec, which will be further discussed later. Apparently, in such an ideal network condition, the playout buffer is unnecessary (no delay variation). Therefore, once the codec buffer reaches the corresponding MCB for a given codec, the measured MOS value should present a clear jump when the codec buffer size is changed from right below MCB to MCB.
- To investigate QoE dimensions using network loss and delay: We still keep the playout buffer removed, and control the network with various constant delays and loss rates. With all network conditions, we use the

measured MOS values of a continuous voice steam to find the user tolerable QoE boundary dimensioned in network loss and delay.

- To validate the new proposed playout buffer QoE impact factor IF^{QoE}: We validate the accuracy of IF^{QoE} for measuring the effectiveness of a playout buffer on QoE improvement. Specifically, for a given network condition, we configure two VoIP systems with and without playout buffer, respectively. We use the measured MOS values in these two cases to evaluate the improvement of QoE, which is compared to the results according to the computation of IF^{QoE}. We present a statistical study to show the accuracy of IF^{QoE} in measuring the impact of playout buffer on QoE. Finally, we use IF^{QoE} to evaluate several playout strategies in VoIP applications.

B. Experimental Design

For simplicity of presentation, we show in Table II all configurable parameters of our study model and the measured objects.

Table II: Configurable Parameters and Measurable Results

Configuration Parameters & Their Settings	
Codec	Encode/decode schemes (G.711, etc.)
Network Discard Ratio	The percentage of packets dropped
Network Latency	Delay dist, fixed values, scripted dist
Buffer Sizing Interval	Playout Buffer Resizing time
Maximum Buffer Size	Measured by buffer delay
Sliding Mean Coefficient	Coefficient for new talkspurt data
Playout Mode	fixed or adaptive buffer size

Measured Objects & Their Implications	
MOS	Estimated mean opinion score
Jitter	delay variation
Instant Playout Buffer Delay	the same as current buffer size in ms
Instant Playout Buffer Loss	pkt loss rate due to large pkt intervals
Network Loss Rate	ratio of lost pkts in network
End-to-end Delay	the total pkt delay from mouth to ear
Traffic Sent	Average received pkts/bytes per second
Traffic Received	Average sent pkts/bytes per second

For each experiment run with a specific setting, we keep the sender continuously sending voice stream to the receiver for one hour, and take 100 samples every second for all measured objects. We repeat 100 runs for each experiment and report the corresponding sample means. Please note, the actual execution time for each run is much shorter than the simulated running period. For example, the average execution time for a one-hour run is only $36s$ in a regular PC with Intel Core 2 Duo 2.66 GHz CPU and 3 GB memory.

C. Playout Strategies

Most of the adaptive playout algorithms described in the literature perform continuous estimation of the network delay and its variation to dynamically adjust the talkspurt playout time. Standard adaptive playout algorithm is based on Jacobsons work on TCP round trip time estimation. The algorithm estimates two statistics: the delay itself and its variance as shown in Eq. 4 and Eq. 5, and uses them to calculate the playout time [12].

$$\widehat{d_i} = (1-\alpha) \times \widehat{d}_{i-1} + \alpha \times n_i \tag{4}$$

$$\widehat{v_i} = (1-\alpha) \times \widehat{v}_{i-1} + \alpha \times |\widehat{d_i} - n_i| \tag{5}$$

Here, $\widehat{d_i}$ is the estimated amount of time from when the i^{th} packet is generated by the sender until it is played out at the receiver; n_i is the total delay introduced by the network. $\widehat{v_i}$ is the delay variance of i^{th} packet. α is called sliding mean coefficient in our study ($0 \leq \alpha \leq 1$).

Several other methods were also introduced to better estimate network delay. For example, instead of using a single sliding mean coefficient, two different sliding mean coefficients were used to adapt more quickly to short burst of packets incurring long delays. The idea behind the different playout strategies described in this paper is simple and all follow the so-called absolute timing method as defined by Montgomery [23].

If both the propagation delay and the distribution of the variable component of network delay are known, a fixed playout delay can be computed such that no more than a given fraction of arriving packets are lost due to late arrival. In such approach, the playout delay is fixed either for the length of the voice call, or is recalculated at the beginning of each talkspurt.

One potential problem with this approach is that the propagation delay is not known (although it can be estimated and typically remains fixed throughout the duration of the voice call). A more serious concern is that the end-to-end delay distribution of packets within a talkspurt is not known, and can change over relatively short time scales.

An approach to dealing with the unknown nature of the delay distribution is to estimate these delays and adaptively respond to their change by dynamically adjusting the playout delay. In this study, we define four playout strategies to describe such delay estimation and dynamic playout delay adaptation. As we will see, these strategies determine a playout delay on a per-talkspurt basis. Within a talkspurt, packets are played out in a periodic manner, thus reproducing their periodic generation at the sender. However, the playout strategies may change the playout delay from one talkspurt to the next, and thus the silence periods between two talkspurts at the receiver may be artificially elongated or compressed (with respect to the original length of the corresponding silence period at the sender). Compression or expansion of silence by a small amount is not noticeable in the played out speech.

When playout buffer resizing is necessary, an appropriate new buffer size can only be estimated, which also reflects the estimation of the network condition before next resizing opportunity. Algorithm 1 shows a commonly used dichotomic search algorithm for computing new buffer size. In this algorithm, first, an expected MOS value is calculated with new buffer size set to the average of maximum and minimum buffer sizes (line 2). Then, the new (expected) MOS value is used to update the smaller one between the MOS values when choosing the minimum and maximum buffer sizes, respectively (line 3-9). Finally, the algorithm chooses the buffer size that generates a higher MOS value (line 10-13). It is worth noting that the buffer size is not proportional to the MOS value, and thus it is possible that MOS$_{min}$ may be larger than MOS$_{max}$ (line 10).

Algorithm 1 PlayoutBufferResizing()

```
1: while (BuffSize$_{max}$− BuffSize$_{min}$ $> 1$) do
2:     MOS$_{current}$ $\leftarrow$ MOSCompute((BuffSize$_{max}$+ BuffSize$_{min}$)$/2$)
3:     if MOS$_{min}$ $<$ MOS$_{max}$ then
4:         MOS$_{min}$ $\leftarrow$ MOS$_{current}$
5:         BuffSize$_{min}$ $\leftarrow$ (BuffSize$_{max}$+ BuffSize$_{min}$)$/2$
6:     else
7:         MOS$_{max}$ $\leftarrow$ MOS$_{current}$
8:         BuffSize$_{max}$ $\leftarrow$ (BuffSize$_{max}$+ BuffSize$_{min}$)$/2$
9:     end if
10:    if MOS$_{min}$ $>$ MOS$_{max}$ then
11:        BuffSize $\leftarrow$ BuffSize$_{min}$
12:    else
13:        BuffSize $\leftarrow$ BuffSize$_{max}$
14:    end if
15: end while
16: return BuffSize
```

Clearly, these control parameters discussed above play important role in the performance of a playout buffer in terms of its impact on QoE assessment. In this paper, we denote a playout strategy s as a tuple: <Buffer Sizing Interval τ, Sliding Mean Coefficient α, Maximum Buffer Value $\nu >$, or simply $< \tau, \alpha, \nu >$. Buffer Sizing Interval τ decides how often the adaptive resizing should be decided. For example, resizing can be taken at the moment between talkspurts or in a fixed periodic interval (e.g., $10ms$). Sliding Mean Coefficient α is a coefficient for new spurt data to compute the playout buffer size, which can be set empirically. For example, as the experimental results shown in [12], α was set to 0.998002 in a single parameter estimation function as Eq. 4, or two different values in a double parameter estimation function with $\alpha = 0.998002$ for increasing trends in the delay and $\alpha = 0.75$ for decreasing trends. Maximum Buffer Value ν specifies the maximum buffer limit, which is measured in the delay experienced by a packet in the buffer.

D. Playout Buffer QoE Impact Factor: IF^{QoE}

Essentially, a playout buffer is designed to improve QoE, especially when experiencing fluctuating network delays. To the best of our knowledge, there is no prior work showing how to practically and accurately evaluate the effectiveness of a playout buffer from the perspective of QoE improvement. In this section, we tackle this challenge by proposing a new performance metric for playout buffer evaluation.

Recalling our discussion in Section III-B, we have presented the R-factor as the following function, which has been also shown previously in Eq. 3 with $I_d = \mathcal{F}(d_{tot})$ and $I_e = \mathcal{G}(e_{tot})$:

$$\mathcal{R} = 94.2 - \mathcal{F}(d_{tot}) - \mathcal{G}(e_{tot}) \tag{6}$$

Both $\mathcal{F}()$ and $\mathcal{G}()$ are monotonically increasing functions. Assuming that the same voice stream is sent over the same network to two VoIP systems with the only difference that one has playout buffer (denoted as S_{buff}) and another one does not (denoted as S_{nobuff}). The playout buffer in S_{buff} will introduce buffering delay and buffering loss, which does not appear in S_{nobuff}. With the above assumption, we have the following conclusion:

$$\begin{aligned} \mathcal{R}_{buff} &= 94.2 - \mathcal{F}(d_{net} + d_{buff}) - \mathcal{G}(e_{net} + e_{buff}) \\ \mathcal{R}_{nobuff} &= 94.2 - \mathcal{F}(d_{net}) - \mathcal{G}(e_{net}) \end{aligned} \tag{7}$$

The above equations imply that $\mathcal{R}_{buff} \leq \mathcal{R}_{nobuff}$ is always true, which apparently contradicts our intuition. The contradiction results from mistakenly calculated $\mathcal{G}(e_{net})$ in $\mathcal{R}_{nobuff}$. For a network with varying delays, the received VoIP packets may be dropped due to their varying arrival intervals that cannot meet their presentation deadlines required by the decoding process on the receiver. We refer to such packet loss due to missing playout buffer as e_{nobuff}. Thus, we rewrite the above equation Eq. 7as:

$$\begin{aligned} \mathcal{R}_{buff} &= 94.2 - \mathcal{F}(d_{net} + d_{buff}) - \mathcal{G}(e_{net} + e_{buff}) \\ \mathcal{R}_{nobuff} &= 94.2 - \mathcal{F}(d_{net}) - \mathcal{G}(e_{net} + e_{nobuff}) \end{aligned} \tag{8}$$

In order to make $\mathcal{R}_{buff} > \mathcal{R}_{nobuff}$, the following condition should hold:

$$\mathcal{G}(e_{net}+e_{nobuff})-\mathcal{G}(e_{net}+e_{buff}) > \mathcal{F}(d_{net}+d_{buff})-\mathcal{F}(d_{net}) \tag{9}$$

The condition above clearly shows the tradeoff between two conflicting design objectives of playout buffer to minimize both d_{buff} and e_{buff}. A good playout algorithm should pay minimal cost d_{buff} to gain maximum reward $e_{nobuff} - e_{buff}$. To fairly evaluate different playout strategies in terms of QoE improvement, the new performance metric of playout buffer should indicate both the absolute QoE gain (denoted as Q_{gain}) and the relative QoE gain ratio (denoted as Q_{ratio}) as defined in Eq. 10:

Considering various empirical functions proposed for practically calculating $\mathcal{F}(d_{tot})$ and $\mathcal{G}(e_{tot})$ (e.g., [24]), the relation between $\mathcal{F}(d_{tot})$ and d_{tot} can be regressed to a linear function; and a logarithmic line can fit the correlation curve between $\mathcal{G}(e_{tot})$ and e_{tot}. According, we propose IF^{QoE} as the new performance metric for playout buffer shown in Eq. 11.

$$\begin{aligned} Q_{gain} &= Q_{buff} - Q_{nobuff} = [\mathcal{G}(e_{net}) - \mathcal{G}(e_{net} + e_{buff})] - [\mathcal{F}(d_{net} + d_{buff}) - \mathcal{F}(d_{net})] \\ Q_{ratio} &= \frac{Q_{buff} - Q_{nobuff}}{Q_{nobuff}} = \frac{[\mathcal{G}(e_{net}) - \mathcal{G}(e_{net} + e_{buff})] - [\mathcal{F}(d_{net} + d_{buff}) - \mathcal{F}(d_{net})]}{94.2 - \mathcal{F}(d_{net}) - \mathcal{G}(e_{net} + e_{nobuff})} \end{aligned} \tag{10}$$

$$IF^{QoE} = [\mathcal{G}(e_{nobuff}) - \mathcal{G}(e_{buff})] \times \frac{\mathcal{F}(d_{nobuff})}{\mathcal{F}(d_{buff})} \tag{11}$$

Intuitively, the more the reward indicated by$\mathcal{G}(e_{nobuff}) - \mathcal{G}(e_{buff})$ and the less the cost indicated by $\frac{\mathcal{F}(d_{nobuff})}{\mathcal{F}(d_{buff})}$, the higher the IF^{QoE}.

To analyze the accuracy of IF^{QoE}, we adopt the following two commonly used empirical functions introduced in [24]. Here, the empirical function for $\mathcal{G}(e)$ is specific to G.711. Similar functions exist for other codecs, but will not be discussed in this paper.

$$\begin{aligned} \mathcal{F}(d) &= 0.024d + 0.11(d - 177.3)H(d - 177.3) \\ \mathcal{G}(e)_{G.711} &= 30\ln(1 + 15e)H(0.04 - e) + \\ &\quad 19\ln(1 + 70e)H(e - 0.04) \end{aligned} \tag{12}$$

where $H(x)$ is the Heavyside (or step) function such that:

$$H(x) = \begin{cases} 0, & if\ x < 0 \\ 1, & if\ x \geq 0 \end{cases} \tag{13}$$

In the case of packet loss rate greater than 4%, which is used in our following study, we can calculate IF^{QoE} as the following:

$$IF^{QoE} = 19\frac{d_{nobuff}}{d_{buff}} \times \ln \frac{1 + 70 \times e_{nobuff}}{1 + 70 \times e_{buff}} \tag{14}$$

Among the four parameters in Eq. 14, e_{buff} and d_{buff} are commonly obtained by monitoring the impact of playout buffer on packet loss and delay. d_{nobuff} can be calculated using Eq. 4 to estimate end-to-end delay between the sender and receiver. Different codec has different jitter tolerance. For example, G.711 can tolerate jitter up to $20ms$. For obtaining e_{nobuff}, we first use the information from RTP header to estimate the current network jitter. Then all incoming voice packets with jitter more than the tolerance will be counted as dropped ones to estimate e_{nobuff}.

V. QoE Dimensioning

In this section, we first identify minimum codec buffer. Then we present our study on QoE dimensioning using network loss and delay.

A. Minimum Codec Buffer

With respect to voice over IP, a codec is an algorithm used to encode and decode the voice conversation. A original analog voice signal needs to be converted (or encoded) to a digital format suitable for transmission over the Internet. Once at the other end, it needs to be decoded for the receiver. There are a variety of Codecs available and many of which utilize compression in order to reduce the required bandwidth of the conversation. The impairment of Codec on QoE comes from two aspects: (1) compression reduces the signal to noise ratio, and (2) when heavy compression is used, it takes time which adds a delay to conversation.

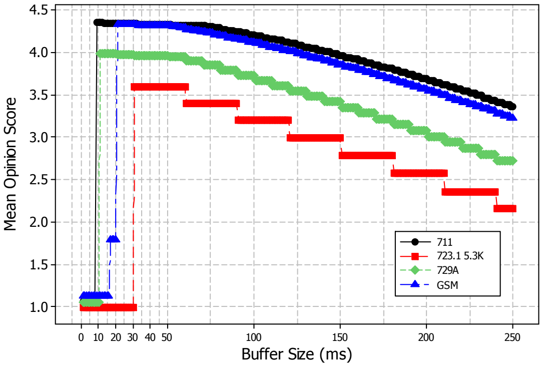


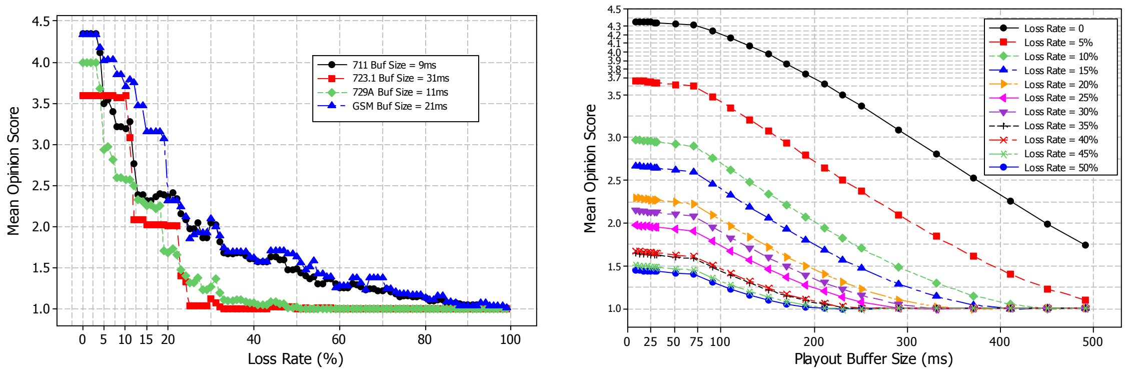
Figure 2: The Impact of Minimum Codec Buffer on MOS.

To experimentally find the MCB for each codec, we set the network to an ideal condition with only a minimum constant network delay and no network loss. Then we increase the codec buffer size from $1ms$ to $250ms$. In such an ideal network condition, a close to maximal MOS is expected if the codec buffer size is set to MCB. Thus, we use the measured MOS with increasing codec buffer size to detect the MCB for each codec. Fig. 2 shows the experiments results with clearly detected MCB. However, when the codec buffer size is further increasing after MCB, the MOS value decreases due to the extra delay incurred at the expanding codec buffer.

Table III: Minimum Codec Buffer and MOS

CODEC	$< BelowMCB, MOS >$	$< MCB, MOS >$
G.711	$< 8, 1.06 >$	$< 9, 4.35 >$
G.723.1	$< 30, 0.99 >$	$< 31, 3.59 >$
G.729A	$< 10, 1.05 >$	$< 11, 3.98 >$
GSM	$< 20, 1.79 >$	$< 21, 4.33 >$

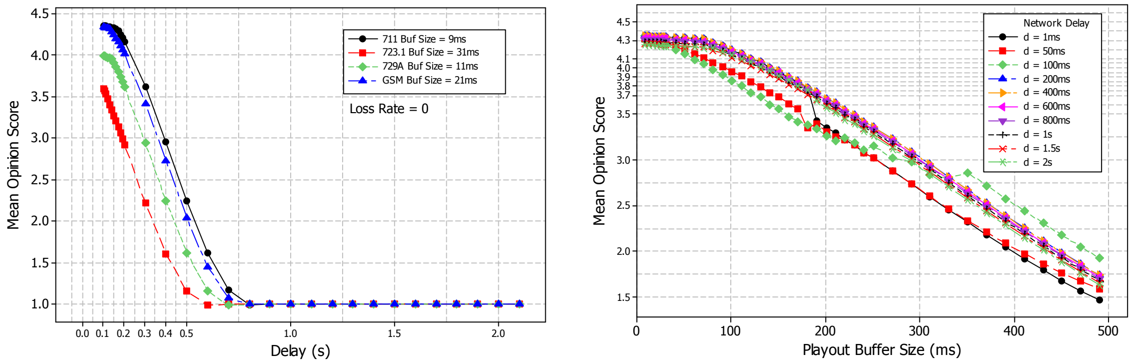
We summarize the Minimum Codec Buffer (MCB) of four investigated codec and their corresponding MOS values in an ideal network condition in Table III. The second column shows when codec buffer cannot reach MCB (only $1ms$



(a) The Impact of Loss on MOS with Minimum Playout Buffer Size.

(b) The Impact of Constant Loss Rate on MOS with Varying Playout Buffer Size (codec = G.711).

Figure 3: The Impact of Loss



(a) The Impact of Delay on MOS with Minimum Playout Buffer Size.

(b) The Impact of Constant Delay on MOS with Varying Playout Buffer Size (codec = G.711).

Figure 4: The Impact of Constant Delay

less), the corresponding MOS value is significantly low (e.g., 1.06 for G.711). In contrast as shown in the third column, when the codec buffer size is set to MCB (e.g., $9ms$ for G.711), the MOS value reaches its maximum (e.g., 4.35 for G.711) when the network is in an ideal condition. In our study, we use time delay to measure buffer size.

B. The Impact of Network Loss

Network loss can significantly degrade user satisfaction on received VoIP data. We conducted a variety of experiments and use the measured MOS values to find the user tolerable boundary impacted by various network losses. In these experiments, we choose four codecs: G.711, G.723.1, G.729A and GSM with their codec buffer sizes set to their specific MCB as in Table III. We control the network loss rates varying from 0% to 100%.

Fig. 3(a) depicts how network loss could seriously degrade user satisfaction in a VoIP application no matter which codec is used. For example, for GSM codec, when the network loss rate increases to 15% or beyond, most users cannot tolerate the perceived voice quality, which is indicated by the boundary MOS value 3.5. Similarly, the user tolerable boundaries for network loss when using G.711, G.723.1 and G.729A are 9%, 13% and 7%, respectively.

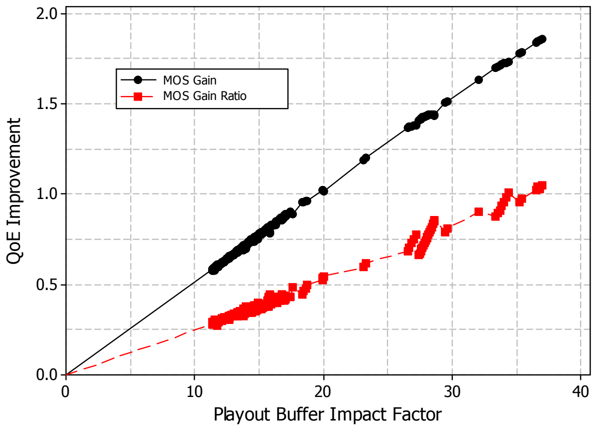
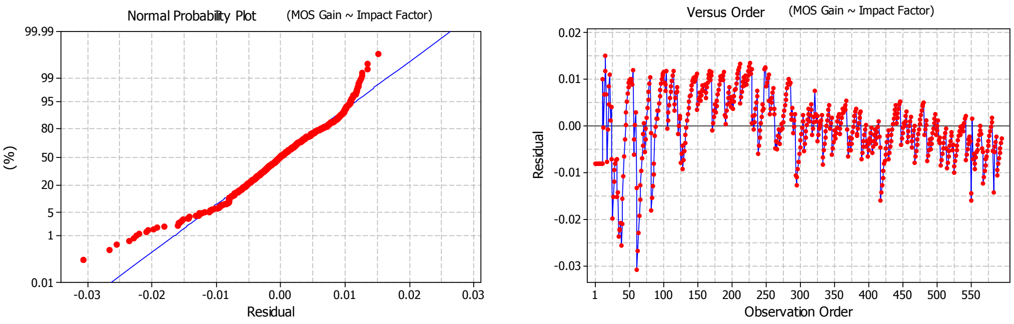


Figure 5: Validity of IF^{QoE}.

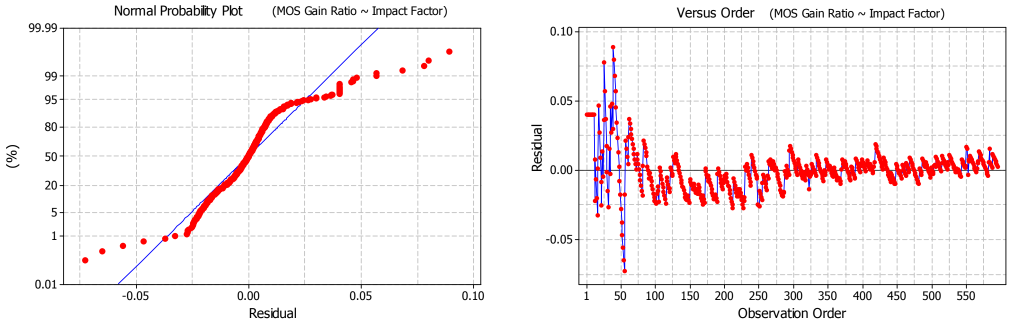
We further verify if a VoIP application with playout



(a) Normal Probability Plot of MOS Gain.

(b) Residual versus Order of MOS Gain.

Figure 6: Residual Analysis for MOS Gain $\sim IF^{QoE}$



(a) Normal Probability Plot of MOS Gain Ratio.

(b) Residual versus Order of MOS Gain Ratio.

Figure 7: Residual Analysis for MOS Gain Ratio $\sim IF^{QoE}$

buffer can have any positive impact on degraded user satisfaction due to network loss. For this purpose, we control the network loss rate increased from 0% up to 50%, and vary playout buffer size from $0ms$ to $500ms$. We use the measured MOS values to analyze the impact of playout buffer. The experiment results are shown in Fig. 3(b), which clearly confirms that playout buffer cannot improve the user satisfaction on received voice quality impaired by network loss, and even worse, it may further degrade user satisfaction due to the unnecessary playout buffer delay.

C. The Impact of Constant Network Delay

In this section, we continue our study on measuring QoE in another dimension: network delay. Similarly, we conducted experiments and use the measured MOS values to analyze the user satisfaction tolerable boundary impacted by different constant network delays. In these experiments, network loss rate is set to 0. We choose the same codecs with their codec buffer sizes set to their specific MCB. We vary network delays from $0ms$ to $2{,}000ms$.

Fig. 4(a) depicts how constant network delays could seriously degrade user satisfaction in a VoIP application for all selected codecs. For example, for G.711, when the constant network delay increases up to $350ms$ or more, most users cannot tolerate the perceived voice quality, which is again indicated by the MOS value 3.5. Similarly, the user satisfaction tolerable boundaries due to different constant network delays when using G.723.1, G.729A and GSM are $100ms$, $250ms$ and $300ms$, respectively.

We also verify if a playout buffer can help in such situation. For this purpose, we set network delay in each experiment to a constant value, and increase it from $1ms$ up to $2{,}000ms$, and vary playout buffer size from $0ms$ to $500ms$. The experiment results are shown in Fig. 4(b), which clearly confirms that playout buffer cannot improve the user satisfaction impaired by constant network delays, and even worse as the previous case, it may further degrade user satisfaction due to unnecessary playout buffer delay.

VI. IF^{QoE} Validation

In this section, we validate and analyze the accuracy of IF^{QoE} in evaluating the effectiveness on improving QoE of a playout buffer.

We conducted similar experiments as we discussed in Section IV-D. In these experiments, the sender sends the same voice stream over the network with controlled delay distribution to two VoIP systems. The only difference between these two systems is that one has playout buffer (denoted as S_{buff}) and another one does not (denoted as S_{nobuff}). For each sampled value in each experiment run, we use the measured MOS values from both S_{nobuff} and S_{buff} to calculate MOS gain and MOS gain ratio.

Meanwhile, we derive the corresponding IF^{QoE} using $e_{nobuff}, e_{buff}, d_{nobuff}$ and d_{buff}.

The result is reported in Fig. 5, which indicates a strong linear correlation between IF^{QoE} and MOS_{gain}, as well as between IF^{QoE} and MOS_{ratio}. In order to be more specific, we denote QoE_{gain} and QoE_{ratio} as MOS_{gain} and MOS_{ratio}.

Simple linear regression shows us the following two linear correlation functions:

$$MOS_{gain} = 0.00800 + 0.0507 \times IF^{QoE} \tag{15}$$

$$MOS_{ratio} = -0.0403 + 0.0282 \times IF^{QoE} \tag{16}$$

The coefficients of determination or r^2 for the two linear regression functions $MOS_{gain}(IF^{QoE})$ and $MOS_{ratio}(IF^{QoE})$ are 99.9% and 98.8%, respectively, which clearly shows that IF^{QoE} is a valid performance metric in measuring the effectiveness on QoE improvement of playout buffer.

A. Residual Analysis

To illustrate the accuracy of IF^{QoE}, we show residual plots for both $MOS_{gain}(IF^{QoE})$ and $MOS_{ratio}(IF^{QoE})$ in Fig. 6 and Fig. 7, respectively. In both Fig. 6(a) and Fig. 7(a), the residuals close to zero and as moving farther away from zero fewer residuals appear, which prove that the condition of normality is clearly met for both regression functions. The randomness shown in Fig. 6(b) and Fig. 7(b) further confirms the fitness of the regression functions.

VII. Conclusion

By identifying QoE as a function along two dimensions of network loss and delay, we have shown how different network factors as well as playout buffer can affect QoE assessment. Then, we have proposed a new performance metric called Playout Buffer QoE Impact Factor or IF^{QoE} for evaluating the effectiveness of playout buffer on QoE improvement. IF^{QoE} can be calculated using directly measurable performance metrics, which can accurately represent the effectiveness of a playout buffer on both absolute and relative QoE improvement. IF^{QoE} is the first proposed method bridging QoE assessment, QoS measurement and the evaluation on the impact of playout buffer. Our future work will include applying IF^{QoE} to evaluate specific playout algorithms used in real wired and wireless (e.g., WiFi and WiMax) network environments.

References

[1] Sat, B., and Wah, B. W. (2009). Analyzing Voice Quality in Popular VoIP Applications. IEEE MultiMedia, vol. 16, no. 1, pp. 46-59.

[2] Zwar, E. J., and Munch, B. (2006). Voice Quality and Network Capacity Planning for VoIP.

[3] Narbutt, M., and Davis, M. (2005). Assessing the Quality of VoIP Transmission Affected by Playout Buffer Scheme. 4th International Conference on Measurement of Speech and Audio Quality, Prague, Czech Republic.

[4] International Telecommunications Union (1996). ITU-T P.800. Methods for Subjective Determination of Transmission Quality.

[5] International Telecommunications Union (2007). ITU-T P.862 Corrigendum. Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.

[6] Telecommunications Industry Association (2005). Telecommunications, IP Telephony Equipment and Voice Quality Recommendations for IP Telephony.

[7] Morris, M. G., Venkatesh, V., Davis, G. B., and Davis, F.D. (2003). User Acceptance of Information Technology: Toward a Unified View.

[8] Becvar, Z., Mach. P., and Bestak, R. (2009). Impact of Handover on VoIP Speech Quality in WiMAX Networks. Eighth International Conference on Networks, icn, pp.281-286, Gosier, Guadeloupe, France.

[9] ITU-T Recommendation G.107 (1998). The E-Model, a computational model for use in transmission planning.

[10] Robinson, P. and Yedwab, D. (2009). Voice and Video Application Performance Management in UC Deployments.

[11] OPNET Technologies: http://www.opnet.com/. Last accessed: 2/20/2012.

[12] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne (1994) Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks. In Proceeding of IEEE INFOCOM.

[13] Gierlich, H. W. and Kettler, F. (2006). Advanced speech quality testing of modern telecommunication equipment: an overview. Signal Processing, 86(6), 1327 - 1340.

[14] Wang, S., Mai, Z., Xuan, D., and Zhao, W. (2006). Design and Implementation of QoS-Provisioning System for Voice over IP. IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 3, pp. 276-288.

[15] Myakotnykh, E. S. and Thompson, R. A. (2009). Adaptive Speech Quality Management in Voice-over-IP Communications. Fifth Advanced International Conference on Telecommunications, aict, pp.64-71, Venice/Mestre, Italy.

[16] Raja, A., Azad, R. M. A., and Flanagan, C. (2008). VoIP Speech Quality Estimation in a Mixed Context with Genetic Programming. 10th Annual Conference on Genetic and evolutionary computation, Atlanta, Georgia, United States.

[17] Zha, W. and Chan, W. (2005). Objective Speech Quality Measurement Using Statistical Data Mining. EURASIP Journal on Applied Signal Processing, no. 9, 1410-1424.

[18] Gardner, M., Frost, V.S. and Petr, D.W. (2003). Using optimization to achieve efficient quality of service in Voice over IP networks. IEEE International Performance, Computing, and Communications Conference, Phoenix, Arizona, United States.

[19] Mazurczyk, W. and Kotulski, Z. (2007). Adaptive VoIP with Audio Watermarking for Improved Call Quality and Security. Journal of Information Assurance and Security 2, 226-234.

[20] Wu, C., Chen, K., Huang, C., and Lei, C. (2009). An Empirical Evaluation of VoIP Playout Buffer Dimensioning in Skype, Google Talk and MSN Messenger. Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital and Video, Williamsburg, VA, United States.

[21] Narbutt, M. and Davis, M. (2005). Assessing the Quality of VoIP Transmission Affected by Playout Buffer Scheme.

4th International Conference on Measurement of Speech and Audio Quality, Prague, Czech Republic.

[22] Mohamed, S., Cervantes-Perez, F., and Afifi, H. Integrating Network Measurements and Speech Quality Subjective Scores for Control Purposes. IEEE Infocom, Anchorage, Alaska, (2001)

[23] W. Montgomery. Techniques for Packet Voice Synchronization. IEEE Journal on Selected Areas in Communications, Sol. SAC-6, No. 1 (Dec. 1983), pp. 1022 - 1028.

[24] Cole, R.G. and J. Rosenbluth, Voice Over IP Performance Monitoring, Journal of Computer Communications Review, vol. 4, no. 3, April (2001).

[25] O. Obafemi, T. Gyires and Y. Tang. An Analytic and Experimental Study on the Impact of Jitter Playout Buffer on the E-model in VoIP Quality Measurement. The 10th International Conference on Networks, 2011

Design of Half-Band FIR Filters for Signal Compression

Pavel Zahradnik, Boris Šimák and Michal Kopp
Department of Telecommunication Engineering
Czech Technical University in Prague
Prague, Czech Republic
zahradni, simak, koppmich@fel.cvut.cz

Miroslav Vlček
Department of Applied Mathematics
Czech Technical University in Prague
Prague, Czech Republic
vlcek@fd.cvut.cz

***Abstract*—An efficient design of equiripple half-band FIR filters for signal compression is presented. Solution of the approximation problem in terms of generating function and zero phase transfer function for the equiripple half-band FIR filter is shown. The equiripple half-band FIR filters are optimal in the Chebyshev sense. The closed form solution provides an efficient computation of the impulse response of the filter. Two examples are included. The robustness of the design is emphasized. The Matlab code of the design procedure is included.**

***Keywords*-*FIR filter; half-band filter; equiripple approximation; signal compression.*

I. Introduction

Half-band (HB) filter is a fundamental building block in multirate signal processing [2]. HB filters are used among others in filter banks and in image compression techniques, where the signal is iteratively decomposed using filtering and downsampling into its lower and higher subbands. This procedure is found, e.g., in the JPEG2000 compression [3]. Finite impulse response (FIR) filters are preferred because of their linear phase which is essential in the digital image processing. The equiripple (ER) filters are attractive because of their optimality in terms of the filter degree for the specified filter selectivity. Hence, the ER HB FIR filters are appreciated in these tasks. There is a numerical method for designing of ER HB FIR filters available. It is based on the numerical McClellan - Parks program [4]. It is usually combined with a clever ”Trick” [5]. The analytical design procedure [6] for trivial lowest order ($n \leq 2$) ER HB FIR filters has limited practical value. Besides this, some non-numerical design methods are available for almost ER HB FIR filters, e.g., [7] and [8]. In [1] and [9], we have presented a general non-numerical method for the design of ER HB FIR filters. Here we are focused on this method in more detailed manner. We are primarily concerned with the ER approximation of HB FIR filters and with the related non-numerical design procedure suitable for practical design of ER HB FIR filters. We present the generating function and the zero phase transfer function of the ER HB FIR filter. These functions give an insight into the nature of this approximation problem. Our design procedure is based on Chebyshev polynomials of the second kind [10]. Based on the differential equation for the Chebyshev polynomials of the second kind, we have derived formulas for an effective evaluation of the coefficients of the impulse response. We present an approximating degree equation which is useful in practical filter design. The advantage of the proposed approach over the numerical design procedures consists in the fact that the coefficients of the impulse response are evaluated by formulas. Hence, the design procedure and its speed is deterministic.

The structure of the paper is as follows. After an introduction of the basic terminology in Section II, we present the generating polynomial and the zero phase transfer function of an ER HB FIR filter in Section III. The differential equation for the generating polynomial and the impulse response of an ER HB FIR filter are presented in Section IV. Sections V and VI introduce the degree equation of an ER HB FIR filter and its secondary values. The design procedure is summarizes step by step in Section VII. It is followed by two examples in Section VIII. Section IX emphasizes the robustness of the presented design procedure. Appendix I summarizes the derivation of the algebraic procedure for the evaluation of the impulse response of the filter. In Appendix II, the Matlab code of the design procedure is presented.

II. Impulse Response, Transfer Function and Zero Phase Transfer Function

A HB filter is specified by the minimal passband frequency $\omega_p T$ (or maximal stopband frequency $\omega_s T$) and by the minimal attenuation in the stopband a_s [dB] (or maximal attenuation in the passband a_p [dB]). The antisymmetric behavior of its frequency response implies the relations $\omega_s T = \pi - \omega_p T$ and $10^{0.05 a_p} + 10^{0.05 a_s} = 1$. The goal in the filter design is to get the minimum filter length N satisfying the filter specification and to evaluate the coefficients of the impulse response of the filter. We assume the impulse response $h(k)$ with odd length $N = 2(2n+1)+1$ coefficients and with even symmetry $h(k) = h(N-1-k)$. The impulse response of the HB FIR filter with the length $N = 2(2n+1)+1$ contains $2n$ zero coefficients as follows

$$\begin{aligned} h(2n+1) &= a(0) = 0.5 \qquad (1) \\ 2h(2n+1 \pm 2k) &= a(2k) = 0 \ , \ k = 1 \dots n \end{aligned}$$

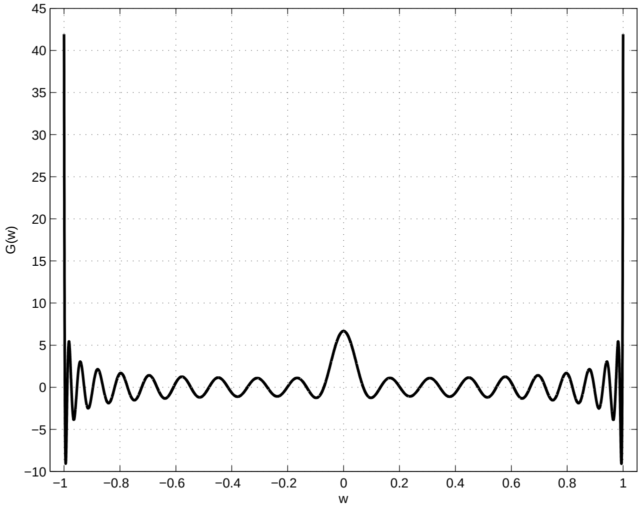


Fig. 1. Generating polynomial $G(w)$ for $n = 20$, $\kappa' = 0.03922835$, $A = 1.08532371$ and $B = 0.95360863$.

$$2h(2n+1 \pm (2k+1)) \quad = \quad a(2k+1) \ \ , \ k = 0 \ldots n \ .$$

The transfer function of the HB FIR filter is

$$H(z) = z^{-(2n+1)} \left[\frac{1}{2} + \sum_{k=0}^{n} a(2k+1)\, T_{2k+1}(w) \right] \qquad (2)$$

where

$$T_n(w) = \cos(n \arccos(w)) \qquad (3)$$

is Chebyshev polynomial of the first kind. The frequency response $H(e^{j\omega T})$ of the HB FIR filter is

$$H(e^{j\omega T}) = e^{-j(2n+1)\omega T} \ Q(\cos \omega T) \qquad (4)$$

where $Q(w)$ is a polynomial in the variable $w = (z + z^{-1})/2$ which on the unit circle reduces to a real valued zero phase transfer function $Q(w)$ of the real argument $w = \cos(\omega T)$.

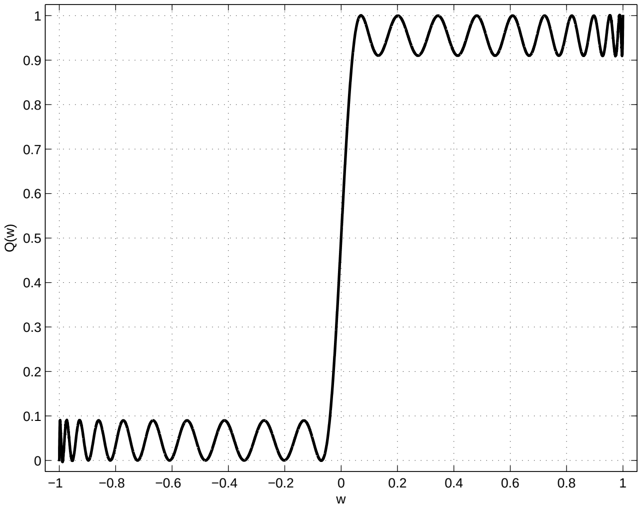


Fig. 2. Zero phase transfer function $Q(w)$ for $n = 20$, $\kappa' = 0.03922835$, $A = 1.08532371$, $B = 0.95360863$ (cf. Fig. 1) and $\mathcal{N} = 0.55091994$.

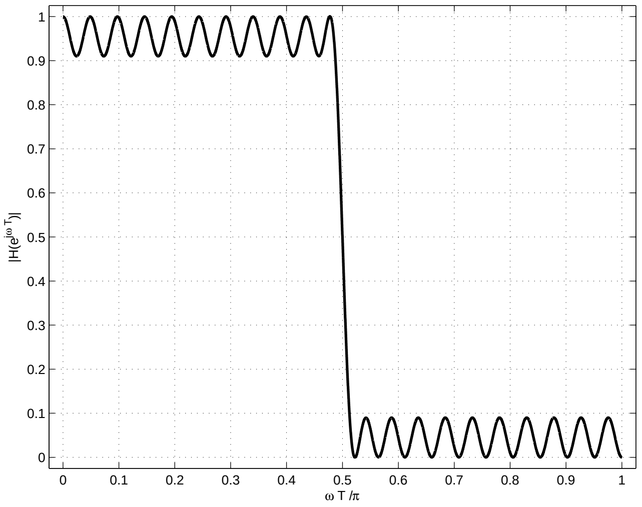


Fig. 3. Amplitude frequency response $|H(e^{j\omega T})|$ corresponding to the zero phase transfer function $Q(w)$ from Fig. 2.

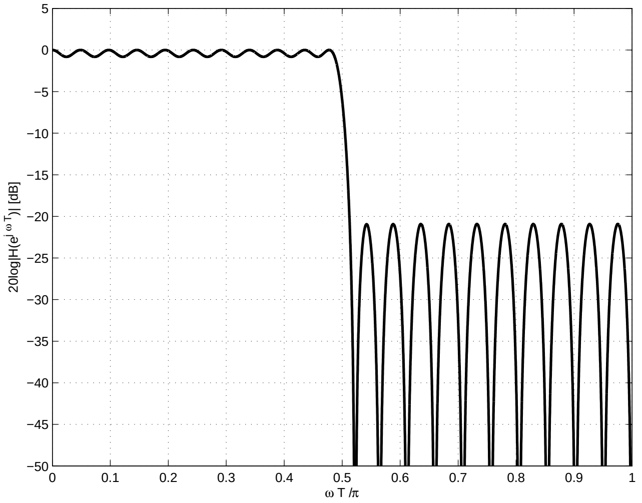


Fig. 4. Amplitude frequency response $20 \log |H(e^{j\omega T})|$ corresponding to the zero phase transfer function $Q(w)$ from Fig. 2.

III. Generating Polynomial and Zero Phase Transfer Function of an ER HB FIR Filter

A straightforward theory for the generating polynomial of an ER HB FIR filter is currently not available. The generating polynomial of an ER HB FIR filter is related to the generating polynomial of the almost ER HB FIR filter presented in [8]. Based on our experiments conducted in [8], we have found that the generating polynomial $G(w)$ (Fig. 1) of the ER HB FIR filter is obtained by weighting of Chebyshev polynomials in the generating polynomial of the AER HB FIR filter, namely

$$G(w) = AU_n\left(\frac{2w^2 - 1 - \kappa'^2}{1 - \kappa'^2}\right) + BU_{n-1}\left(\frac{2w^2 - 1 - \kappa'^2}{1 - \kappa'^2}\right) \qquad (5)$$

where

$$U_n(x) = \frac{\sin\left[(n+1)\arccos(x)\right]}{\sin\left[\arccos(x)\right]} \qquad (6)$$

is Chebyshev polynomial of the second kind and A, B, κ' are real numbers. The zero phase transfer function $Q(w)$ (Fig. 2) of the ER HB FIR is related to the generating polynomial

$$Q(w) = \frac{1}{2} + \frac{1}{\mathcal{N}} \int G(w) dw \tag{7}$$

where the norming factor $\mathcal{N}$ is given by (19). Both the generating polynomial $G(w)$ and the zero phase transfer function $Q(w)$ show the nature of the approximation of an ER HB FIR filter.

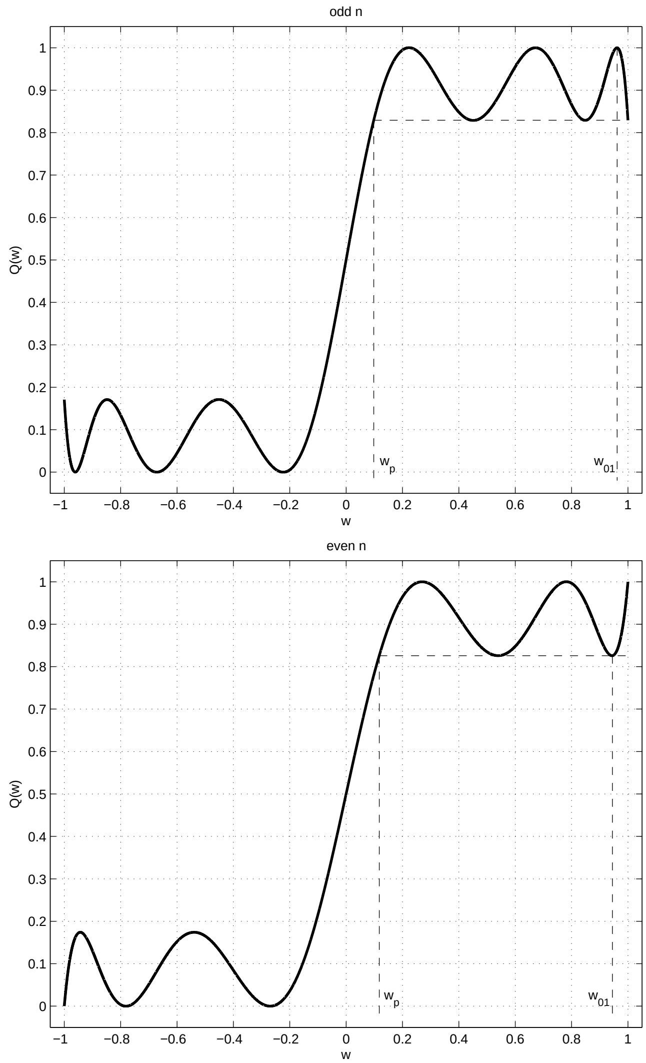


Fig. 5. $Q(w)$ for odd and even n.

IV. Differential Equation and Impulse Response of an ER HB FIR Filter

The Chebyshev polynomial of the second kind $U_x(w)$ fulfils the differential equation

$$(1-x^2)\frac{d^2U_n(x)}{dx^2} - 3x\frac{dU_n(x)}{dx} + n(n+2)U_n(x) = 0 \ . \tag{8}$$

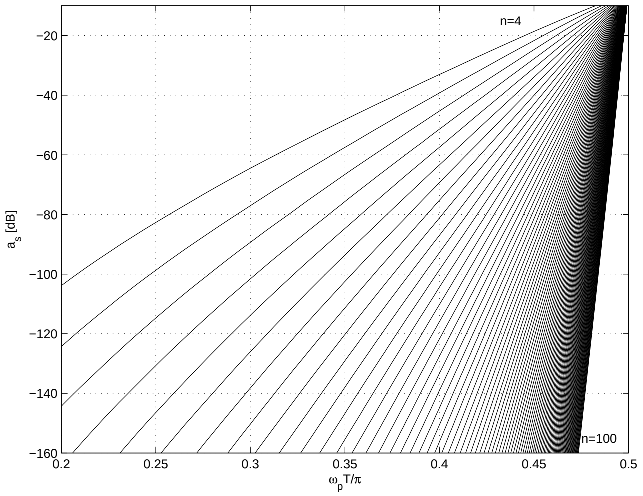


Fig. 6. Empirical dependence of a_s[dB] on $\omega_p T/\pi$ and n.

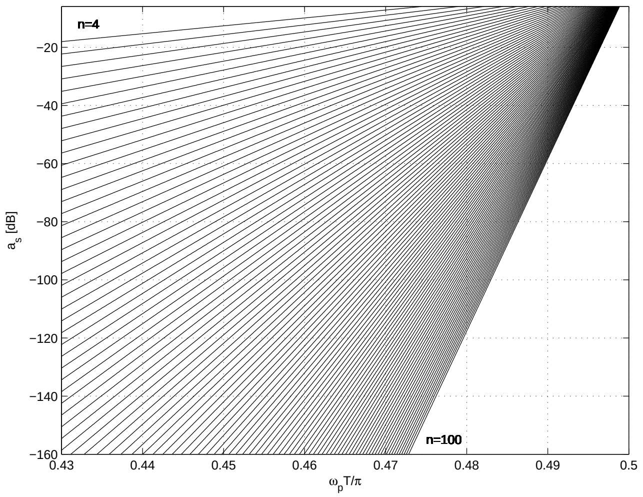


Fig. 7. Detailed view of Fig. 6 near $\omega_p T = 0.5\pi$.

Using substitution

$$x = \left(\frac{2w^2 - 1 - \kappa'^2}{1-\kappa'^2}\right) \tag{9}$$

we get the differential equation (8) in the form

$$\begin{aligned} w(w^2-\kappa'^2)&\left[(1-w^2)\frac{d^2U_n(w)}{dw^2} - 3w\frac{dU_n(w)}{dw}\right] \\ &+ \left[\kappa'^2(1-w^2) + 2w^2(1-w^2)\right]\frac{dU_n(w)}{dw} \\ &+ 4w^3 n(n+2)U_n(w) = 0 \ . \end{aligned} \tag{10}$$

Based on the differential equation (10), we have derived the non-numerical procedure for the evaluation of the impulse response $h_n(k)$ corresponding to polynomial $\mathcal{U}_n(w)$

$$\mathcal{U}_n(w) = \int U_n\left(\frac{2w^2-1-\kappa'^2}{1-\kappa'^2}\right) dw \ . \tag{11}$$

This procedure is summarized in Tab. I. The principle of its derivation is shown in the Appendix I. The impulse response

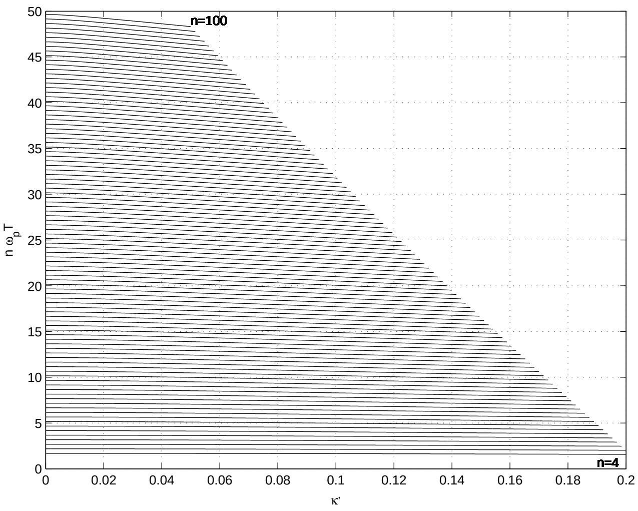


Fig. 8. Empirical dependence of $n\,\omega_p T$ on κ'.

$h(k)$ of the ER HB FIR filter is

$$h(k) = \frac{1}{2} + \frac{A}{\mathcal{N}} h_n(k) + \frac{B}{\mathcal{N}} h_{n-1}(k) \qquad (12)$$

where the real norming factor $\mathcal{N}$ is given by (19). The non-numerical evaluation of the impulse response $h(k)$ is essential in the practical filter design because of its determinism.

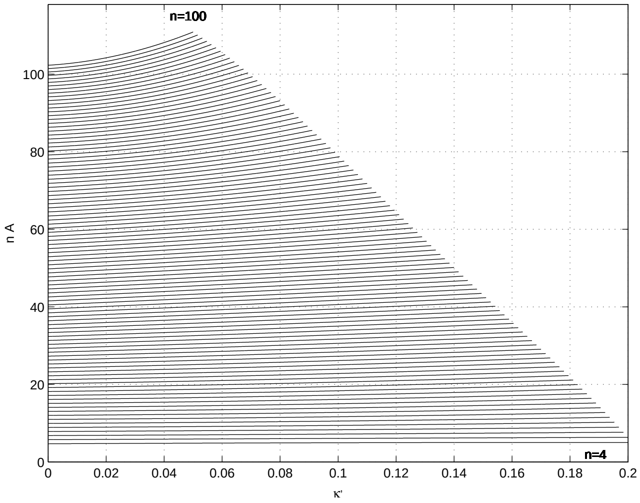


Fig. 9. Empirical dependence of $n\,A$ on κ'.

V. Degree of an ER HB FIR Filter

The exact degree formula is not available. In the practical filter design, the degree n can be obtained with excellent accuracy from the specified minimal passband frequency $\omega_p T$ and from the minimal attenuation in the stopband a_s [dB] using the approximating degree formula

$$n \doteq \frac{a_s[dB] - 18.18840664\,\omega_p T + 33.64775300}{18.54155181\,\omega_p T - 29.13196871} \quad . \qquad (13)$$

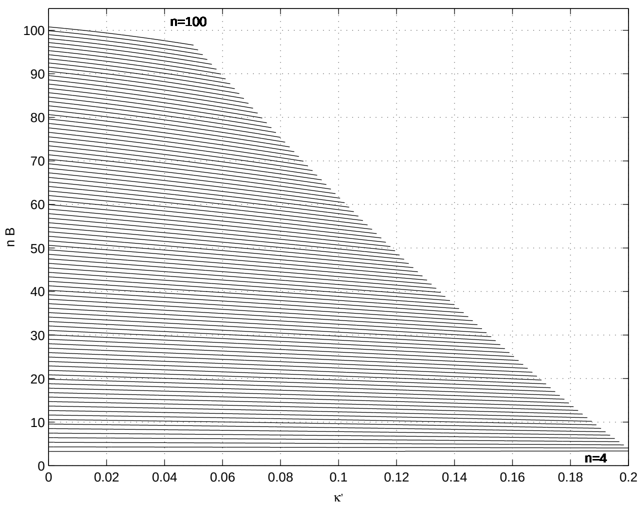


Fig. 10. Empirical dependence of $n\,B$ on κ'.

The exact relation between the minimal attenuation in the stopband a_s [dB], the minimal passband frequency $\omega_p T$ and the degree n were obtained experimentally. It is shown in Fig. 6 and Fig. 7. Equation (13) was obtained by the approximation of exact experimental values in Fig. 7. The approximating degree formula (13) is very precise. However, for very low values $\omega_p T$ its precision slightly decreases. In order to demonstrate the negligible inaccuracy for very low values $\omega_p T$, let us assume an ER HB FIR filter specified by by the minimal passband frequency $\omega_p T = 0.25\pi$ and by the minimal attenuation in the stopband $a_s = -80$ dB. The approximating degree formula (13) results in $n = 4.16194938$ while the filter specification is met for $n = 4$, cf. Fig. 6.

VI. Secondary Values of the ER HB FIR Filter

The secondary real values κ', A and B can be obtained from the specified passband frequency $\omega_p T$ and from the degree n of the generating polynomial. The approximating formulas

$$\kappa' = \frac{n\omega_p T - 1.57111377\,n + 0.00665857}{-1.01927560\,n + 0.37221484} \qquad (14)$$

$$A = \left(0.01525753\,n + 0.03682344 + \frac{9.24760314}{n}\right)\kappa' + 1.01701407 + \frac{0.73512298}{n} \qquad (15)$$

and

$$B = \left(0.00233667\,n - 1.35418408 + \frac{5.75145813}{n}\right)\kappa' + 1.02999650 - \frac{0.72759508}{n} \qquad (16)$$

are obtained by the approximation of experimental values summarized in graphs in Fig. 8 - Fig. 10. The approximating formulas provide a very good accuracy useful in practical filter design. If desired, the exact values κ', A and B can be

TABLE I

ALGORITHM FOR THE EVALUATION OF THE COEFFICIENTS $h_n(k)$.

```
given                      $n$ (integer value),  $\kappa'$ (real value)
initialization             $\alpha(2n) = \frac{1}{(1-\kappa'^2)^n}$
                           $\alpha(2n-2) = -(2n\kappa'^2+1)\,\alpha(2n)$
                           $\alpha(2n-4) = -\frac{4n+1+(n-1)(2n-1)\kappa'^2}{2n}\,\alpha(2n-2) - \frac{(2n+1)(n+1)\kappa'^2}{2n}\,\alpha(2n)$
body
  (for $k=n$ down to 3)
                           $\alpha(2k-6) =$
                              $\{\ -\left[3(n(n+2)-k(k-2))+2k-3+2(k-2)(2k-3)\kappa'^2\right]\alpha(2k-4)$
                                $-\left[3(n(n+2)-(k-1)(k+1))+2(2k-1)+2k(2k-1)\kappa'^2\right]\alpha(2k-2)$
                                $-\left[n(n+2)-(k-1)(k+1)\right]\alpha(2k)\ \ \}\ /\ \left[n(n+2)-(k-3)(k-1)\right]$
      (end loop on $k$)
integration
  (for $k=0$ to $n$)       $a(2k+1) = \frac{\alpha(2k)}{2k+1}$                    (end loop on $k$)
impulse response $h_n(k)$
                           $h_n(2n+1) = 0$
  (for $k=0$ to $n$)       $h_n(2n+1\pm(2k+1)) = \frac{a(2k+1)}{2}$              (end loop on $k$)
```

obtained from (7) numerically (e.g. using the Matlab function $fminsearch$) by satisfying the equality (see Fig. 5)

$$Q(w_p) = \begin{cases} Q(1) & \text{if n is odd} \\ Q(w_{01}) & \text{if n is even .} \end{cases} \tag{17}$$

The position of the local extremal value w_{01} (Fig. 5)

$$w_{01} = \sqrt{\kappa'^2 + (1-\kappa'^2)\cos^2\frac{\pi}{2n+1}} \tag{18}$$

was introduced in [8]. The relation (17) guarantees the equiriple behaviour of the zero phase transfer function $Q(w)$.

VII. DESIGN OF THE ER HB FIR FILTER

The design procedure is as follows:

- Specify the ER HB FIR filter by the minimal passband frequency $\omega_p T$ and by the minimal attenuation in the stopband a_s [dB].
- Calculate the integer degree n of the generating polynomial (13).
- Calculate the real values κ' (14), A (15) and B (16).
- Evaluate the partial impulse responses $h_n(k)$ and $h_{n-1}(k)$ (Tab. I).
- Evaluate the final impulse response $h(k)$ (12) where the real norming factor $\mathcal{N}$ is

$$\mathcal{N} = \begin{cases} 2\,[\,A\mathcal{U}_n(1) + \mathrm{B}\mathcal{U}_{n-1}(1)\,] & \text{if n is even} \\ 2\,[\,A\mathcal{U}_n(w_{01}) + \mathrm{B}\mathcal{U}_{n-1}(w_{01})\,] & \text{if n is odd .} \end{cases} \tag{19}$$

The Matlab source code of the design procedure is summarized in Appendix II.

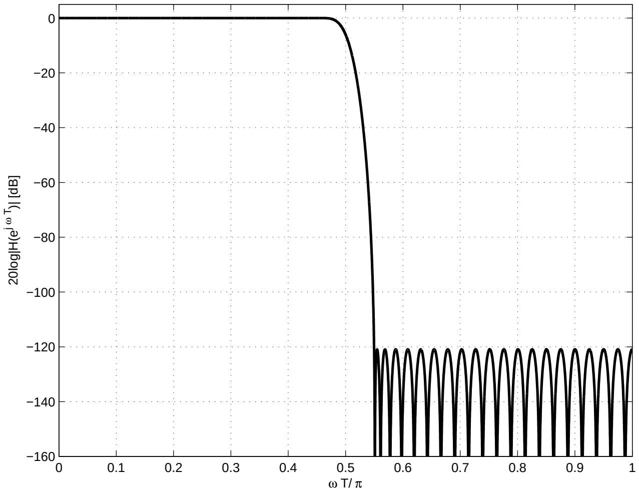


Fig. 11. Amplitude frequency response $20\log|H(e^{j\omega T})|$ [dB].

VIII. EXAMPLES OF THE DESIGN

Example 1.

Design an ER HB FIR filter specified by the minimal passband frequency $\omega_p T = 0.45\pi$ and by the minimal attenuation in the stopband $a_s = -120$ dB.

Using formulas we get $n = 38.3856 \rightarrow 39$ (13), $\kappa' = 0.15571103$ (14), $A = 1.17117396$ (15), $B = 0.83763199$ (16) and $\mathcal{N} = -2747.96038544$ (19). The impulse response $h(k)$ (Tab. II) with the length $N = 159$ coefficients is evaluated using algorithm summarized in Tab. I and eq. (12). The actual values $\omega_{p\ act}T = 0.4502\pi$ and $a_{act} = -120.91$ dB satisfy the filter specification. The

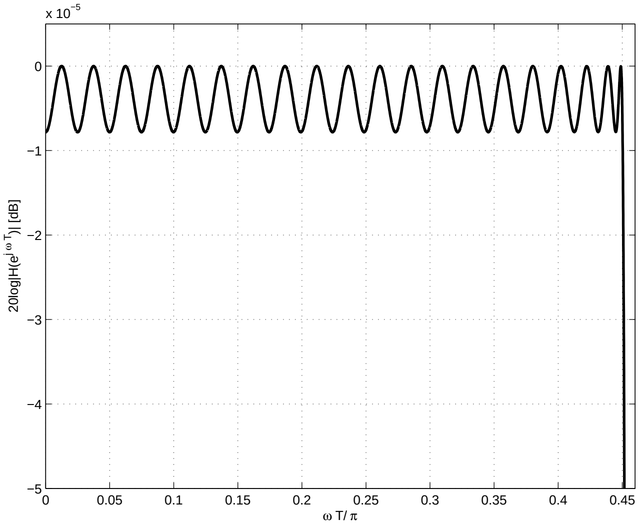


Fig. 12. Passband of the filter.

TABLE II
COEFFICIENTS OF THE IMPULSE RESPONSE.

k	h(k)	k	h(k)
0 , 158	-0.00000070	42 , 116	0.00231877
2 , 156	0.00000158	44 , 114	-0.00283354
4 , 154	-0.00000331	46 , 112	0.00344038
6 , 152	0.00000622	48 , 110	-0.00415347
8 , 150	-0.00001087	50 , 108	0.00498985
10 , 148	0.00001799	52 , 106	-0.00597048
12 , 146	-0.00002852	54 , 104	0.00712193
14 , 144	0.00004363	56 , 102	-0.00847897
16 , 142	-0.00006481	58 , 100	0.01008867
18 , 140	0.00009384	60 , 98	-0.01201717
20 , 138	-0.00013287	62 , 96	0.01436125
22 , 136	0.00018446	64 , 94	-0.01726924
24 , 134	-0.00025161	66 , 92	0.02098117
26 , 132	0.00033779	68 , 90	-0.02591284
28 , 130	-0.00044697	70 , 88	0.03285186
30 , 128	0.00058370	72 , 86	-0.04348979
32 , 126	-0.00075311	74 , 84	0.06223123
34 , 124	0.00096097	76 , 82	-0.10523903
36 , 122	-0.00121375	78 , 80	0.31802058
38 , 120	0.00151871	79	0.50000000
40 , 118	-0.00188398		

amplitude frequency response 20log$|H(e^{j\omega T})|$ [dB] of the filter is shown in Fig. 11. The detailed view of its passband is shown in Fig. 12.

Example 2.
Design an ER HB FIR filter specified by the minimal passband frequency $\omega_p T = 0.475\pi$ and by the minimal attenuation in the stopband $a_s = -80$ dB.
We get $n = 50.2277 \rightarrow 51$ (13), $\kappa' = 0.07779493$ (14), $A = 1.10893402$ (15), $B = 0.92842531$ (16) and $\mathcal{N} = -226.43793850$ (19). The impulse response $h(k)$ (Tab. III) with the length $N = 207$ coefficients is evaluated using algorithm summarized in Tab. I and eq. (12). The actual values $\omega_{p\ act} T = 0.4752\pi$ and $a_{act} = -81.23$ dB satisfy the filter specification. The amplitude frequency response 20log$|H(e^{j\omega T})|$ [dB] of the filter is shown in Fig. 13. The detailed view of its passband is shown in Fig. 14.

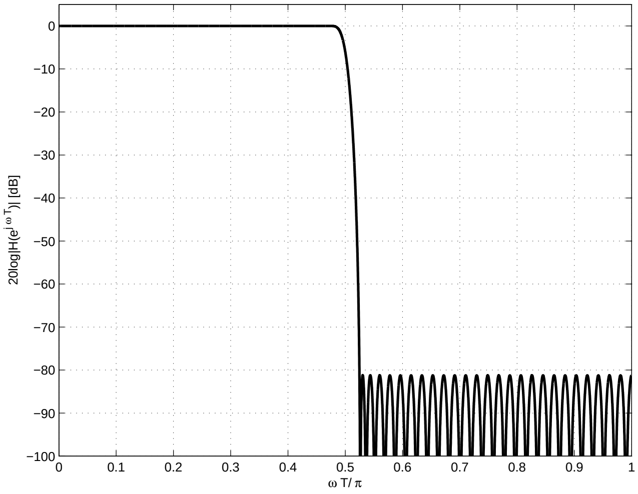


Fig. 13. Amplitude frequency response $20\log|H(e^{j\omega T})|$ [dB].

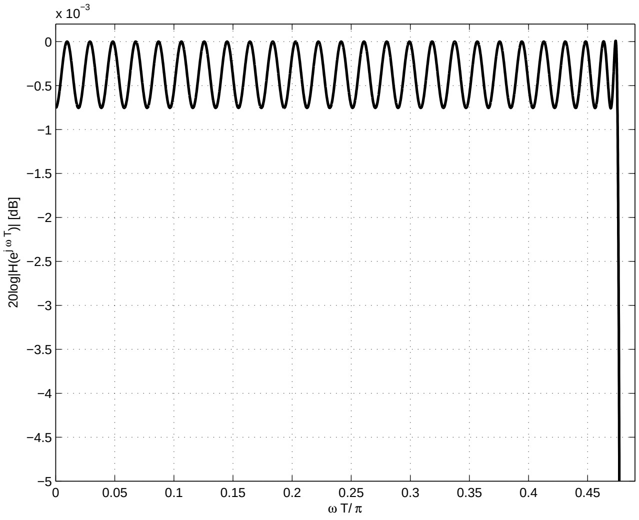


Fig. 14. Passband of the filter.

IX. ROBUSTNESS OF THE DESIGN

The recursive evaluation of the impulse response summarized in Tab. I is extremely fast and robust. Using the proposed procedure it is possible to evaluate the impulse response of the ER HB FIR filter with the length of thousands of coefficients within a fraction of a second. In order to demonstrate this robustness, let us design an ER HB FIR filter with strict specification, namely with the specified minimal passband frequency $\omega_p T = 0.495\pi$ and minimal attenuation in the stopband $a_s = -180$ dB. The length of the designed filter is $N = 2347$ coefficients. The amplitude frequency response $|H(e^{j\omega T})|$ of the filter is shown in Fig. 15.

X. CONCLUSIONS

The presented design procedure is useful in the design of equiripple halfband FIR filters. The designed filters are optimal in Chebyshev sense. The generating polynomial and

TABLE III
COEFFICIENTS OF THE IMPULSE RESPONSE.

k	h(k)	k	h(k)
0 , 206	-0.00003240	54 , 152	0.00279585
2 , 204	0.00002594	56 , 150	-0.00312880
4 , 202	-0.00003613	58 , 148	0.00349527
6 , 200	0.00004879	60 , 146	-0.00389883
8 , 198	-0.00006433	62 , 144	0.00434368
10 , 196	0.00008318	64 , 142	-0.00483480
12 , 194	-0.00010580	66 , 140	0.00537821
14 , 192	0.00013270	68 , 138	-0.00598122
16 , 190	-0.00016445	70 , 136	0.00665292
18 , 188	0.00020162	72 , 134	-0.00740467
20 , 186	-0.00024486	74 , 132	0.00825098
22 , 184	0.00029484	76 , 130	-0.00921066
24 , 182	-0.00035231	78 , 128	0.01030857
26 , 180	0.00041804	80 , 126	-0.01157828
28 , 178	-0.00049287	82 , 124	0.01306624
30 , 176	0.00057766	84 , 122	-0.01483857
32 , 174	-0.00067338	86 , 120	0.01699269
34 , 172	0.00078102	88 , 118	-0.01967820
36 , 170	-0.00090166	90 , 116	0.02313707
38 , 168	0.00103644	92 , 114	-0.02778764
40 , 166	-0.00118660	94 , 112	0.03442063
42 , 164	0.00135345	96 , 110	-0.04473116
44 , 162	-0.00153844	98 , 108	0.06312764
46 , 160	0.00174312	100 , 106	-0.10577606
48 , 158	-0.00196920	102 , 104	0.31817614
50 , 156	0.00221857	103	0.50000000
52 , 154	-0.00249333		

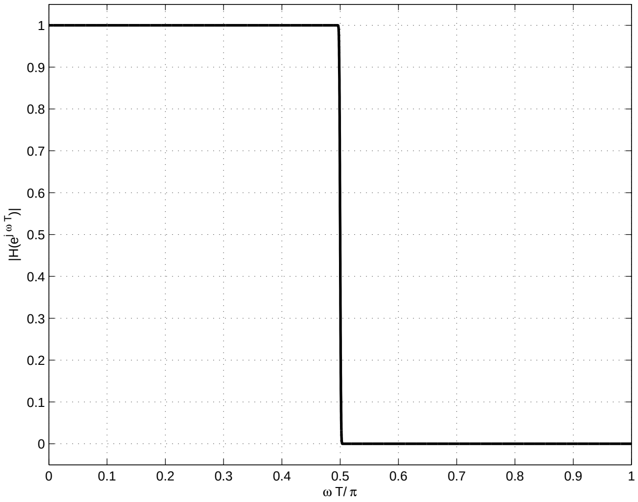


Fig. 15. Amplitude frequency response $|H(e^{j\omega T})|$.

the zero phase transfer function based on the Chebyshev polynomials of second kind illustrate the nature of the approximation problem. The degree formula indispensable for the filter design is presented. The strength of the proposed design method consists in the fact that the coefficients of the impulse response of the filter are straightforwardly evaluated using formulas from the filter specification. Because of its inherent determinism resulting from the non-numerical approach, the presented design procedure is useful in the adaptive digital signal processing as well. The enclosed Matlab source code is useful in the filter design.

ACKNOWLEDGEMENT

This activity was gratefully supported by the grant No. MSM6840770014, Ministry of Education, Czech Republic.

REFERENCES

[1] P. Zahradnik, B. Šimák, M. Vlček, Half-band FIR Filters for Signal Compression, *The Tenth International Conference on Networks ICN 2011*, January 2011, St. Maarten, pp. 359-362.

[2] N. J. Fliege, Multirate Digital Signal Processing, John Wiley and Sons, New York, 1994.

[3] D. S. Taubman, M. W. Marcellin JPEG2000 Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, 2002.

[4] J.H. McClellan, T. W. Parks, L. R. Rabiner, A Computer Program for Designing Optimum FIR Linear Phase Digital Filters, *IEEE Trans. Audio Electroacoust.*, Vol. AU - 21, Dec. 1973, pp. 506 - 526.

[5] P. P. Vaidyanathan, T. Q. Nguyen, A "TRICK" for the Design of FIR Half-Band Filters, *IEEE Transactions on Circuits and Systems*, Vol. CAS - 34, No. 3, March 1987, pp. 297 - 300.

[6] M. Lutovac, L. Milic, Design of Optimal Halfband FIR Filters with Minimum Phase, *Proc. XLVII ETRAN Conference*, Herceg Novi, June 8-13, 2003, Vol. I, pp. 209-212.

[7] P. N. Wilson, H. J. Orchard, A Design Method for Half-Band FIR Filters, *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. CAS - 46, No. 1, January 1999, pp. 95 - 101.

[8] P. Zahradnik, M. Vlček, R. Unbehauen, Almost Equiripple FIR Half-Band Filters, *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. CAS - 46, No. 6, June 1999, pp. 744 - 748.

[9] P. Zahradnik, M. Vlček, Equiripple Approximation of Half-Band FIR Filters, *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 56, No. 12, December 2009, pp. 941-945.

[10] M. Abramowitz, I. Stegun, *Handbook of Mathematical Function*, Dover Publication, New York Inc., 1972.

APPENDIX I

In the following considerations, the identities

$$2nU_{n-1}(w) = \frac{dU_n(w)}{dw} - \frac{dU_{n-2}(w)}{dw} \tag{20}$$

$$\frac{dU_{2n-1}(w)}{dw} = \sum_{k=0}^{n-1} 2(2k+1)U_{2k}(w) \tag{21}$$

$$\frac{dU_{2n}(w)}{dw} = \sum_{k=1}^{n} 2(2k)U_{2k-1}(w) \tag{22}$$

$$w\frac{dU_{2n-1}(w)}{dw} = \sum_{k=0}^{n-1}(2k+1)\left[U_{2k+1}(w) + U_{2k-1}(w)\right] \tag{23}$$

$$w\frac{dU_{2n}(w)}{dw} = \sum_{k=1}^{n} 2k\left[U_{2k}(w) + U_{2k-2}(w)\right] \tag{24}$$

are useful. For the three terms in the differential equation (10) we get the relations

$$\begin{aligned} w(w^2-\kappa'^2)\left[(1-w^2)\frac{d^2U_n(w)}{dw^2} - 3w\frac{dU_n(w)}{dw}\right] = \\ \sum_{k=0}^{n} -\alpha(2k)k(k+1)\frac{1}{2}\left[U_{2k+3}(w) + 3U_{2k+1}(w)\right. \\ \left. + 3U_{2k-1}(w) + U_{2k-3}(w)\right] \\ +\kappa'^2\alpha(2k)4k(k+1)\frac{1}{2}\left[U_{2k+1}(w) + U_{2k-1}(w)\right] \end{aligned} \tag{25}$$

$$\begin{aligned}
&\left[\kappa'^2(1-w^2)+2w^2(1-w^2)\right]\frac{dU_n(w)}{dw}= \\
&\quad\sum_{k=0}^{n}\alpha(2k)\kappa'^2\left[(k+1)U_{2k-1}(w)-kU_{2k+1}(w)\right] \\
&\quad+\alpha(2k)\frac{1}{2}\left[(k+1)(U_{2k+1}(w)+2U_{2k-1}(w)+U_{2k-3}(w))\right. \\
&\quad\left.-k(U_{2k+3}(w)+2U_{2k+1}(w)+U_{2k-1}(w))\right]
\end{aligned} \tag{26}$$

and

$$\begin{aligned}
&4w^3n(n+2)U_n(w)= \\
&\qquad\sum_{k=0}^{n}n(n+2)\alpha(2k)\frac{1}{2}\left[U_{2k+3}(w)+3U_{2k+1}\right. \\
&\qquad\qquad\left.+3U_{2k-1}+U_{2k-3}(w)\right].
\end{aligned} \tag{27}$$

By collecting and summing of the coefficients belonging to the particular degree of the Chebyshev polynomial we get

$$U_{2k+3}(w):\frac{1}{2}\left[n(n+2)-k(k+2)\right]\alpha(2k) \tag{28}$$

$$\begin{aligned}
U_{2k+1}(w):&\frac{1}{2}\left[3(n(n+2)-k(k+2))\right. \\
&\left.+2k+1+2k(2k+1)\kappa'^2\right]\alpha(2k)
\end{aligned} \tag{29}$$

$$\begin{aligned}
U_{2k-1}(w):&\frac{1}{2}\left[3(n(n+2)-k(k+2))+2(2k+1)\right. \\
&\left.+2(k+1)(2k+1)\kappa'^2\right]\alpha(2k)
\end{aligned} \tag{30}$$

$$U_{2k-3}(w):\frac{1}{2}\left[n(n+2)-(k-1)(k+1)\right]\alpha(2k)\ . \tag{31}$$

By manipulation of k in (28)-(30) we get

$$\frac{1}{2}\left[n(n+2)-(k-3)(k-1)\right]\alpha(2k-6) \tag{32}$$

$$\begin{aligned}
&\frac{1}{2}\left[3(n(n+2)-(k-2)k)+2k-3\right. \\
&\left.\qquad+2(k-2)(2k-3)\kappa'^2\right]\alpha(2k-4)
\end{aligned} \tag{33}$$

$$\begin{aligned}
&\frac{1}{2}\left[3(n(n+2)-(k-1)(k+1))+2(2k-1)\right. \\
&\left.\qquad+2k(2k-1)\kappa'^2\right]\alpha(2k-2)
\end{aligned} \tag{34}$$

$$\frac{1}{2}\left[n(n+2)-(k-1)(k+1)\right]\alpha(2k)\ . \tag{35}$$

The initial values $\alpha(2n)$, $\alpha(2n-2)$ and $\alpha(2n-4)$ follow from (28)-(31). For $k=n$ we get from (28) the relation

$$\left[n(n+2)-n(n+2)\right]\alpha(2n)=0 \tag{36}$$

which is fulfilled for arbitrary value $\alpha(2n)$. Let us choose the value (Tab. I)

$$\alpha(2n)=\frac{1}{(1-\kappa'^2)^n}\ . \tag{37}$$

For $k=n-1$ we get from (28) and (29) the relation

$$\begin{aligned}
&(n(n+2)-(n-1)(n+1))\alpha(2n-2) \\
&\qquad+(2n+1+2n(2n+1)\kappa'^2)\alpha(2n)=0
\end{aligned} \tag{38}$$

yielding the initial value $\alpha(2n-2)$ (Tab. I).

$$\alpha(2n-2)=-(2n\kappa'^2+1)\,\alpha(2n) \tag{39}$$

Finally, from (28)-(30) for $k=n-2$ we get

$$\begin{aligned}
&2n\alpha(2n-4) \\
&\qquad+(4n+1+(n-1)(2n-1)\kappa'^2)\alpha(2n-2) \\
&\qquad+(2n+1)((n+1)\kappa'^2+1)\alpha(2n)=0
\end{aligned} \tag{40}$$

which yields the initial value $\alpha(2n-4)$ (Tab. I). The formula for the evaluation of the remaining coefficients $\alpha(2k-6)$ (body in Tab. I) follows from (32)-(35).

APPENDIX II

```
% Design of Equiripple Half-Band FIR Filter
%
clear, clf reset,
adB=-80;                                % Specifications for Example No. 2
omp=0.475*pi;
A00=-33.64775299940740; A01=-29.13196870512581;
A10= 18.18840663850262; A11 =18.54155180910656;
N=(adB-A10*omp-A00)/(A11*omp+A01); N=ceil(N);
k00=-0.00665856769717; k01= 1.57111377495119;
k10= 0.37221483652163; k11=-1.01927559802890;
k=(N*omp-k01*N-k00)/(k11*N+k10);
a12=0.01525753184125;   a11=0.03682344002622;   a10=9.24760314166335;
a01=1.01701406973534;   a00=0.73512297663750;
b12=0.00233666682716;   b11=-1.35418408482371;  b10=5.75145813400838;
b01=1.02999650170704;   b00=-0.72759508233144;
AA=(a12*N*N+a11*N+a10)*k+a01*N+a00; BB=(b12*N*N+b11*N+b10)*k+b01*N+b00;
h1=h_uarg(N,k); h2=h_uarg(N-1,k); h = h1_plus_h2(AA*h1,BB*h2);
w01=sqrt(k*k+(1-k*k)*(cos(pi/(2*N+1)))ˆ2); om01=acos(-w01);
vals=abs(freqz(h,1,[om01 pi]));
if 2*floor(N/2)==N  extr=vals(2);  else   extr=-vals(1);  end
NN=1+2*(2*N+1);   N2=floor(NN/2);
hleft=h(1,1:N2);  hright=h(1,N2+2:NN);
h=[hleft/(2*extr) 0.5 hright/(2*extr)]; % Impulse Response
[H,om]=freqz(h,1,10000);
figure(1),
plot(om/pi,20*log10(abs(H))), grid on, axis([0 1 -100 5]),
xlabel(’\omega T/ \pi’), ylabel(’20log|H(eˆ{j \omega T})| [dB]’),

function [alfa,kc]=uarg(n,kc)
%
if n==0
alfa(2*n+1)=1/(1-kc*kc)ˆn;
elseif n==1
alfa(2*n+1)=1/(1-kc*kc)ˆn; alfa(2*n-1)=-(2*n*kc*kc+1)*alfa(2*n+1);
else
alfa=zeros(1,2*n+3);
%initialization
alfa(2*n+1)=1/(1-kc*kc)ˆn; alfa(2*n-1)=-(2*n*kc*kc+1)*alfa(2*n+1);
alfa(2*n-3)=-(4*n+1+kc*kc*(n-1)*(2*n-1))/2/n*alfa(2*n-1)-(2*n+1)*((n+1)*kc*kc+1)/2/n*alfa(2*n+1);
m=n;
% recursion
for j=m:-1:3
  c7=m*(m+2)-(j-3)*(j-1); c5=3*(m*(m+2)-(j-2)*j)+2*j-3 + 2*(j-2)*(2*j-3)*kc*kc;
  c3=3*(m*(m+2)-(j-1)*(j+1))+2*(2*j-1)+ 2*j*(2*j-1)*kc*kc; c1=m*(m+2)-(j-1)*(j+1);
  alfa(2*j-5)= - (c5*alfa(2*j-3) + c3*alfa(2*j-1)  + c1*alfa(2*j+1))/c7;
end
end

function h=h_uarg(m,k)
%
a=uarg(m,k);
ai=0;
for j=0:1:m,  ai(1+2*j)=a(1+2*j)/(1+2*j);  end
ai=[0 ai];
h=a2h(ai);
```

Towards Efficient Energy Management: Defining HEMS and Smart Grid Objectives

Ana Rosselló-Busquet and José Soler
Networks Technology & Service Platforms group, Department of Photonics Engineering,
Technical University of Denmark, 2800 Kgs. Lyngby, Denmark
{aros, joss}@fotonik.dtu.dk

Abstract—**Energy consumption has increased considerably in the last years. The way to reduce and make energy consumption more efficient has become of great interest for researchers. One of the research areas is the reduction of energy consumption in users' residences. In order to reduce energy consumption in home environments, researches have been designing Home Energy Management Systems (HEMS). Efficiently managing and distributing electricity in the grid will also help to reduce the increase of energy consumption in the future. The power grid is evolving into the Smart Grid, which is being developed to distribute and produce electricity more efficiently. This paper presents the high level goals and requirements of HEMS and the Smart Grid. Additionally, it provides an overview on how Information and Communication Technologies (ICT) is involved in the Smart Grid and how they help to achieve the emerging functionalities that the Smart Grid can provide.**

***Keywords*-Home Gateway, Home Energy Management System (HEMS), Smart Grid, power grid, Advanced Metering Infrastructure (AMI), Demand-Response (DR), Information and Communication Technologies (ICT)**

I. Introduction

Despite the fact that home appliances have become more energy efficient [1], electricity consumption in households has increased 30% over the last 30 years [2]. This is due to the fact that the number of appliances that can be found in households is also increasing. According to the International Energy Agency (IEA), European electricity consumption is going to increase 1.4% per year up to 2030, unless countermeasures are taken [3].

Residential buildings can reduce their energy consumption by becoming more energy efficient. This paper tries to identify the objectives that need to be fulfilled in order to deploy an energy efficient infrastructure. This infrastructure will help reduce the electricity consumption in users' residencies and make the electric grid more efficient by having more control over the electricity flow.

The research areas of efficient energy management can be divided into three more specific research areas: energy management in-home environments with Information and Communication Technologies (ICT), energy management in the power grid with ICT and ICT linking the Smart Grid and customers.

In this paper, 'utilities' are referred as the parties involved in the production and distribution of electricity, through the power grid. In addition, we use the term 'distribution' to refer to the process of electricity transport from the generation plants to the users' residences.

As shown in Table I, energy consumption in home environments can be reduced by installing Home Energy Management System (HEMS) [1] in users' residences. A HEMS will provide the users the necessary tools to manage and reduce their consumption. ICT will enable two way communication among the customers-location and utilities. ICT will benefit the users, as it will enable the provision of real time rates and billing status. If users take into consideration the price of electricity while consuming and reduce their consumption when the price is high, consumption will be optimized, as the utilities will be able to shift and shape demand. In addition, providing this exchange of information is one of the first steps towards optimization of energy distribution and production. This will provide the utilities with statistical data that will help predict energy consumption. In order to reduce losses and optimize energy distribution and production, the power grid elements need to be upgraded. Upgrading and introducing ICT in the power grid will lead to the so called Smart Grid. The Smart Grid will include new components and functionalities to efficiently manage the electricity distribution and production.

In this paper, the different goals that should be achieved in these areas in order to reduce energy consumption in home environment and make distribution and production of electricity more efficient are presented. When designing HEMS and Smart Grid systems, researchers usually focus on one of the existing goals such as the integration of the electrical vehicles or renewable energy sources. However, it is important that these systems are designed in the framework they are going to be deployed in and with all the goals they should achieve to maximize the benefits. This paper summarizes the different objectives of these research areas which can be used as a guideline.

The remainder of this paper is organized as follows: Section II introduces the concept of Home Energy Management System (HEMS) and describes the high level goals and requirements to deploy it successfully. Section III provides an overview of the actual power grid and introduces the Smart Grid concept. The Smart Grid objectives and functional areas are also introduced in this section. An overview of ICT in the Smart Grid is provided in Section IV, where the communication requirements for the functional areas of

Table I
IMPROVING ENERGY MANAGEMENT

Research areas / Issues	**Home environments**	**ICT**	**Power Grid**
Energy Goals	Reduce energy consumption	Provide grid's status information for efficient consumption and distribution	Reduce losses and integrate DER and EV
Who benefits?	Users	Users and utilities	Utilities
How?	HEMS	Information exchange	Smart Grid

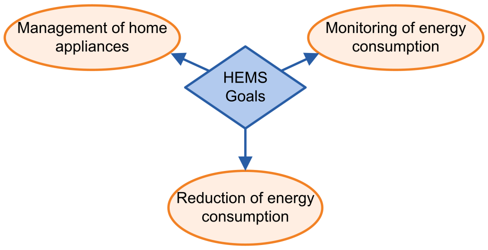

Management of home appliances
Monitoring of energy consumption
HEMS Goals
Reduction of energy consumption

Figure 1. Home Energy Management System Goals

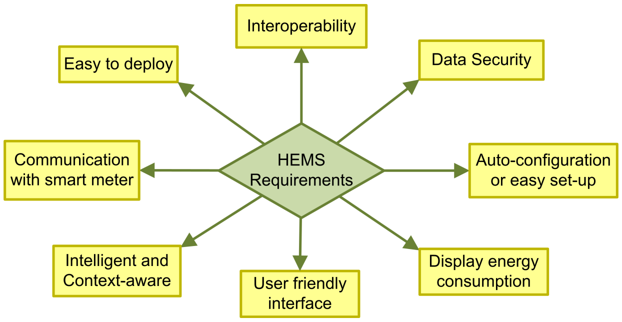

Interoperability
Easy to deploy
Data Security
Communication with smart meter
HEMS Requirements
Auto-configuration or easy set-up
Intelligent and Context-aware
User friendly interface
Display energy consumption

Figure 2. Home Energy Management System Requirements

the Smart Grid and issues are presented together with the ongoing ICT European projects. Finally, in Section V the conclusions are presented.

II. ENERGY MANAGEMENT IN HOME ENVIRONMENTS

Introducing Information and Communication Technologies (ICT) into home environments can help to reduce energy consumption of users. A HEMS is a system that includes all the necessary elements to achieve this reduction of electricity consumption in home environments. One of its main elements is the so called home gateway or residential gateway which is able to communicate and manage the rest of the home appliances and offers the users tools to reduce their consumption. Using context-aware information in HEMS will provide knowledge, which can be used to further decrease energy consumption.

Section II-A presents the goals HEMS should achieve and the high-level requirements it should fulfil. Section II-B will present the major challenges when designing HEMS.

A. HEMS High-Level Objectives and Requirements

The main objectives of a HEMS are shown in Figure 1. HEMS main goal is to reduce the energy consumption. However, to achieve this, monitoring energy consumption and managing appliances are needed. In order to reduce energy consumption, it is first necessary to know how energy is consumed. Therefore, consumption monitoring is needed. Secondly, it is necessary to manage the appliances in order to apply energy reduction strategies.

We consider that HEMS has to fulfil the requirements summarized in Figure 2 to achieve these goals satisfactorily:

- Easy to deploy: It has to be taken into consideration that HEMS should be easy to deploy into users' houses because deploying new cables or infrastructure is sometimes not a feasible nor cost efficient solution. Using already installed communication systems, such as wireless communication or power line communication which will minimize the costs and gain users' acceptance.
- Interoperability: in order to monitor and manage users' appliances efficiently, a home network has to be introduced, where devices can exchange information and commands, without interoperability conflicts. The appliances found in users' premises are usually manufactured by different producers and may use different communication technologies can lead to interoperability issues among devices.
- Data security: Security has to be incorporated into HEMS in terms of data encryption and authentication to protect the system against external threats. However, security issues will not be analyzed as they are out of scope of this paper.
- Auto-configuration or easy set-up: HEMS is going to be used by users without enough knowledge to perform network configuration tasks. Taking into consideration that users may add or change their home appliances, HEMS should provide easy to use configuration tools or in the best case the network should be auto-configurable.
- Display energy consumption: One of HEMS goals is to monitor energy consumption. This information should be available to users through the user interface. Furthermore, the displayed information could be shared with the utilities, to create statistical data of energy usage in home environments, or with third parties. Current consumption and also previous con-

sumptions, providing daily, monthly and even annual reports should be provided. Additionally, the possibility to compare the electricity consumption between months or even compare it to other sources, such as average neighborhood consumption or other users' consumption is an interesting feature.

- User friendly interface: The user interface should display information about the current consumption and also previous consumptions as stated above. Additionally, this interface should also provide management options, where the users may modify their preferences and control their appliances. User preferences are related to the strategy used to reduce users' energy consumption and may vary from system to system. It is also important to provide the possibility of controlling devices, as the system may apply undesired configurations and the user has to be able to correct them.
- Context-aware and intelligent: HEMS should have some intelligence to facilitate efficient energy management. This can be achieved by creating a context-aware system. A context-aware system is capable of collecting information from the environment, or context, and react accordingly. It is considered that a context-aware system can significantly improve the reduction of energy consumption. There are different ways in which context-aware systems can be implemented in HEMS: by defining energy polices or rules or by creating intelligent system.

  A HEMS that uses energy policies is a context-aware system which collects information from the environment and then uses this information together with the policies or rules to reduce energy consumption. This type of system is a static system and contains predefined policies or rules. However this policies and rules may be modified at any time by the user, by deleting, creating or modifying them.

  An intelligent HEMS requires a more complex system. In the context of this paper the intelligent HEMS is defined as a system that reduces energy consumption by using context-aware information to predict users' behavior and then applies the energy management strategy without compromising the users' comfort. Before being able to predict users' behavior and apply the energy management strategy, there has to be a learning process. This learning process includes (1) collecting context-aware information, which can include location-aware information, and (2) analyzing and processing this information to extract the users' routines and patterns. Once the learning process is completed, the system can extract the settings needed to reduce energy consumption. Unlike a rule based HEMS, this system is dynamic, adapts to user and also self-evolving.
- Communication with smart meter: As it will be explained in Section IV-B, the smart meter is found in costumers premises to measure the electricity consumption and communicate this information to the utility. Enabling this communication will provide the user with real-time price and billing status, energy consumption information, as well as possible services that may arise. HEMS should communicate with the smart meter to obtain this information. An example of a new service HEMS could obtain from the smart meter could be comparing the household energy consumption to other users' consumption. However, some of these new services could be offered through the Internet and not through the smart meter.

In the next section, the challenges found designing HEMS, when trying to comply with the above requirements, are presented.

B. Issues and Challenges

The main challenge to provide an efficient HEMS is interoperability. HEMS should provide seamless interaction between devices. However, there are a number of different home appliance manufacturers and communication technologies available for the user, which makes device interoperability problematic. In addition, devices of the same type, such as washing machines, can have different functionalities depending on the model. Technical incompatibility has limited market possibilities. Users are looking for a 'one size fits all' solution without having to worry about compatibility requirements. Therefore, one of the main challenge in HEMS is the variety of technologies, providing different communication methods, as well as the diversity of producers, providing different types of devices and services. This seamless interaction between devices could be provided by creating a central element, the home gateway, which would be able to communicate with all the devices. An example of how to solve this problem can be found in [4], [5].

Additionally, other challenging expectations from users have to be fulfilled, related to the following requirements: auto-configuration or easy set-up, user friendly interface and easy deployment:

- Easy to use and control: there is diversity in users' preferences and expectations when interacting with HEMS. Some users would like an interface that will give them advanced options while others would just like a simple system but without losing control of their devices [6]. Furthermore, users have different user interface preferences, some users would like to use their mobile phone or PDA, while others would rather use their computer or a controller. An example of how to deal with this can be found in [7].
- Easy to configure: complex configuration or professional help to configure the network is a drawback. HEMS should be easy to configure or even be auto-configurable. However this can be a challenge due

to the heterogeneity of home appliances and home technologies. Strategies and mechanism for software configuration and updates for devices installed in home environments can be found in [8], [9].

- Easy software upgrade: home appliances can have software installed, which in some occasions has to be updated. Software update should be easy for users to perform. The authors in [8], [9] present how to deal with software upgrades in home devices.

Moreover, designing HEMS as an intelligent and context-aware system is not an easy task and presents these following challenges:

- Design of context-aware systems, data collection and interpretation: HEMS may use sensors to collect information about the users' behavior. The system may have to work with different types of sensors and from different brands. This will force the system to be designed to deal with different sensor details which sets a barrier to interoperability. Dey et al. [10] proposes an infrastructure to support software design and execution of context-aware applications using sensors to collect data.

  Another issue is coping with the amount of data transmitted from home appliances and likely sensors. An example of how to deal with this can be found in [11].
- Policies and rules: There are two main challenges when using policies to implement energy management: coordination and contradiction. As the number of appliances in the house increases so does the number of policies, which can lead to coordination problems and contradictory rules. Tools to identify interactions and detect conflict resolution between policies should be incorporated into HEMS to manage rules and policies more efficiently. An example of this can be found in [12], [13].
- Intelligent HEMS: This type of HEMS should include an algorithm which after processing the collected data will be able to learn and predict the users' behavior. Examples of such algorithms can be found in [14], [15].
- Multiple-inhabitants: Prediction of users' behavior when there is more than one user in the home environment adds complexity to the predicting algorithm as each user has his/her own routines, practices and policies/rules, which may be different for each user.
- Not compromising users' comfort: HEMS should not have undesirable outcomes, it should be an intelligent system that can adapt to different situations and user behavior.

III. Energy Management in the Power Grid

As stated before, energy consumption in home environments is increasing and consumption patterns have considerably changed in the last years. The National Academy of Engineering acknowledges the power grid as the supreme engineering achievement of the 20th century, due to its ingenious engineering, its support for other technologies and the impact in improving quality life style for society [16]. However, the power grid has not changed significantly during the last century (the average substation transformer age is over 40 years old [17]). The power grid was designed to provide one-way flow of electricity and centralized generation. Furthermore, the actual power grid has limited automation and situational awareness and there is a lack of customer-side data. Therefore, an upgrade is needed to achieve efficient energy distribution and production and to fulfil the new power grid functionalities, which will be explained in Section III-D. The integration of these functionalities into the power grid will lead to the so called Smart Grid. Smart Grids will incorporate ICT to fulfil the new requirements and functionalities.

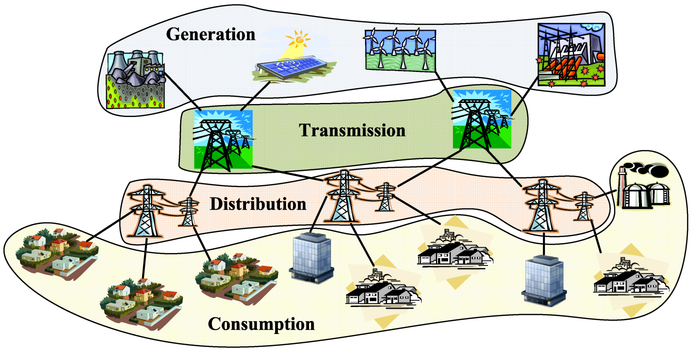


Figure 3. Power Grid Architecture Overview

The next section provides an overview of the power grid and Section III-B presents the Smart Grid concept and NIST and ETP Smart Grid view. In Section II-A, the high-level objectives of the Smart Grid are explained. Finally, in Subsection III-D the functional areas of the Smart Grid are presented.

A. Power Grid Architecture Overview

The power grid is an interconnected electricity network, which includes infrastructure of power generation, power transmission, power distribution and control. A typical power grid system is illustrated in Figure 3. A brief description of the different sections of the power grid is given below.

- Generation: it is composed of different types of power generators, such as coal, fossil fuels, natural gas, nuclear, wind turbines and hydraulic power plants among other. The main part of the electricity generated comes from large power plants located in strategic locations. For instance a coal power plant could be found near a coal mine. This section of the power grid is responsible to generate enough power to supply the demand.
- Transmission system: it carries electric power in an efficient way from the power plants to the distribution

system, where the power will be later consumed. Transmission system may span hundred of kilometers and line voltage are above 100 kV in Europe. The electricity is transmitted at very high and high voltages to reduce the energy losses in long distance transmission. The power plants are connected to the transmission system by a substation or transformer. A substation contains circuit breakers, switches and transformers, which step-up or step-down the voltages in the lines. Furthermore, substations can be found through the transmission system to change the voltages in the lines.

- Distribution system: it is similar to transmission system as their function is to carry electricity. However, the distribution system carries the electricity from the transmission system to the to consumers. This system includes medium and low voltages and also contains substations to interconnect it with the transmission system. The substations are also found thought the distribution system. The voltage in the transmission system lies between 69 kV and 240 V in Europe.
- Consumption system: composed by residential/home consumers, commercial and industrial consumers. The consumers are equipped with an electricity meter that register the consumers electricity consumption. Consumers may be found in rural, suburban or urban areas. Moreover, industrial consumers may be connected to the distribution grid at power levels higher than 240 V.
- Control Centers: In the traditional power grid the system operation and maintenance is done through centralized control and monitoring, Supervisory Control and Data Acquisition (SCADA). The control centers, communicate with the substations found around the power grid through private microwave radio, private fiber and public communication network [18].

As it will be explained in the following sections, the power grid architecture is changing into a more distributed and decentralized grid which will introduce new functionalities and ICT requirements.

B. Smart Grid Definitions

The Smart Grid concept is used by the different players involved in the power grid, not only utilities but also governmental entities. However, there is no standard definition. Different definitions exist, some functional, some technological, and some benefits-oriented:

- European Technology Platform (ETP): ETP defines the Smart Grid as "an electricity network that can intelligently integrate the actions of all users connected to it generators, consumers and those that do both - in order to efficiently deliver sustainable, economic and secure electricity supplies" [19].
- European Commission: European Commission defines the Smart Grid is "an electricity network that can efficiently integrate the behavior and actions of all users connected to it generators, consumers and those that do both in order to ensure economically efficient, sustainable power system with low losses and high levels of quality and security of supply and safety" [20].

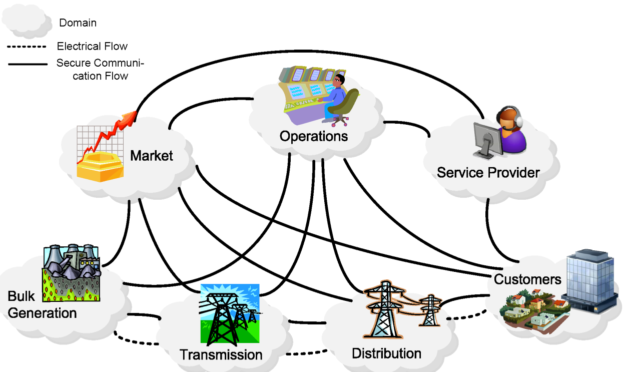


Figure 4. NIST Smart Grid Reference Model [24]

- Electric Power Research Institute (EPRI): "The term Smart Grid refers to a modernization of the electricity delivery system so it monitors, protects and automatically optimizes the operation of its interconnected elements from the central and distributed generator through the high-voltage network and distribution system, to industrial users and building automation systems, to energy storage installations and to end-use consumers and their thermostats, electric vehicles, appliances and other household devices" (EPRI) [21].
- US Department of Energy (DOE): "An automated, widely distributed energy delivery network, the Smart Grid will be characterized by a two-way flow of electricity and information and will be capable of monitoring everything from power plants to customer preferences to individual appliances. It incorporates into the grid the benefits of distributed computing and communications to deliver real-time information and enable the near-instantaneous balance of supply and demand at the device level" [22].

The Smart Grid is still an open concept. However, from these definitions it can be seen that the Smart Grid will enable a more dynamic, resilient, sustainable, efficient and adaptable grid with new capabilities and will involve the participation of different players, from customers to utilities. The Smart Grid will not only supply power but also information and intelligence. As the European Technology Platform (ETP) states "the smartness is manifested in making better use of technologies and solutions to better plan and run existing electricity grids, to intelligently control generation and to enable new energy services and energy efficiency improvements" [23].

Furthermore, NIST has developed a *Framework and Roadmap for Smart Grid Interoperability Standards* which "presents the first steps of a Smart Grid interoperability framework based upon initial standards and priorities to achieve interoperability of Smart Grid devices and systems" [24]. Additionally, it provides a conceptual architectural reference model shown in Figure 4. This model divides the power grid into domains, actors and applications. There are seven domains, which are further divided into subdomains. Each domain is a high-level grouping of actors, organizations, individuals, systems or devices which have similar objectives and may have overlapping functionality. NIST defines actors as devices, systems, or programs which are capable to make decisions, exchange information and interact with actors from the same or other domains. The tasks performed by one or more actors are defined as applications. In the following, there is a brief description of each domain and also some of the identified interfaces by NIST for which interoperability standards are needed.

- Bulk Generation: More renewable energy resources will be deployed into the Smart Grid. The main actors in this domain are big power plants such as renewable variable sources (solar and wind), renewable non-variable (hydro, biomass and geothermal) and non-renewable (nuclear, coal and gas). This domain may also include energy storage for later distribution of electricity.
- Transmission: Similar to the electricity transmission system today, this domain carries electricity over long distances. However, a two-way communication system will be deployed in substations and other intelligent devices found inside this domain, which will it make it substantially different from the current one.
- Distribution: The main changes to the distribution is the two-way communication system for monitoring and controlling. It may also include storage of energy and connection with alternative distributed energy resources (DER), such as wind farms and solar panels systems.
- Customers: In the Smart Grid, customers will be capable of generating, storing and managing the use of energy as well as the connectivity with their plug-in vehicles into the power grid. In this conceptual model the smart meters, besides being able to control and manage the flow of electricity to and from the customers, will provide information about energy usage and consumption patterns. Consumers will have a two-way communication with utilities and other third parties.
- Markets: Markets domain includes the operators and participants in electricity markets, such as market management, DER aggregation, retailing, wholesaling and trading among others. This domain is in charge of coordinating all the participants in the electricity market and ensuring a competitive market, in addition to exchanging information with third-party providers. For instance, roaming billing information for inter-utility plug-in-vehicles could be an example of a third-party service.
- Service Providers: This domain handles all third-party operations between domains. Examples of actors in this domain are installation and maintenance, billing, customer management and emerging services among other. Through this domain customers and utilities can exchange data regarding energy management.
- Operations: Operations domain is in charge of managing and operating the flow of energy through the power grid. It is connected to customers, substations and other intelligent devices through a two-way communication. Actors included in this domain are metering reading, maintenance and construction, security management and network operations among others. This domain provides monitoring, reporting, controlling and supervision status, which is an important to obtain a reliable and resilient power grid.

The European Technology Platform (ETP) network vision of the Smart Grid is shown in Figure 5 [3]. ETP does not define domains, however, it identifies the stakeholders involved, which may include governmental entities, consumers, traders, transmission and distribution companies, ICT providers and power equipment manufactures among others. Their vision on for the Smart Grid includes a two-way communication among these stakeholders which will provide coordination at regional, national and European level. ETP expects deployment of intelligent devices and distributed energy resources along the power grid. Furthermore, the European Commission identifies the following challenges for enabling Smart Grid deployment in Europe that have to be addressed [25]:

- Developing common European Smart Grids standards.
- Addressing data privacy and security issues.
- Regulatory incentives for Smart Grids deployment.
- Smart Grids in a competitive retail market in the interest of consumers.
- Continuous support for innovation and its rapid application.

C. *Smart Grid Objectives*

One of the main objectives of the Smart Grid is to make the power grid more efficient and to incorporate renewable energies. These objectives can help to reach the targets set by the European Commission.

Figure 6 summarizes the main high-level objectives the Smart Grid should fulfil. When designing the Smart Grid these goals should be taken into consideration and be integrated together in order to maximize the benefits.

- Enable the active participation of consumers: In the Smart Grid, customers will become active participants and will play a role in optimizing the operation of the system. The grid can ask the users to reduce their consumption to avoid shortages and reward them with

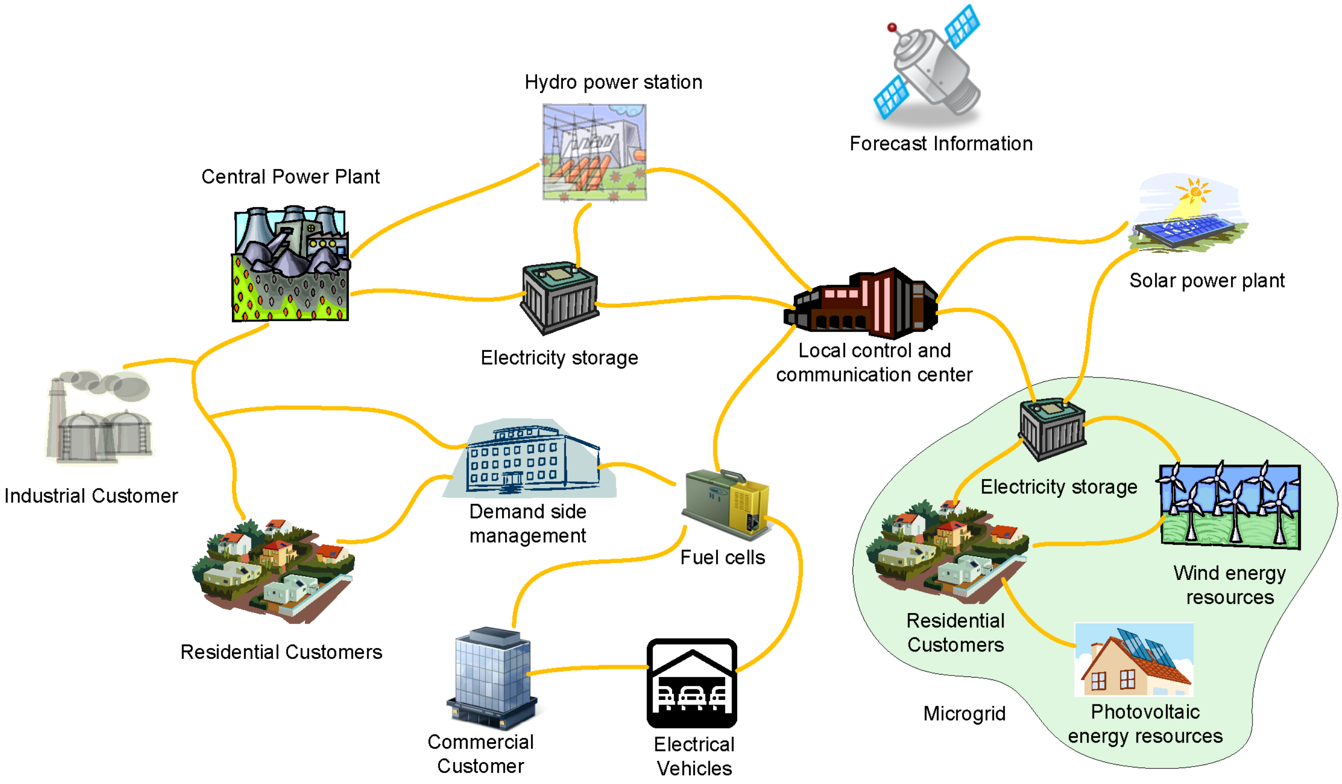


Figure 5. European Technology Platform Network Vision [3]

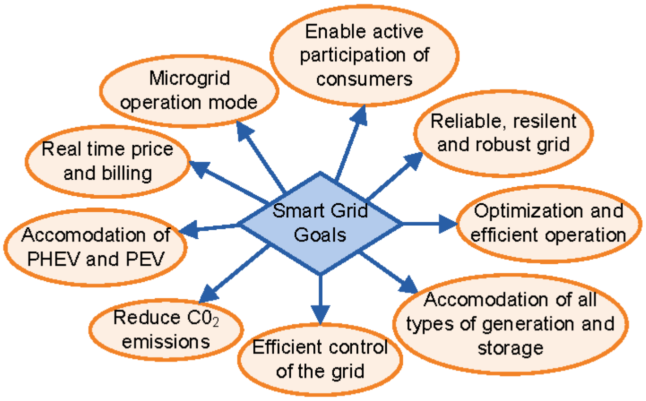

Enable active participation of consumers
Microgrid operation mode
Reliable, resilent and robust grid
Real time price and billing
Smart Grid Goals
Optimization and efficient operation
Accomodation of PHEV and PEV
Accomodation of all types of generation and storage
Reduce $C0_2$ emissions
Efficient control of the grid

Figure 6. Smart Grid Objectives

economical benefits. This process is referred as Demand Response (DR), discussed in more detail in III-D. DR will help utilities shape the demand according to the available production. Enabling this interactive service network in the power grid will improve the efficiency, safety ad reliability of the electricity transmission and distribution [22].

Customers are installing renewable energies in their premises and the Smart Grid should be capable of accepting these injections into the grid. Consumers in the Smart Grid that consume and generate energy will be called prosumers (**pro**-duces and con-**sumers**).

- Reliable, resilient and robust grid: the Smart Grid should improve security and quality of supply and reduce the number of blackouts and shortages to increase system reliability. In order to achieve a more resilient, reliable and robust grid than the actual power grid, Smart Grids should be easily reconfigurable and dynamic, in other words they should be self-healing system.
- Optimization and efficient operation: optimization and efficient operation of the grid implies a reduction of energy losses in power grid. Moreover, the Smart Grid should significantly reduce the environmental impact of the whole electricity supply system and improve the grid infrastructure operation. This can be achieved by upgrading the grid components and by using consumption statistics to foresee the electricity usage. It should also embody efficient and reliable alarm and fault management for self-healing procedures.
- Accommodation of all types of generation and storage: the Smart Grid has to accommodate from large centralized power plants to renewable energies installed in the users' premises or distribution systems. In addition, it is foreseen that new storage systems, such as community storages, may be included in the Smart Grid. To properly manage and control these new components, the

Smart Grid should be designed as a decentralized and distributed grid to better facilitate the connection and operation of generators of all sizes and technologies.

- Efficient control of the grid: introducing ICT into the power grid can help collect real-time data if the consumption, production and grid status. This information can be used to achieve efficient control of the power grid to balance loads and avoid blackouts and electricity shortages.
- Reduce carbon emissions: The European Commission Climate Action set three energy targets to be met by 2020, known as the "20-20-20" targets which are [26]:
  - A reduction in EU greenhouse gas emissions of at least 20% below 1990 levels
  - 20% of EU energy consumption to come from renewable resources
  - A 20% reduction in primary energy use compared with projected levels, to be achieved by improving energy efficiency.

  The carbon emissions can be reduced by incorporating some of the already mentioned objectives:

  - Reduce network losses by using more efficient components
  - Facilitate penetration of renewable energies, such as wind turbines and photovoltaic cells installed in the distribution grid or in users' premises.
  - Improve operational decisions in the power grid by using weather forecast to estimate the production of solar and wind farms. Also load forecast will make the power grid more efficient.
  - Use DR to reduce or even avoid demand peaks. [27] presents a system where the house consumption is kept under certain limit to avoid demand peaks.
- Accommodation of PHEV and PEV: Even though Plug-in Hybrid Electric Vehicles (PHEVs) and Plug-in Electric Vehicles (PEVs) are not yet wide-scale adopted, they should be taken into consideration when designing the Smart Grid. It is foreseen that the amount of PHEV and PEV will increase which will lead to a considerable increase of electricity demand which cannot be supported by the current power grid.
- Real time price and billing: ICT will provide the means to transmit real time price and billing information to the user. Furthermore, the consumers will be provided with greater information and options for choice of supply. The Smart Grid should incorporate the necessary elements to make this information available. The price of electricity is determined on the basis of supply and demand.
- Microgrid operation mode: Microgrids are generally defined as low voltage grids, which can range between few hundred kilowatts to a couple of megawatts, and include distributed generation sources, storage devices and consumer side. The ETP defines a microgrid "a controlled entity which can be operated as a single aggregated load or generator and, given attractive remuneration, as a small source of power or as ancillary services supporting the network" [3]. Although microgrids mainly operate connected to the rest of the power grid, they can automatically switch to islanded mode when faults occur, which could cause shortages or blackouts in the microgrid area. Microgrids should later on re-synchronize with the rest of the power grid with minimal service disruption, when the fault is stabilized. During islanded mode, the microgrid main functionality is to balance the distributed resources with local energy loads. Therefore, microgrids are capable of taking decision locally, while operating in islanded mode, but also when connected to the rest of the grid. However, coordination with the rest of the power grid and actors is necessary when the microgrid is connected to the power grid. Microgrids provide a new level of flexibility in configuring and operating the power grid, which may makes the grid more efficient, reliable, resilient and dynamic.

D. *Smart Grid Functional Areas*

The high level objectives described in the previous subsection can be classified into the six functional areas defined by DOE [28] in which most Smart Grid applications fall:

- Advanced Metering Infrastructure (AMI): refers to the infrastructure capable of measuring and collection consumption and generation at the consumer side and communicating it, in a two-way communication flow, to management system which makes this information available to the service provider. Additionally, AMI can provide real-time price information to the consumers. The consumers data can be collected and transmitted by using the smart meter installed in the consumer premises. The energy consumption data can be then be used by the service provider and utilities for grid management, outage notification, and billing purposes.
- Demand-Response (DR): is a reduction of consumption by the consumers, residential users, commercial or industrial as a response to high electricity prices or a request from the utilities in order to reduce heavy loads in the system. By using DR systems demand can be shaped to follow the production and therefore shortages and blackouts can be avoided. Furthermore, renewable energies, such as wind energy, have very variable power output depending on weather conditions. DR can help balance such loads providing a more flexible and dynamic power grid. However, accepting power reduction request is voluntary and can create some operational complexities.

- Wide-Area Situational Awareness (WASA): is a set of technologies that will enable improved reliability and prevention of power supply disruption by monitoring the grid status. WASA systems include sensors, which monitor the status of different elements in the power grid, intelligent devices, which can trigger an alarm in case of a critical situation, and two-way communication with the service providers. The main objective of WASA is to provide information about the grid status on real time. WASA will transform the power grid into a proactive systems, which will prevent critical situations instead of reacting to them.
- Distributed Energy Resources (DER): extends from distributed renewable energy sources to electric vehicle batteries, combined heat and power (CHP), uninterruptible power supplies (UPS), utility-scale energy storage (USES) and community energy storage (CES). It is expected that DER will be deployed along the power grid, specially on consumers and distribution system. Integrating DER into the power grid involves a major change as it implies decentralized generation and multi-directional flow of electricity, from utility-to-consumer, from prosumer-to-utility and even prosumer-to-consumer. DER applications require a more complex control situation and effective communication technologies to keep the balance in the power grid.
- Electric Transportation (ET): plug-in hybrid electric vehicle and plug-in electrical vehicles will drastically change the users consumption. This change of load has to be taken into consideration when designing the Smart Grid, which has to provide sufficient energy supply for electric vehicles and effectively manage the demand. This new kind of vehicles offer also the potential to function as storage devices, thus helping balancing the load in the Smart Grid by reducing the demand in energy shortage periods and absorbing the demand during excess supply periods.
- Distribution Grid Management (DGM): involves remotely control of the components found in the power grid. By using real time information about the power grid status and being able to remotely control the power grid, the Smart Grid becomes a more reliable power grid. Furthermore, Distribution and substation automation are part of the distribution Grid Management, which will provide more effective fault detection and power restoration. Supervisory Control and Data Acquisition (SCADA) and Distribution Management Systems (DMS), examples of DGM, require center-based control and monitoring systems in order to coordinate the power grid and keep balance.

Table II matches the presented objectives to be fulfilled by the Smart Grid in Section II-A and the above functional areas.

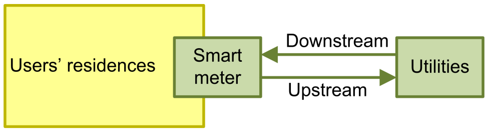


Figure 7. Communication between users' residences and utilities

E. Towards Smart Grid

The electrical grid has to undertake a transformation to reach the Smart Grids objectives. Introducing ICT into the grid will provide the communication tools to help reach some of the Smart Grids objectives. However, further changes in the Smart Grid components have to be done to successfully fulfil these goals. Advanced components, advanced control methods and improved decision support will be introduced in the power grid as it moves towards becoming a Smart Grid. In addition, sensing and measurements technologies should also be incorporated to evaluate the correct functionality of all elements in the grid and enable and efficient control.

IV. ICT in the Smart Grid

As it has been stated through this article ICT deployment in the power grid is an important step towards the Smart Grid. A reliable communication system that fulfills the Smart Grid's objectives, defined in III-C, and functionalities, defined in III-D, will determine the efficiency of the new power grid. The aim of ICT in the Smart Grid is provide more information about:

- Consumption: Knowledge about energy demand of the consumers will efficiency distribution. By providing this knowledge to the utilities, they can also foresee the energy needs of their consumers and avoid electricity shortages or blackouts. ICT can also be used to collect real-time consumption data to maintain the equilibrium between consumption and production.
- Production: Deployment of renewable energies, such as photovoltaic panels, is increasing in home environment and in the distribution system. Monitor and control is necessary for an efficient power grid.
- Status: Real-time monitoring the grid will help to detect critical situations. Remote control of the grid's component will help solve or even avoid this situations.

Deploying ICT in the power grid will start a cooperation between consumers and utilities. As shown in Figure 7, the upstream communication is defined as the transmission of data from the user to the provider and the downstream is defined as the one from provider to user. As stated before, the data transmitted in the upstream will include information about users' electricity patterns taking into consideration the likely installed renewable energies resources in their premises. The downstream communication is the transmission of electricity price and billing information from the utilities to the users. Having access to real time price

Table II
IMPROVING ENERGY MANAGEMENT

Functionalities Objectives	AMI	DR	WASA	DER	ET	DGM
Active participation of costumers	✓	✓	-	✓	✓	-
Reliable and secure supply	-	-	✓	-	-	✓
Self-healing	-	-	✓	-	-	✓
Optimization and efficient operation	-	-	✓	✓	✓	✓
Accommodation of all types of generation and storage	-	-	-	✓	-	✓
Efficient Control	-	✓	✓	-	✓	✓
Reduce Carbon Emissions	-	✓	-	✓	✓	✓
Accommodation of PHEV and PEV	-	-	-	✓	✓	-
Real-Time price and billing	✓	-	-	-	-	-
Microgird operation	-	-	✓	✓	✓	✓

and billing information will make the users become more conscious about their electricity consumption and they may try to reduce the associated costs, by avoiding peak hours, leading to a more distributed and efficient consumption. In addition, the utilities can use this downstream to ask their users to reduce their demand when demand peaks occur. This communication system will enable utilities to be proactive, acting before the problem occurs instead of reacting to it. Furthermore, utilities can offer new service that can be access by the user through this downstream.

Next section presents the communication requirements for the functional areas presented previously. Section IV-B explains how the smart meter and the home gateway collaborate with the smart meter. Finally, Section IV-C describes some of the issues raised by ICT in the Smart Grid.

A. ICT Requirements

Many communication and networking technologies can be used to support Smart Grid applications, which can vary around the power grid. S

Due to the different functionalities to be carried out by the Smart Grid, the ICT supporting them have different communication requirements. This will likely lead to a variety of communication technologies to be deployed in the Smart Grid. One of the technologies supporting ICT in the Smart Grid could be cable lines, fiber optic cable, cellular, satellite, microwave, WiMAX, power line communication, as well as short-range in-home technologies such as WiFi and ZigBee. Furthermore, different players in the Smart Grid have different views on what the communication requirements are for each of the functionalities presented in Section III-D as they are still under development. US Department of Energy (DOE) has written a technical report, *Communication Requirements of the Smart Grid* [28], which encloses the communication requirements based on the projections of future communications needs and the input of the different actors involved in ICT for the Smart Grid. Table III summarizes the Communication requirements found in this report and commented in the following subsections.

1) Advanced Metering Infrastructure (AMI):

- Bandwidth: It has been estimated that the bandwidth required for AMI will be between 10 to 100 kbps per node. However, communication among the aggregation point and the utility will likely have bandwidth requirements in the 500 kbps range.
- Latency: The delay between the consumption measurement and the moment at which the information is reported to the utility is not critical for AMI. However, demand response applications may be affected as they depend on this information.
- Reliability: AMI's level of reliability falls into the 99 percent to 99.99 percent range. The information provided by AMI has to reach the utilities. However, if some consumption measurement is lost it can be updated by the next measurement packet.
- Security: This network will carry consumption private information, therefore the it has to have a high security level.
- Backup Power: Backup power is not necessary as there is no consumption during blackouts.

2) Demand-Response (DR):

- Bandwidth: There are different DR programs that can be implemented: incentive-based, rate-based DR, demand reduction bids. Therefore, communications requirements of DR program may vary depending on the sophistication of the system.
- Latency: The latency requirements start from as little as 500ms, to 2 seconds, up to several minutes. This wide range is likely due to the different DR programs. Some programs maybe considered time-critical as if the demand is not reduced, it would lead to a system overload situation. However, if not used for grid balancing, relatively lower latencies may be necessary.
- Reliability: DR is likely to be used as a grid management tool, reliability will be important, and experts have estimated reliability will range between 99% percent to 99.99% level.

Table III
COMMUNICATION REQUIREMENTS FOR SMART GRID FUNCTIONAL AREAS [28]

Communication Requirements Functional Areas	**Bandwidth**	**Latency**	**Reliability**	**Security**	**Backup Power**
Advanced Metering Infrastructure	10-100 kbps per node 500 kbps for backhaul	2-15 sec	99-99.99%	High	Not necessary
Demand-Response	14-100 kbps per node	500 ms-several minutes	99-99.99%	High	Not necessary
Wide-Area Situational Awareness	600-1500 kbps	20 ms-200 ms	99.999-99.9999%	High	24 hour supply
Distributed Energy Resources	9.6-56 kbps per node	20 ms-15 sec	99-99.99%	High	1 hour
Electric Transportation	9.6-100 kbps,	2 sec-5 min	99-99.99%	Relatively high	Not necessary
Distribution Grid Management	9.6-100 kbps	100 ms-2 sec	99-99.999%	High	24-72 hours

- Security: As DR messages can be used for load management, its is important to verify the integrity of the information exchanged.
- Backup Power: There is no need for backup power, as the load management functions and DR programs are not necessary if there is a blackout.

3) Wide-Area Situational Awareness (WASA):

- Bandwidth: Data transferred in WASA is continuous and periodical rather than variable, but its throughput is expected to be high. Furthermore, the increase of distributed generation resources deployed and the introduction of new applications for phasor data, the bandwidth requirements will increased.
- Latency: WASA is used for real-time monitoring, therefore, the latency requirements are low between 20 to 200 ms. However, if historical data is transmitted the latency requirements can be higher.
- Reliability: Due to WASA is used to avoid critical situations its reliability is higher than AMI and DR.
- Security: Due to the same reason as above security in WASA systems is expected to be high.
- Backup Power: Backup power is necessary as the information provided by WASA is used to find the why and where the failure occurred and repair it.

4) Distributed Energy Resources (DER):

- Bandwidth: DER are unpredictable energy sources as they depend on weather conditions. Therefore there generation measurements need to be transmitted. It is estimated that the bandwidth will range between 9,6 kbps to 56 kbps.
- Latency: DER will imply having multiple energy sources feeding the distribution grid at multiple locations, which will complicate service restoration. Although the estimated latency ranges from 20 ms to 15 s, during faults, a lower latency maybe needed.
- Reliability: It has been estimated that the reliability of DER should be similar to AMI and DR.
- Security: As this information is expected to be critical during failures, security should be high.
- Backup Power: Backup power is estimated to last 1 hour as DER information can be used to restore power.

5) Electric Transportation (ET):

- Bandwidth: Electrical vehicles (EV) will cause a considerable increase in demand, which needs to be coordinated to not overload the system. ICT are needed for this coordination and also for billing purposes as the ET will likely charge at a variety of locations, including customers premises, office parkings, and other public or private locations during long-distance travel. Bandwidth requirements have been estimated to be between 9,6 to 56 kbps. However if EV batteries are used in DR programs to reduce demand peaks or absorb excess of electricity the necessary bandwidth could be up to 100 kbps.
- Latency: Latency estimates depend on the communication's main purpose, if it is used only for billing or also used for DR programs.
- Reliability: Due to the fact that some of the functionalities are similar to AMI and DR the reliability is the same for ET communications.
- Security: Security is important as information about the vehicles location should be protected. In addition, the charge and discharge of the EV should be done by authorized parties.
- Backup Power: Charging will not occur during a blackout, and backup power will likely not be critical. In fact, EV batteries, if charged, may serve as backup power not only for likely ET communication but also other potentially critical applications on the Smart Grid.

6) Distribution Grid Management (DGM):

- Bandwidth: DGM will make possible to remotely monitor and control the grid through automated decision-making, providing more effective fault detection and power restoration. Therefore, bandwidth requirement varies depending on the task to be performed, it is expected that during faults and critical situations more bandwidth will be needed.
- Latency: DGM latency requirements vary, from less than 1 s for alarms and alerts to 100 ms for messaging between peer-to-peer nodes. However, the maximum

latency is expected not to exceed 2 s.

- Reliability: The reliability is expected to be similar to the rest of Smart Grid applications, between 99% to 99,99%.
- Security: Due to the management nature of this functionality high security is required.
- Backup Power: Backup power is necessary to effectively restore power in case of blackout.

B. Smart Meter and Home Gateway

The smart meter is found in costumers premises to measure the electricity consumption. The smart meter is equipped with communication capabilities to transfer the measured information to the utilities. Therefore, the smart meter is one of the main elements of AMI and the information collected can be used for WASA and DR. The smart meter is in charge of communicating this information to the utility as this element should ensure the validity of the data collected. Furthermore, the smart meter may also receive data from utilities, such as real time price. It can also be used to communicate with the components involved in DR, to transmit or receive information from/to the utilities.

The home gateway, besides being the main element of HEMS and being able to communicate with all the home appliances and the smart meter, can be involved in DR and AMI. The smart meter can communicate the real time price information to the home gateway which can display this information in a user graphical interface. Then, the HEMS or the user can react accordingly to the price. The request to reduce electricity consumption associated to DR can also be forwarded from the smart meter to the home gateway, or depending on the DR implementation, be send directly to the home gateway. The home gateway, can then, according to user preferences, accept or decline the utilities requests.

In order to reduce the the information flow through the smart meter, historical consumption and billing information can be directly transmitted to the home gateway.

C. Issues

As it has been presented in the previous sections the Smart Grid is a system of systems which involves different actors and has different functional areas that require ICT. Deploying ICT will require that these parties work together to obtain the maximum benefits. This may require data interfaces between the different parties that ensure interoperability. Integrating ICT into the grid may require (1) to deploy a new communication infrastructure in the grid, (2) to standardize the communication between parties, (3) to fulfill all the necessary requirements for the Smart Grid applications.

Some of the barriers the stakeholders have to overcome when developing ICT for the Smart Grid are:

- Diversity of available technologies: There are different available communication technologies that can implement the ICT in the Smart Grid. This leads to a diversity of possible architectures, which can cause division in the power grid.
- Different entities: There are different players involved in the development of the Smart Grid. It is important that these entities can communicate with each other and exchange information in order to achieve the objectives and functionalities of the Smart Grid. However, due to the fact that there is diversity of available technologies and different protocols for machine-to-machine communication, interoperability between the different entities involved in the Smart Grid is becoming a challenge.
- Different functional requirements: The different functionalities have different requirements and requires communication between different players which is a challenge to incorporate the ICT into Smart Grid as more than one communication network maybe needed.
- Security: As Smart Grid will enable remote control of the power grid elements, therefore security becomes an issue. The ICT incorporated into the Smart Grid should be resilient to cyberattacks. Furthermore, as the consumption data of consumers is transmitted through AMI, data authenticity should be ensured.
- Privacy: Detailed private information about the consumption in consumers premises will become available in the Smart Grid. This information may be of interest to different entities, which the user might not be interesting to share it with, as they can extract patterns of home activity from metering data. For example, which devices they own, when do they use them, and lifestyle routines can be deduced from the users' load profiles. Therefore, the data regarding consumption of users should be protected when transmitted through the Smart Grid and maybe restricted to only some entities.
- Data Ownership: Data ownership is closely related to privacy issues and is still a topic under discussion. One may think that the data should be owned by the consumers and they should agree to share it with third-parties or not. On the other hand, this data is crucial for utilities as it can be used to forecast consumption and improve the power grid efficiency. However, these two statements are not mutually excluding. A solution could be that load profiles of consumers are owned by the user but aggregated data of load profiles can be used by utilities for consumption forecasting.

D. Ongoing European Projects

The aim of this section is to provide an overview of some of the ongoing European Projects dealing with research and development on the Smart Grid and its ICT aspect. The projects described below belong to the 7th FWP (Seventh Framework Programme) and have been paired up with the functional areas described in Section III-D.

- BEAMS, Buildings Energy Advanced Management System [29], 2010-2013: The aim of this project is to develop an advanced and integrated management system to enable energy efficiency in buildings, indoors areas and public spaces. This project is takes into consideration the interaction of the overall system with the power grid and management of heterogenous loads, such as public lighting, heating, ventilation and air conditioning (HVAC), and sources, such as renewable energy sources (RES) and electric vehicles. The goal of the project is not to develop new technologies but to integrate and combine already existing technologies to reduce the CO2 emissions. This project objectives are similar to the herein presented HEMS, but aimed to buildings and public areas instead. Furthermore, it is related to the integration of ET and DER in the customers side.
- CASSANDRA, A multivariate platform for assessing the impact of strategic decisions in electrical power systems [30], 2011-2014: The goal of this project is to develop a platform for realistic modeling of the energy market stakeholder aimed to provide test and benchmark working scenarios. This project aims to provide an aggregation methodology and software platform, and find the key performance indicators. Due to the fact that it deals with energy markets, this project is a step forward towards DR.
- HIPERDNO, High Performance Computing Technologies for Smart Distribution Network Operation [31], 2010-2013: This projects mainly objective is to enable DGM functionality, and indirectly AMI. This project's goal is to develop and demonstrate high performance computing solutions for realistic distribution network data traffic and management scenarios. This solution is based in near to real time data traffic with built-in security and intelligent communications for smart distribution network operation and management.
- INTEGRIS, INTelligent Electrical Grid Sensor communications [32], 2010-2012: The aim of this project is to develop a novel and flexible ICT infrastructure for the new Smart applications such as as monitoring, operation, customer integration, demand side management, quality of service and voltage control, Distributed Energy Resources and power system operations management. Therefore this projects offers the ICT infrastructure necessary for WASA and DGM. The Smart Grid ICT infrastructure developed is based on a hybrid Power Line Communication (PLC)/ wireless integrated communications system, which will be able to fulfill the communications requirements foreseen for the Smart Grid.
- MIRABEL, Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution [33], 2010-2012: This project is closely related to DR systems and supporting AMI. This is due to the fact that some of their main objectives are to: develop a concept of micro-requests to handle the energy demand and supply on a household level, design a decentralized scalable distributed system to handle the data load from customers, to forecast demand and supply based on historical and additional data (e.i. weather forecasts) and standardize the exchange of information between customers and utilities among others.
- OPENNODE, Open Architecture for Secondary Nodes of the Electricity SmartGrid [34], 2010-2012: This project has three main goals: 1- develop an open secondary substation node (SSN), 2- develop a Middleware to deal with the SSN and the utilities systems interaction for grid and utility operation, and 3- develop, based on standardized communication protocols, a modular and flexible communication architecture for the operation of the distribution grid. Therefore, this project is linked to DGM functionality.
- SMARTV2G, Smart Vehicle to Grid Interface [35], 2011-2014: The goal of this project is to create a safe, secure, energy efficient, controlled and convenient transfer of electricity and data to connect the electric vehicle to the Smart Grid. In this project, a new generation of technologies will be developed. This project aims to allow seamless and user-friendly energy load of electric vehicles in urban environments in the Smart Grid context. It is obvious that this project functionality is ET.

V. Conclusion

There is considerable literature on energy management and Smart Grid. This paper has tried to outline the main goals that have to be fulfilled by the Home Energy Management System and the Smart Grids. Additionally, an overview of the actual power grid and definitions for the term Smart Grid have been provided. Smart Grid models from NIST and ETP have also been presented and the functional areas of the Smart Grid have been explained. Furthermore, the role of ICT in the Smart Grid has been introduced and its benefits have been discussed. This article also presented some of the issues raised when introducing ICT in the Smart Grid. In the last section of this article some of the ongoing European Seventh Framework Programme have been presented. This projects have been related with the Smart Grid functionalities described in Section III-D.

When developing systems to reduce or make energy consumption more efficient, such systems usually focus on one specific capability. It is important that the overall framework and objectives are taken into consideration during the design of such systems to maximize their benefits. This paper can be used as a guideline of the objectives that should be fulfilled by HEMS and Smart Grid specially regarding ICT.

Furthermore, an interesting aspect of enabling communication between users and utilities is to use this bidirectional communication for Demand-Response systems. DR systems can be as simple as changes in electricity price or more complex such as the presented in [36]: Incentive-Based DR Programs, where utilities send a reduction requests to customers, and Demand Reduction Bids, where customer sends a demand reduction bid to the utility. An example of a DR system based on electricity price changes can be found in [37].

Finally, the authors of this article have developed a home gateway in JAVA using OSGi and ontologies and knowledge database. A detailed description of this home gateway can be found in [4], [38]. This home gateway has been specifically designed for HEMS and to help fulfill the requirements and objectives presented in Section II-A. It offers interoperability at the service level by using ontologies and incorporates a rule engine. Moreover, this home gateway can be used to exchange information with the utilities. The customer can define rules to automatically react to electricity price changes and reduction requests from the utility and generate Demand Reduction Bids.

References

[1] A. Rosselló-Busquet, G. Kardaras, J. Soler, and L. Dittmann, "Towards Efficient Energy Management: Defining HEMS, AMI and Smart Grid Objectives," in *The Tenth International Conference on Networks (ICN 2011)*, The Netherlands Antilles, 2011.

[2] BeyWatch Consortium, "D2.1:Service Requirement Specification," 2009.

[3] European Commission, *European Technology Platform SmartGrids - Vision and Strategy for Europe's Electricity Networks of the Future*. Office for Official Publications of the European Communities, 2006.

[4] A. Rosselló-Busquet, J. Soler, and L. Dittmann, "A Novel Home Energy Management System Architecture," in *UkSim 13th International Conference on Computer Modelling and Simulation*, 2011.

[5] D. Bonino, E. Castellina, and F. Corno, "The DOG gateway: enabling ontology-based intelligent domotic environments," *Consumer Electronics, IEEE Transactions on*, vol. 54, no. 4, pp. 1656 –1664, 2008.

[6] L T McCalley and C J H Midden and K Haagdorens, "Computing systems for household energy conservation: Consumer response and social ecological considerations," in *in Proceedings of CHI 2005 Workshop on Social Implications of Ubiquitous Computing*, 2005.

[7] Kistler, R. and Knauth, S. and Kaslin, D. and Klapproth, A., "CARUSO - Towards a context-sensitive architecture for unified supervision and control," in *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, 25-28 2007, pp. 1445 –1448.

[8] S. Grilli, A. Villa, and C. Kavadias, "COMANCHE: An Architecture for Software Configuration Management in the Home Environment," in *NBiS '08: Proceedings of the 2nd international conference on Network-Based Information Systems*. Springer-Verlag, 2008, pp. 283–292.

[9] J. Zhang, A. Rossell-Busquet, J. Soler, M. S. Berger, and L. Dittmann, "Home Environment Service Knowledge Management System," in *The 11th International Conference on Telecommunications (ConTEL)*, 2011.

[10] A. K. Dey, D. Salber, and G. D. Abowd, "A Context-based Infrastructure for Smart Environments," Georgia Institute of Technology, Tech. Rep., 1999. [Online]. Available: http://hdl.handle.net/1853/3406

[11] N. Shah, C.-F. Tsai, and K.-M. Chao, "Monitoring Appliances Sensor Data in Home Environment: Issues and Challanges," in *Commerce and Enterprise Computing, 2009. CEC '09. IEEE Conference on*, 20-23 2009, pp. 439 –444.

[12] M. Shehata, A. Ebertein, and A. Fapojuwo, "IRIS-TS: Detecting Interactions between Requirements in DOORS." *INFOCOMP Journal of Computer Science*, vol. 5, no. 4, pp. 34–43, 2006.

[13] M. Shehata, A. Eberlein, and A. Fapojuwo, "Managing Policy Interactions in KNX-Based Smart Homes," in *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, vol. 2, 2007, pp. 367 –378.

[14] H. Si, S. Saruwatari, M. Minami, and H. Morikawa, "A Ubiquitous Power Management System to Balance Energy Saving and Response Time based on Device-level Usage Prediction," *IPSJ Journal*, vol. 18, pp. 147–163, 2010.

[15] S. K. Das and D. J. Cook, "Designing Smart Environments: A Paradigm Based on Learning and Prediction," in *International Conference on Pattern Recognition and Machine Intelligence (PReMI)*, 2005.

[16] Greatest Engineering Achivements of the 20th Century. [Online]. Available: http://www.greatachievements.org/ Accessed: 25/06/2011

[17] G. W. Arnold, "Laying the Foundation for the Electric Grids Next 100 Years." Presented at ETSI Smart Grid Workshop: Standards: An Architecture for the Smart Grid Sophia Antipolis, France, 2011.

[18] K. Barnes and B. Johnso, "Introduction To SCADA Protection And Vulnerabilities," Idaho National Engineering and Environmental Laboratory, Tech. Rep. INEEL/EXT-04-01710, 2004.

[19] European Technology Platform SmartGrids for the Electricity networks of the Future, "What is a Smart Grid? Definition." [Online]. Available: http://www.smartgrids.eu/?q=node/163 Accessed: 25/06/2011

[20] S. Jiménez, "Smart Grid Mandate, Standardization Mandate to European Standardisation Organisations (ESOs) to support European Smart Grid deployment," European Commission Directorate-General for Energy, Tech. Rep. Ref. Ares(2011)233514, 2011.

[21] “Report to NIST on the Smart Grid Interoperability Standards Roadmap,” Electric Power Research Institut, Tech. Rep. Contract No. SB1341-09-CN-0031Deliverable 10, 2009.

[22] Department of Energy, “The Smart Grid An Introduction,” U.S. Department of Energy, 2008.

[23] “Activity Report 2009,” Union of the Electricity Industry - EURELECTRIC, Tech. Rep., 2009.

[24] National Institute of Standards and Technology, “NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0,” National Institute of Standards and Technology, Tech. Rep. NIST Special Publication 1108, 2010.

[25] “Smart Grids: from innovation to deployment,” Communication from the Commission to the European Parliament, the council, the European Economic and Social Committee of the Regions, Tech. Rep., 2011.

[26] “The EU climate and energy package,” European Commission Climate Action, accessed: 25/06/2011. [Online]. Available: http://ec.europa.eu/clima/policies/package/index_en.htm

[27] A. Rosselló-Busquet, G. Kardaras, V. B. Iversen, J. Soler, and L. Dittmann, “Scheduling home appliances usage to reduce electricity demand peaks,” in *Risø International Energy Conference 2011: Energy Systems and Technologies for the Coming Century*, 2011.

[28] “Communications Requirements of the Smart Grid Technologies,” Department of Energy (DOE), Tech. Rep., 2010.

[29] “BEAMS : Buildings Energy Advanced Management System,” October 2010, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=100732

[30] “CASSANDRA : A multivariate platform for assessing the impact of strategic decisions in electrical power systems,” November 2011, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=100985

[31] “HIPERDNO: High Performance Computing Technologies for Smart Distribution Network Operation,” February 2010, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=93772

[32] “INTEGRIS: INTelligent Electrical Grid Sensor communications,” February 2010, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=93726

[33] “MIRABEL: Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution,” January 2010, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=93821

[34] “OPENNODE: Open Architecture for Secondary Nodes of the Electricity SmartGrid,” January 2010, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PROJ&RCN=93771

[35] “MARTV2G: Smart Vehicle to Grid Interface,” June 2011, accessed: 23/01/2012. [Online]. Available: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION=D&CAT=PRO

[36] S. Mohagheghi, J. Stoupis, Z. Wang, Z. Li, and H. Kazemzadeh, “Demand Response Architecture: Integration into the Distribution Management System,” in *First IEEE International Conference on Smart Grid Communications*, 2010.

[37] A. Iwayemi, P. Yi, X. Dong, and C. Zhou, “Knowing When to Act: An Optimal Stopping Method for Smart Grid Demand Response,” *Network, IEEE*, 2011.

[38] A. Rosselló-Busquet, L. Brewka, J. Soler, and L. Dittmann, “OWL Ontologies and SWRL Rules Applied to Energy Management,” in *Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on*, 2011.

An Effective Usage of Transmitted Directivity Information for Target Position Estimation Algorithm

Hiroyuki Hatano
Faculty of Engineering,
Shizuoka University
3-5-1 Johoku, Naka-ku,
Hamamatsu-shi, Shizuoka
432-8561, JAPAN
thhatan@ipc.shizuoka.ac.jp

Tomoharu Mizutani
Graduate School of Engineering,
Shizuoka University,
3-5-1 Johoku, Naka-ku,
Hamamatsu-shi, Shizuoka
432-8561, JAPAN
f0930142@ipc.shizuoka.ac.jp

Yoshihiko Kuwahara
Faculty of Engineering,
Shizuoka University
3-5-1, Johoku, Naka-ku,
Hamamatsu-shi, Shizuoka
432-8561, JAPAN
tykuwab@ipc.shizuoka.ac.jp

***Abstract*—We consider localization systems for targets because information of the targets position, such as human, cars, dangerous objects, is very meaningful. Especially, we focus on the estimation of targets that exist in near wide area. In order to cover the wide area, the system, which has multiple networked ranging sensors, is useful. Such networked systems are often called as radar network systems or sensor network. The straight arrangement of the sensors is very useful because of easy setting and installing. Previously, receivers arranged in a straight line would generate a large positioning error in the same direction of the line, that is horizontal direction. In this paper, for reduction of this error, we propose a novel estimation algorithm using the directivity information of the transmitter. The proposed system has a electrical directional antenna in a transmitter. So the transmitter can emit signals to intended directions. In this paper, the error characteristic, which must be solved, is introduced firstly. Then, the proposed algorithm is presented theoretically. Finally, through the results of the computer simulations, the examples of the error reduction are demonstrated under various situations such as different target positions or type of sensors. The obtained results indicate the estimation characteristics that our proposal achieves the some reduction of the horizontal error even the sensors are arranged in a straight line.**

Keywords-*position estimation; localization; directivity of array; sensor network; radar network.*

I. Introduction

Information of target positions, such as human location, suspicious person detection and dangerous objectives, is very meaningful and attractive. On the basis of these demands, interest in position estimation systems has been growing. For far targets, it is relatively easy because the sensors can be allowed to using a narrow sensing area. However, for near targets, the sensing area is needed as the wide cover area. It is difficult to realize because wide covered antenna has received unintended signals easily. Then we have focused on the estimation of the position of targets in the near wide area before now [1].

To extract the unintended signals and derived information, the multiple sensors, which generate redundancy, is effective solution (Figure 1). So one of potent position estimation systems are built with the multiple sensors that are connected with networks. These sensors achieve reliable detections and accurate position estimation. Even inexpensive devices such as ultrasonic radars, will be able to achieve good performance. Moreover, networked sensors can obviously cover a wide detection area. Several attractive applications of position estimation systems have been suggested, including indoor monitoring systems (Figure 2(a)) and near-range automotive radars (Figure 2(b)) [2], [3].

The multiple sensor networks can be realized by any devices such as laser radars, radio radars, ultrasonic radars. We assume that the sensors in the network can output only measured ranges (a measured range list) to the targets. The reason is that the only ranging function can be realized with low cost and simple components which are used to construct the sensors The estimator must calculate target positions with high accuracy from only measured range lists provided by multiple sensors. For accurate positioning, it is important to discuss data processing of position estimation, which deals with measured range data from all of the sensors. Because the assumed sensors can have only the range function, we call the multiple sensor networks as radar network in this paper.

The system has the multiple sensors. So it is important about how to arrange the sensors. For easy setting within a limited space with simple wiring, a straight-line arrangement is usually preferred. Now we focus on the error of position estimation. The estimation errors depend on the layout of the receivers. In particular, for the case in which the receivers are arranged in a straight line, large errors are generated in the same direction of the line. Because a straight-line arrangement is useful and preferred very much, thus, a novel technique to reduce the estimation errors is needed.

The goals of the presented paper are as follows:

- Introduction of the conventional algorithm (EPEM) theoretically and particularly,
- Clarification of the error performance depending on the

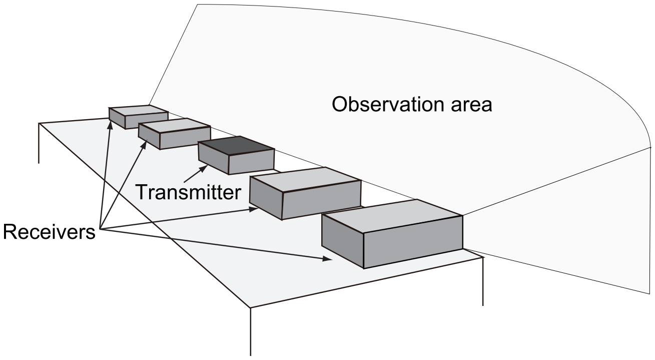
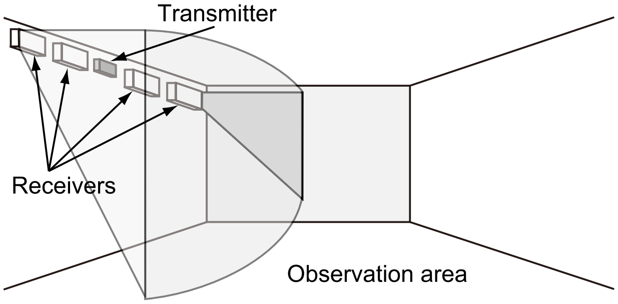
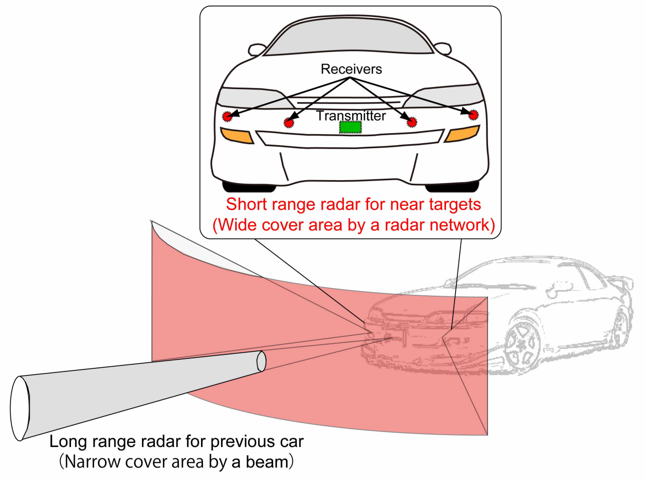


Figure 1. Concept of position estimation system by sensor networks



(a) Indoor monitoring system


Receivers
Transmitter
Short range radar for near targets
(Wide cover area by a radar network)
Long range radar for previous car
(Narrow cover area by a beam)

(b) Automotive radar system

Figure 2. Example of multiple sensing network systems

sensor arrangement and description of the problem,

- Proposal of the existence probability estimation method with directivity information (EPEMD) algorithm,
- Evaluation of the error reduction through various computer simulations.
- Presentation of the example performance through above evaluations under different conditions.

The proposed radar network in this paper has cooperative transmitter. The transmitter has the function of the variable directivity by electrical array antennas. On the processing of positions estimation, our proposal EPEMD calculates the target existence probability. Especially, the EPEMD algorithm uses the transmitted directivity information effectively. In the case of radar network systems, transmitters often use a directivity scan in order to reduce misdetections and expand detectable ranges for limited power [4]. Moreover, the construction of electrical directivity antennas is advantageous because the sensor device requires a long deliverable range with low power. Therefore, it is meaningful to propose an estimation algorithm that considers the directivity pattern.

The remainder of this paper is organized as follows. In Section II, related estimation algorithms, which use the multiple sensors, are introduced. And problems and advantages of our EPEM and EPEMD are described. In Section III, we present the system model and assumptions of the present study. In Section IV, we introduce the EPEM algorithm, which is a position estimation algorithm. The algorithm is explained theoretically, and the error performance and problems are presented. In Section V, the proposed EPEMD algorithm is presented in detail. In Section VI, the performance of the error reduction is demonstrated and evaluated under various situations. Finally, Section VII summarizes the present study and presents suggestions for further research.

II. RELATED RESEARCH

In this section, we introduce related researches, which estimate the target position by multiple sensors. Over the past few years, researchers around the world have developed several algorithms. The related estimation algorithms with multiple sensing devices are growing on the research area of sensor network systems or radar network systems. Here, some works about sensor networks are presented in the following literatures [5]–[9]. Also, radar network systems have been discussed in [10]–[12].

The typical estimation algorithms are summarized as below:

- Trilateration technique
- Stochastic approach

For estimating target positions, the trilateration techniques using geometric operations are most popular. The accuracy of these techniques is not optimum. Moreover they may also detect “ghost targets”, which are falsely detected about non-existent targets. This often occurs when the measured ranging errors are large [2], [11]. These lacks are generated because the trilateration techniques does not consider the influence of the measurement error.

In other techniques, measured ranges are treated as stochastic variables [13]–[15]. That is, by treatment of the stochastic variables, the influence of the measurement error can be considered. Typical techniques are, for example, estimation algorithms using minimum mean square error (MMSE) and maximum a posterior probability (MAP). The accuracy of these techniques is high compared to the above popular trilateration techniques. Among the stochastic methods, the accuracy of the MAP method is optimum. However, the calculation amount is high because data processing of the MAP method is very complex. Moreover the pre-knowledge, such as number of targets, is needed. Therefore, we proposed a novel estimation algorithm, namely, the existence probability estimation method (EPEM) [16]. The EPEM calculates the existence probability of targets and estimates the target positions. In the proposed method, the measured ranges are also treated as stochastic values. Moreover, the proposed method has approximately the same estimation accuracy and a lower calculation cost compared to optimum MAP methods.

However, problems are still remaining. One of the problems is the estimation error, which depends on the layout of the receivers. As mentioned in Sec. I, for the case in which the receivers are arranged in a straight line, large errors are generated in the same direction of the line. Usually, a straight-line arrangement is useful because such an arrangement is easy to build and can be set up within a limited space with easy wiring. Thus, a novel technique to reduce the estimation errors is needed. In this paper, we will try resolving this problem. That is, we will propose the reduction algorithm by using the directional information of the transmitter effectively.

III. Assumption and system model

In this paper, we will firstly point out the problem clearly. This section is described about the system model for explanation the conventional estimation and the problem. Figure 1 shows the system model. The assumed system has a transmitter and multiple receivers. First of all, for easy understanding, we explain using our radar network model with 4 radars and 2 targets. This is simple case. In Section IV-B, we introduce a more complicated case.

Figures 3 and 4 show the system model and the flow of data processing. Figure 4 also indicates the necessary parameters for the estimation. Figure 3 shows the sensor layout and the targets which are estimated. The numbers of receivers and targets are 4 and 2, respectively. The origin of the coordinate system is the center of the receivers. The target positions are given as $(x_1, y_1), (x_2, y_2)$. We note that each receiver is assumed to be located on the x-axis because the straight line layout is useful for setting and wiring to variable applications. The x positions of the receivers are $\alpha_1, \alpha_2, \alpha_3, \alpha_4$.

The kth receiver outputs a measured range list composed of the ranges to the targets, namely, $\tilde{R}_k = (\tilde{r}_{k1}, \tilde{r}_{k2})$. We assume the existence of a only direct path between the target and the transmitter/receiver. Subscript (˜) indicates measured values.

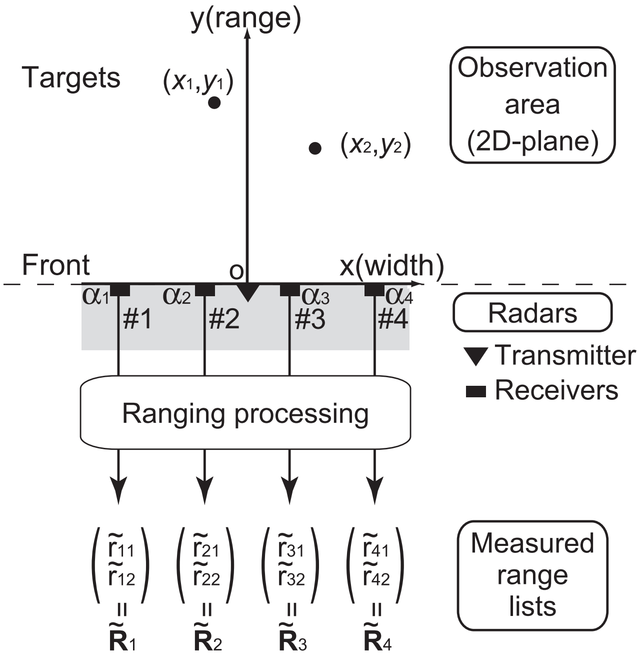

y(range)
Targets
(x_1,y_1)
(x_2,y_2)
Observation area (2D-plane)
Front
o
x(width)
α_1
α_2
α_3
α_4
#1
#2
#3
#4
Radars
▼ Transmitter
▬ Receivers
Ranging processing
$\tilde{r}_{11}$ $\tilde{r}_{12}$ = $\tilde{R}_1$
$\tilde{r}_{21}$ $\tilde{r}_{22}$ = $\tilde{R}_2$
$\tilde{r}_{31}$ $\tilde{r}_{32}$ = $\tilde{R}_3$
$\tilde{r}_{41}$ $\tilde{r}_{42}$ = $\tilde{R}_4$
Measured range lists

Figure 3. Layout of sensors and targets (4 receivers and 2 targets)

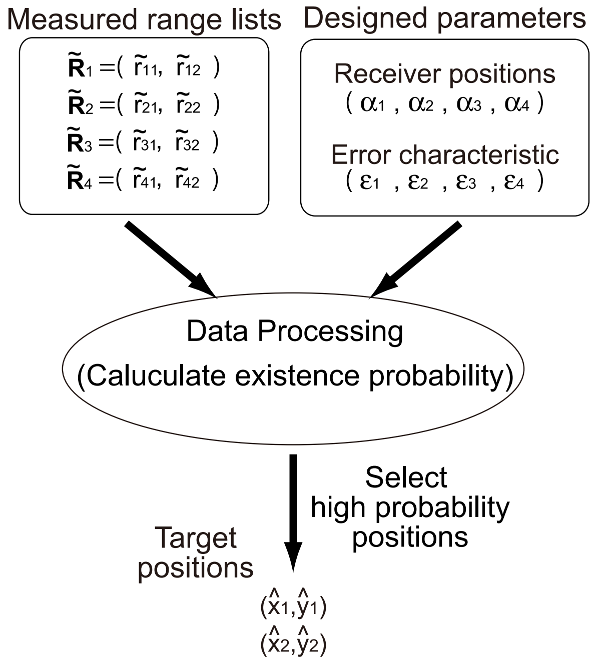

Measured range lists
$\tilde{R}_1 = (\tilde{r}_{11}, \tilde{r}_{12})$
$\tilde{R}_2 = (\tilde{r}_{21}, \tilde{r}_{22})$
$\tilde{R}_3 = (\tilde{r}_{31}, \tilde{r}_{32})$
$\tilde{R}_4 = (\tilde{r}_{41}, \tilde{r}_{42})$
Designed parameters
Receiver positions
$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$
Error characteristic
$(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$
Data Processing
(Caluculate existence probability)
Select high probability positions
Target positions
$(\hat{x}_1,\hat{y}_1)$
$(\hat{x}_2,\hat{y}_2)$

Figure 4. A data flow (4 receivers and 2 targets)

Each measured range $\tilde{r}_{kn}$ in the list includes a measurement error:

$$\tilde{r}_{kn} = r_{kn} + \epsilon_k, \tag{1}$$

where r_{kn} is the true range between the nth target and the transmitter / the kth receiver, and ϵ_k is a stochastic variable, the variance of which is denoted as σ_k^2. Using the measured range lists obtained from all receivers and the positions of the receivers, the target positions are estimated as shown in Figure 4.

IV. Estimation algorithm and its problem

The popular estimation algorithm of a target position is trilateration method which uses geometric operations [2], [11]. This is not optimum accuracy because the measurement errors are not considered. It may also detect "ghost target" that the detector outputs false position even there is no targets. This may happens in case of large measurement error or multipath environment. In order to address these problems, the proposed estimation method, which is described below, deals with the measured rages as stochastic variables.

In the first half of this section, we introduce the position estimation algorithm based on the existence probability which is named as the conventional in this paper. The estimation characteristics are then summarized, and the problem is pointed out. The estimation method, which is presented below, is called as EPEM(Existence probability estimation method).

A. Existence probability estimation method (EPEM)

For estimation of the target positions from the measured range lists provided by the receivers, we consider the following existence probability:

$$P(\hat{x},\hat{y}|\tilde{R}_1,\tilde{R}_2,\tilde{R}_3,\tilde{R}_4). \tag{2}$$

The Equation (2) includes the conditional probability. The above probability is the conditional probability of the target existence at $(\hat{x},\hat{y})$ when the measured range lists $\tilde{R}_1,\tilde{R}_2,\tilde{R}_3,\tilde{R}_4$ are obtained. Next, by using Bayes' theorem, Equation (2) can be written as follows:

$$\frac{P(\tilde{R}_1,\tilde{R}_2,\tilde{R}_3,\tilde{R}_4|\hat{x},\hat{y})}{P(\tilde{R}_1,\tilde{R}_2,\tilde{R}_3,\tilde{R}_4)}\cdot P(\hat{x},\hat{y}). \tag{3}$$

In Equation (3), the denominator does not depend on the estimated parameter $(\hat{x},\hat{y})$. Then, when $P(\hat{x},\hat{y})$ is distributed uniformly, Equation (3) may have the same distribution shape to the following:

$$P(\tilde{R}_1,\tilde{R}_2,\tilde{R}_3,\tilde{R}_4|\hat{x},\hat{y}). \tag{4}$$

Each receiver is assumed as independent. Because the measured range is an independent Gaussian variable, Equation (4) can be expressed as follows.

$$\prod_{k=1}^{4} P(\tilde{R}_k|\hat{x},\hat{y}). \tag{5}$$

There are the relationships between the targets ans ranges. With considering the combinations of targets and ranges, Equation (5) may be expressed as follows:

$$\prod_{k=1}^{4}\left[B_{k,1}P(\tilde{r}_{k1}|\hat{x},\hat{y})+B_{k,2}P(\tilde{r}_{k2}|\hat{x},\hat{y})\right] \tag{6}$$

where $B_{k,n}$ is the probability that the nth measured range in the measured range list of the kth radar, that is $\tilde{r}_{kn}$, means the range to the focused target. Next, the estimated parameters $(\hat{x},\hat{y})$ can be transformed with the distance from the transmitter/kth receiver to the target, that is, $\hat{r}_{kn}$ as follows:

$$\hat{r}_{kn}=\sqrt{(\hat{x}-\alpha_k)^2+\hat{y}^2}+\sqrt{\hat{x}^2+\hat{y}^2} \quad \text{(for all } n). \tag{7}$$

By using the above relational expression, Equation (6) can be converted to the following:

$$\prod_{k=1}^{4}\left[B_{k,1}P(\tilde{r}_{k1}|\hat{r}_k)+B_{k,2}P(\tilde{r}_{k2}|\hat{r}_k)\right]. \tag{8}$$

In this paper, we assume that there is not pre-knowledge at all. It is most difficult case. Hence, the value $B_{k,n}$ is equal value respectively. And the the value $B_{k,n}$ does not also depend on the estimated parameter $(\hat{x},\hat{y})$. So, the distribution of Equation (8) is the same shape to:

$$\prod_{k=1}^{4}\sum_{n=1}^{2}P(\tilde{r}_{kn}|\hat{r}_k). \tag{9}$$

The probability of $P(\tilde{r}|\hat{r})$ indicates the error characteristic of the receiver. The characteristic of the measurement error must be known as the specifications of the own receivers. Using Equations (7) and (9), the distribution of the existence probability of the target at position $(\hat{x},\hat{y})$ can be calculated from the measured range lists $\tilde{R}_1,\tilde{R}_2,\tilde{R}_3,\tilde{R}_4$, which can be obtained by the receivers. By selecting the local maximums of the distribution of Equation (9), the target positions can be estimated. For the multiple targets, the estimator may select multiple candidates, which have high probability. The distribution of Equation (9) is called as Existence probability of the targets. The example distribution of the existence probability is presented as Figure 5. Figure 5 shows the probability in case that the position of the target is $(x,y)=(0.5,9)$[m].

B. Case of N targets and K receivers

The simple case, which is the number of the receivers $K = 4$ and the number of the targets $N = 2$, was presented. Next we introduce more complicated case. That is the number of the receivers K and the number of the targets N. Figure 6 shows the system model. Figure 7 also shows the flow of the data processing.

Figure 6 shows the sensor layout and the targets. The numbers of receivers and targets are K and N, respectively.

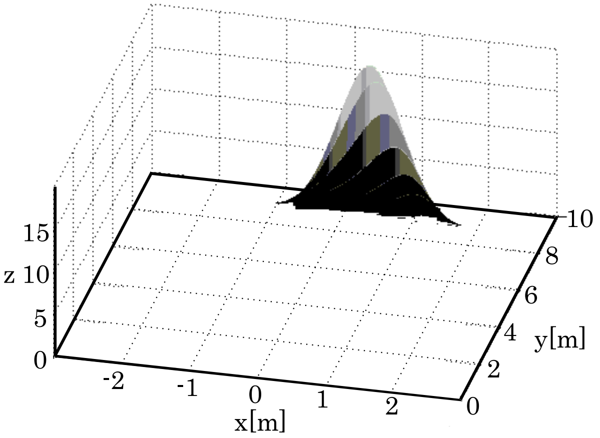


Figure 5. Example of the existence probability (1 target)

The target position is given as (x_n, y_n), $1 \leq n \leq N$. The x positions of the receivers are $\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_K$.

The kth receiver outputs a measured range list composed of the ranges, namely, $\tilde{R}_k = (\tilde{r}_{k1}, \tilde{r}_{k2}, ..., \tilde{r}_{kN_k})$. Here, $N_k(\leq N)$ is the number of ranges included in the measured range list $\tilde{R}_k$.

For estimation of the target positions, we consider the following existence probability, which includes the conditional probability:

$$P(\hat{x}, \hat{y}|\tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \cdots, \tilde{R}_K) \tag{10}$$

which is the same to Equation (2). The probability of Equation (10) is the conditional probability of the target existence at $(\hat{x}, \hat{y})$ when the measured range lists $\tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \cdots, \tilde{R}_K$ are obtained.

By the transformations which are the same to Equation (3)-(9), above Equation (10) has the same shape of the distribution to:

$$\prod_{k=1}^{K} \sum_{n=1}^{N} P(\tilde{r}_{kn}|\hat{r}_k). \tag{11}$$

The probability of $P(\tilde{r}|\hat{r})$ indicates the error characteristic of the known receiver. Using Equations (7) and (11), the distribution of the probability of the target existence at position $(\hat{x}, \hat{y})$ can be calculated when the measured ranges $\tilde{R}_1, \tilde{R}_2, \tilde{R}_3, \cdots, \tilde{R}_K$ are obtained.

The EPEM has approximately the same estimation accuracy as the MAP method, which is optimum in terms of maximum a posteriori probability [16].

C. Estimation characteristics and problems

In the following, we present the estimation characteristics of the EPEM algorithm described in the previous section. The simulation parameters are shown in Table I. In the simulations, we assume the measurement error as 0.3 m,

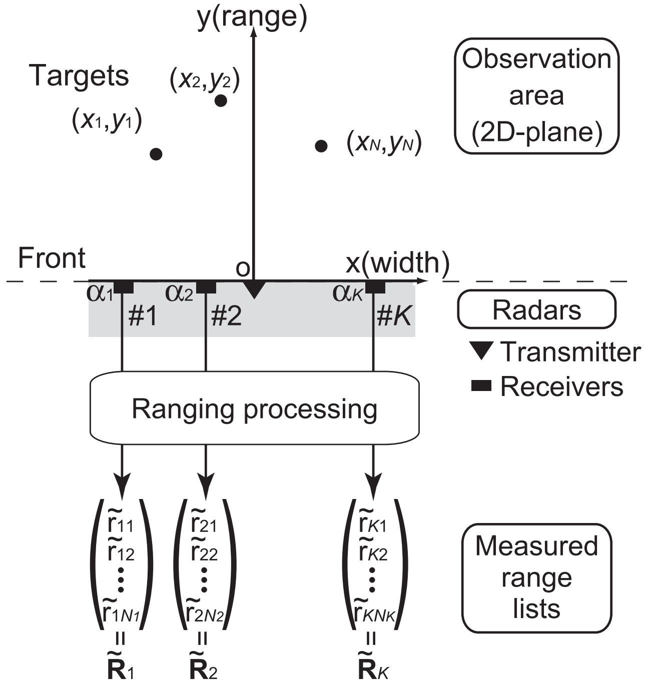


Figure 6. Layout of sensors and targets (K receivers and N targets)

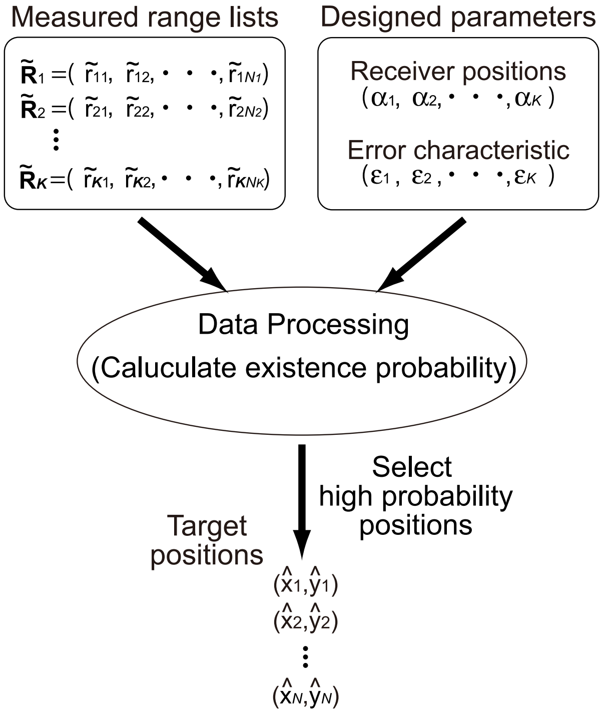


Figure 7. A data flow (K receivers and N targets)

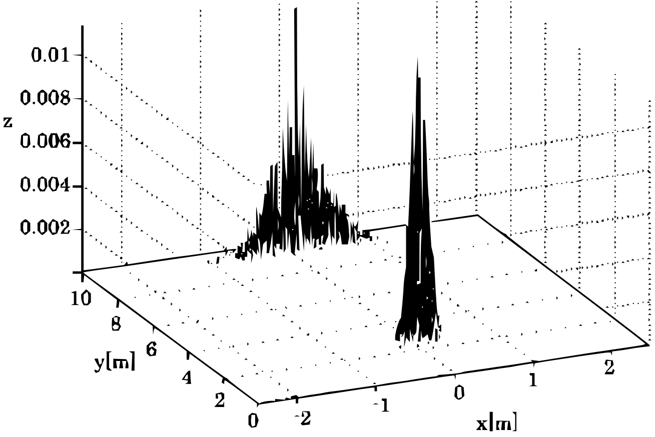


Figure 8. Distribution of estimated target positions (EPEM)

Table I
SIMULATION PARAMETERS

Number of receivers: K	3
Number of targets: N	2
Target positions	#1 $(x_1, y_1) = (0, 2)$[m] #2 $(x_2, y_2) = (0, 9)$[m]
Array width of receivers	2 m
Distribution of measurement error $\tilde{r}$	Gaussian distribution ($\sigma_k = 0.075$)
Number of iterations	50,000

Table II
CHARACTERISTICS OF THE ESTIMATED TARGETS (EPEM)

	Target 1	Target 2
Var$[\hat{x}]$	0.014	0.240
Var$[\hat{y}]$	0.002	0.002

which is typical [17]. According to this value, we set the standard derivation σ_k of the measured ranges ($4\sigma_k = 0.3$ [m]). The estimation trials of the targets are simulated. The trials generate the distribution of estimated positions. The results are shown in Figure 8. Moreover, the variance of the distribution for each of the targets position in Figure 8 are summarized in Table II. Figure 8 and Table II show that the error in the x-direction is larger than that in the y-direction. The reason for this is that the receivers are arranged along the x-axis. That is, large errors are generated in the same direction to the receivers' arrangement. In order to reduce the x-axis errors, we propose an estimation algorithm that uses the directivity of the transmitter.

V. PROPOSAL OF ESTIMATION ALGORITHM USING THE DIRECTIVITY OF THE TRANSMITTER

In this section, we introduce our proposal. The proposal solves the problem of the large error described in the previous section. The proposed algorithm is named as EPEMD (the existence probability estimation method with directivity information).

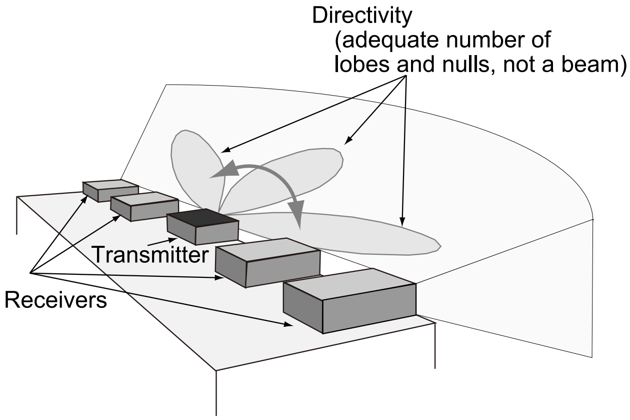


Figure 9. Image of the proposed system

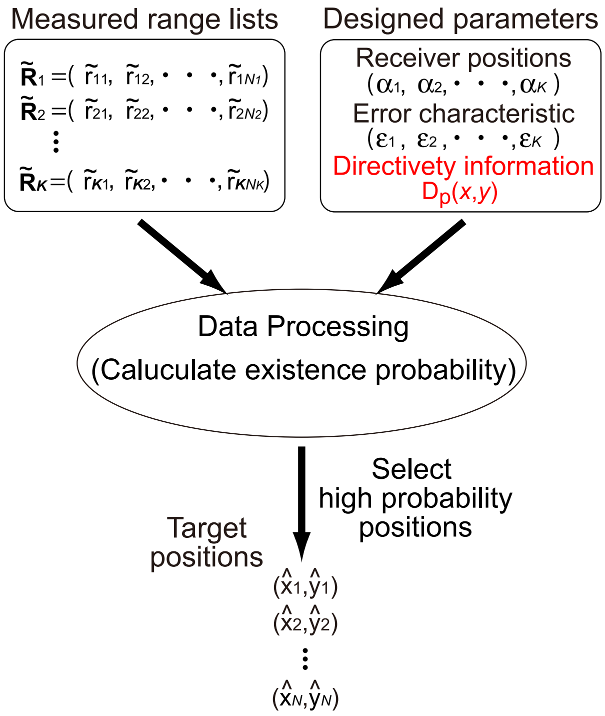

Measured range lists
$\tilde{R}_1 = (\tilde{r}_{11}, \tilde{r}_{12}, \cdots, \tilde{r}_{1N_1})$
$\tilde{R}_2 = (\tilde{r}_{21}, \tilde{r}_{22}, \cdots, \tilde{r}_{2N_2})$
$\tilde{R}_K = (\tilde{r}_{K1}, \tilde{r}_{K2}, \cdots, \tilde{r}_{KN_K})$
Designed parameters
Receiver positions
$(\alpha_1, \alpha_2, \cdots, \alpha_K)$
Error characteristic
$(\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_K)$
Directivety information
$D_p(x,y)$
Data Processing
(Caluculate existence probability)
Select
high probability
positions
Target
positions
$(\hat{x}_1, \hat{y}_1)$
$(\hat{x}_2, \hat{y}_2)$
$(\hat{x}_N, \hat{y}_N)$

Figure 10. A data flow (Proposal EPEMD)

We illustrate the image of our proposal in Figure 9. And Figure 10 also shows the flow of the data processing, which also indicates the necessary parameters for the estimation. The difference between Figures 1 and 9 is the transmitter of Figure 9 has a directivity. So, as seen in Figure 10, the directivity information can be used.

The system model is the same to Section III. That is, a signal is radiated from the transmitter, which is composed of two or more devices to achieve directivity. The reflected

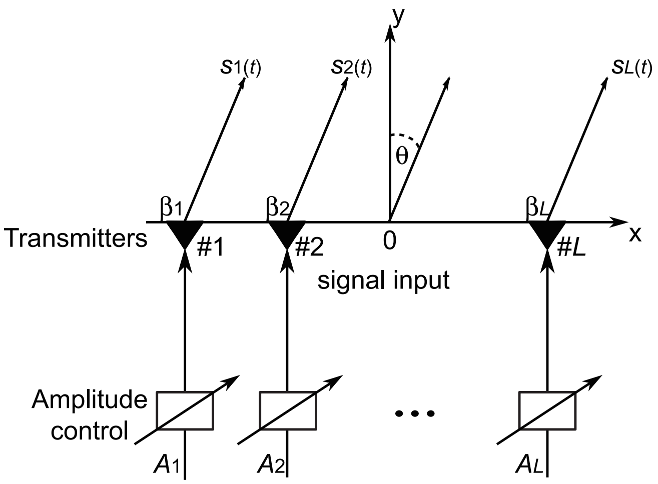


Figure 11. Structure of the transmitter array

signal from the target is received by the receivers, which are placed along a straight line (x-axis).

As shown in Figure 11, a transmission array is composed of L transmitters. The transmitters in the array are arranged symmetrically. The center of the array is the origin of the coordinate. The variables $(\beta_1, 0), (\beta_2, 0), \ldots, (\beta_L, 0)$ mean the positions of the transmitter. The variables A_l indicate the amplitude coefficient, and $s_l(t)$ also indicate the radiated signal from the lth transmitter. The total signal in the θ direction can be expressed as follows:

$$S_{\text{sum}}(\theta, t) = s(\theta, t) \sum_{l=1}^{L} A_l \exp\{j2\pi f_0(\frac{\beta_l}{c} \sin\theta)\} \tag{12}$$

where f_0 is the center frequency of the signal, and c is the speed of wave. The based signal and common characteristics of the transmitters, such as the directivity pattern of the element, is substituted as $s(\theta, t)$. In the present study, $|S_{\text{sum}}(\theta, t)|$, which indicates the gain generated by the array, is named as the directivity response pattern.

In the proposal, the directivity response pattern can be used effectively when the existence probability is calculated. We try to reduce the horizontal estimation errors using this directivity response pattern. The electrical directivity antenna, such as Figure 11, can change the directivity response pattern arbitrarily. However, for the purpose of clarity, we explain the EPEMD method using an example of a directivity pattern. The example is shown in Figure 12. Considering this directivity response pattern, the signal can be reflected only from targets that exist in the area within the beam, such as Target #1. In contrast, Target #2 cannot reflect the signal. Then, the function to specify the reflectable area is as follows:

$$D_p(x, y) = \begin{cases} 1 & \text{(area that can be reflected)} \\ 0 & \text{(area that cannot be reflected)} \end{cases}. \tag{13}$$

That is, Equation (13) means the area in which the target can reflect the signals or not. So, in this paper, the Equation (13) is called as reflectable area function. The above reflectable area function can be derived from the directivity response pattern.

From now, we explain the derivation of the reflectable area function. The directivity response pattern can be converted to the reflectable area of the $x - y$ plane by way of the following radar equation:

$$S = \frac{\gamma P_t}{R^4}. \tag{14}$$

The parameter S means the electric power of the reflected signal, that is, the signal received at the receiver. The parameter γ is determined on the basis of, for example, the antenna gain and the effective reflection area of the targets. In addition, P_t is the power of the transmitter, and R is the range from the transmitter/receivers to the target.

Then, if S is defined as the minimum detectable power at the receiver, the R means the maximum reflectable range obviously. Equation (14) can be rewritten as follows:

$$R = \sqrt[4]{\frac{\gamma}{S}} \sqrt[4]{P_t} \tag{15}$$

Next, we assume that the transmitting power becomes δP_t, that is δ times. Then, maximum reflectable range R' can be rewritten in terms of R as follows:

$$R' = \sqrt[4]{\frac{\gamma}{S}} \sqrt[4]{\delta P_t} = \sqrt[4]{\delta} R \tag{16}$$

As mentioned above, the absolute value $|S_{\text{sum}}(\theta, t)|$ in Equation (12) means the gain of the array. The gain of the array is related to δ. The maximum reflectable range R' can be calculated from Equations (16) when the gain of the transmitted power is $|S_{sum}(\theta, t)|$. As a result, the reflectable area function, that is Equation (13), can be calculated.

In the proposal EPEMD, the reflectable area function is considered into the existence probability of the targets. From Equations (13) and (9), we obtain the following equation:

$$\left[\prod_{k=1}^{K} \sum_{n=1}^{N} P(\tilde{r}_{kn} | \hat{r}_k)\right] \cdot D_p(x, y) \tag{17}$$

Equation (17) gives the existence probability considering the directivity of the transmission signal. The EPEMD estimates the target positions by searching the high values of the above existence probability. This search is the same to the conventional EPEM algorithm in the description of Section IV-A.

The directive antenna also generates null directions. In order to avoid null directions and cover a wide area, the

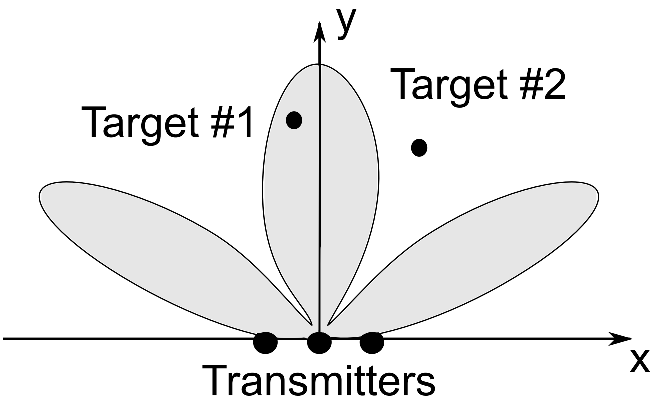


Figure 12. Directivity response pattern and targets

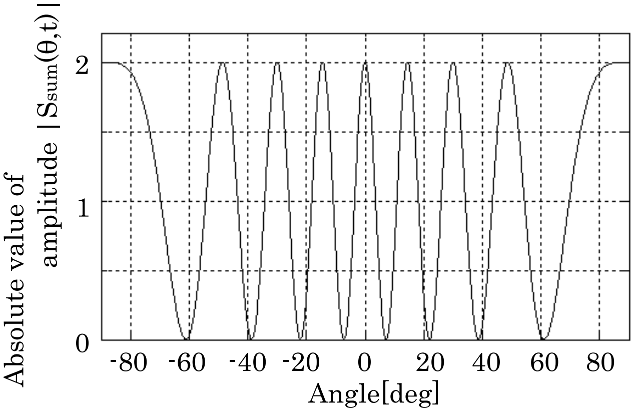


Figure 13. Designed directivity response pattern (Simulation I)

transmitter has to compensate the direction of the nulls. In the case of Figure 11, it is possible to change the directivity electrically, such as beam scans. So, in practice, the direction of the main lobe of the directivity has to be changed a small number of times to compensate the nulls and cover the wide area in a trial.

VI. Numerical examples and evaluation

We demonstrate and evaluate the estimation characteristics of the conventional algorithm and the proposed EPEMD algorithm from the viewpoint of error reduction. In this paper, we will present the characteristics in the case of two different types of the sensors. One is the radar sensors and the other is ultrasonic sensors. The results of these sensors are described as below. Especially, in the case of the radar, we simulate various situations which are different about the position of the target.

A. Simulation I

We designed the directivity pattern as shown in Figure 13. The simulation parameters are shown in Table III. Considering that targets exist in the near field, the width of the transmission array is set to 0.1 m. We then converted the directivity pattern into the reflectable area function using Equations (12) and (16). The reflectable area is shown in Figure 14. In the simulations, the maximum value of the $|S_{sum}(\theta, t)|$ is 2 and we assume that the maximum detectable range R' is 10 m. As mentioned in Section V, it is necessary to change the directivity in a detection trial such as beam scanning in order to detect targets over a wide area. However, for the evaluation of the position estimation characteristics of the algorithms, only one fixed directivity pattern is simulated. The parameters of the receivers are shown in Table IV.

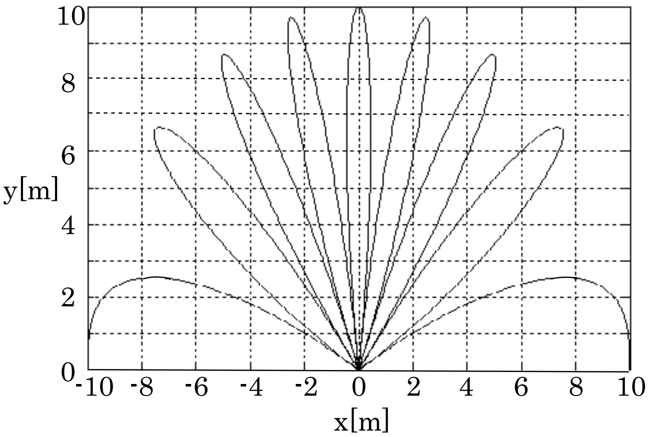


Figure 14. Reflectable area (Simulation I)

We simulate three cases. In Case 1, the target is located at (0,9) [m], which is a relatively long distance from the sensors. In Case 2, the target is located at (0,2) [m], which is short. In Case 3, the target is located at (0.5,5.2) [m], which is middle range. For the evaluation of the estimation errors, we use the variance of the distribution of the estimated positions, which are used in Section IV-C, as the performance measure. The variance is calculated from 50,000 estimation trials.

The results of the variance are shown in Table V. These variances are derived from the distribution of the estimated positions. The obtained distributions in Case 1 are shown in Figure 15. The results of the Case 2, 3 are also shown in Figures 16 and 17, respectively. From Table V, in the case of the EPEMD algorithm, the variance of both the x- and y-directions can be reduced compared to the conventional algorithm. Moreover, in the case of a long distance, the reduction in variance is large compared to the case of a short distance. In particular, the variance in the x-direction, which is the same direction of the arrangement of the receivers, can be decreased significantly.

B. Simulation II

We evaluate our proposal in terms of the use of ultrasonic radar networks. Ultrasonic radars are useful because the

Table III
PARAMETERS OF TRANSMITTER (SIMULATION I)

Frequency: f_0	24 [GHz]
Number of transmitters: L	3
Width of array [m]	0.1
Element positions [m]: B_l	-0.05, 0, 0.05
Amplitude control: A_l	0.5, 1, 0.5

Table IV
PARAMETERS OF RECEIVERS (SIMULATION I)

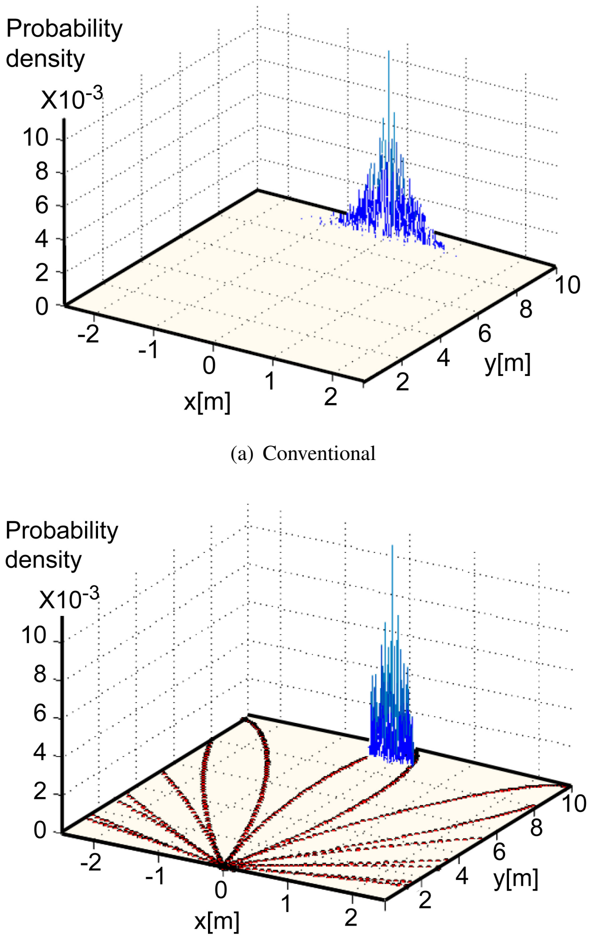
Number of radars K	3
Total width of receivers [m]	2m
Element positions [m]:	-1.0, 0, 1.0
Distribution of measurement error $\tilde{r}$	Gaussian distribution
Standard variation σ_k	0.075
Number of iteration	50,000

Table V
CHARACTERISTICS OF ESTIMATED TARGET (EPEMD, SIMULATION I)

	Target position [m]	Method	Var[$\hat{x}$]	Var[$\hat{y}$]
Case 1	(0,9)	Conventional	0.240	0.00237
		EPEMD	0.0339	0.00179
Case 2	(0,2)	Conventional	0.0139	0.00216
		EPEMD	0.0115	0.00227
Case 3	(0.5,5.2)	Conventional	0.271	0.0507
		EPEMD	0.0932	0.00313

Table VI
PARAMETERS OF TRANSMITTER (SIMULATION II)

Frequency:f_0	40[kHz]
Number of transmitters:L	3
Element positions[m]:B_l	-0.03 , 0 , 0.03
Amplitude control:A_l	1 , 1 , 1



(a) Conventional

(b) Proposed EPEMD

Figure 15. Distribution of estimated positions (Case 1)

Table VII
PARAMETERS OF RECEIVERS (SIMULATION II)

Number of receivers:K	4
x-position of receivers	-0.3, -0.1, 0.1, 0.3
Distribution of $\tilde{r}$	Gaussian Distribution
Standard variation σ	0.025m
Number of iteration	10,000
Observation area	x: -1m $\sim$ 1m y: 0m $\sim$ 3m
Target position	(0, 1.8) [m]

Table VIII
CHARACTERISTICS OF ESTIMATED TARGET (EPEMD, SIMULATION II)

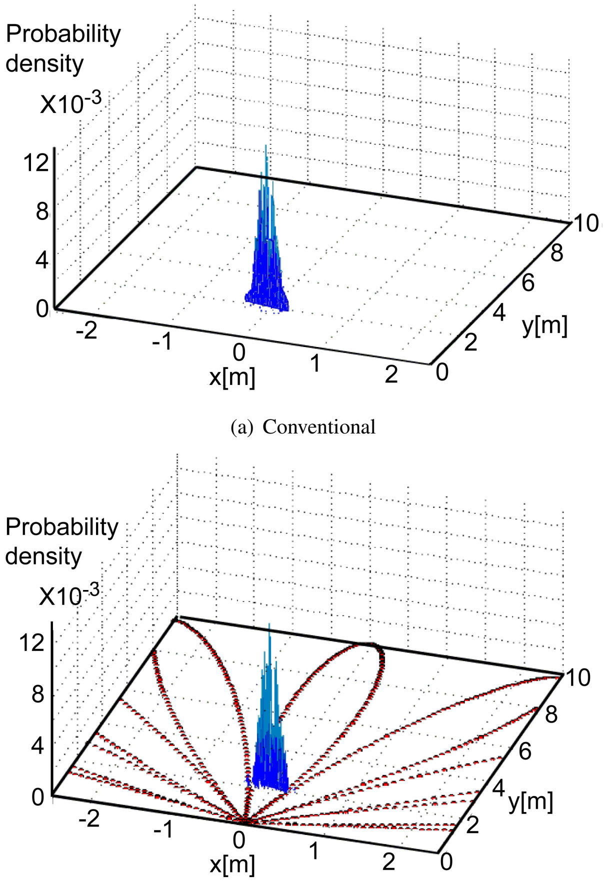
	Var[x]	Var[y]
Conventional	0.241	0.072
Proposal	0.188	0.029

devises are very low cost. In this simulations, we simulate the estimation by using the specification of the real devises.

We assume the real devices as MA40S4S and MA40S4R which are made by MURATA corporation [18]. The directivity of the devices is shown as Figure 18. Figure 18(a) is the directivity of the transmitter and Figure 18(b) is that of the receiver.

Using the above directivity, we design the directivity response pattern of the transmitter array. The specification of the array is summarized in Table VI. The directivity response pattern which is designed empirically is presented at Figure 19. And the reflectable area function, which is converted from the directivity response pattern using Equation (16), is also shown at Figure 20. In this conversion, we assume that the maximum value of the $|S_{\mathrm{sum}}(\theta, t)|$ is 3 and the maximum detectable range R' is 3 m.

The estimation performance of the target position is evaluated when the transmitter is the above array. The simulation parameter is summarized in Table VII. The performance measure is variances of the estimated positions. The variances are calculated in terms of x-direction and y-direction respectively. The statistics are derived from 10,000 estimation trials. The variances are shown in Table VIII. From the table, the variance of the proposal EPEMD is lower than that of the conventional method. That is, the proposal can reduce the estimation error. However the reduction



(a) Conventional

(b) Proposed EPEMD

Figure 16. Distribution of the estimated positions(Case2)

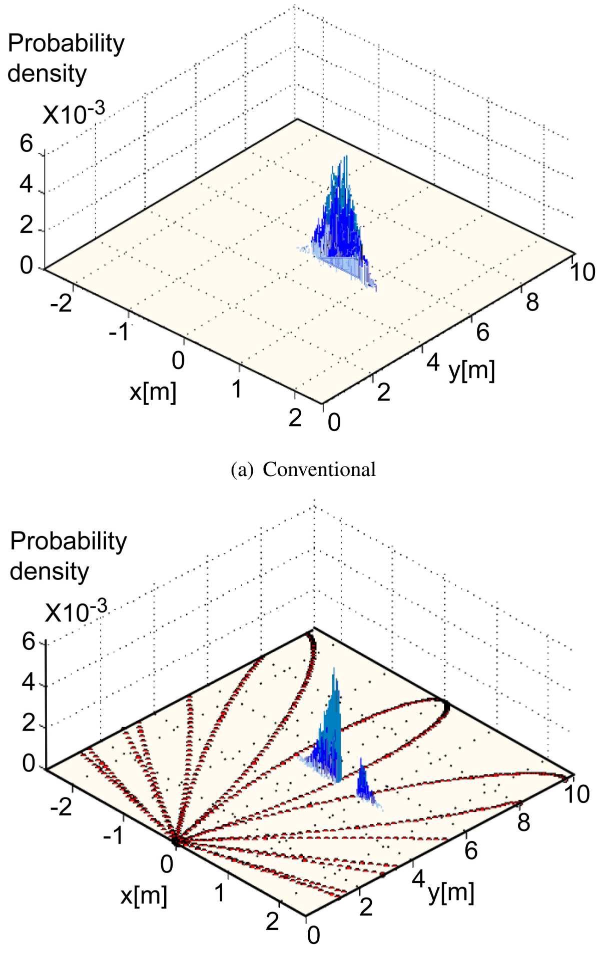
amount is not so large compared to the case of the results in Table V. This is because the target exists in short range.

Compared to the conventional method, our proposal needs the additional complexity such as the electrical directional antenna and the calculating processing. In case of the short range targets, the improvement is small. That is, the nearer the target exists, the smaller the reduction effect becomes. However the problems, which is large errors in case that the target exists in far range (See. Fig. 8), can be reduced by our proposal effectively.

VII. Conclusion and future work

In this paper, we considered the localization algorithm for the targets, which exist in near wide field. In order to cover the wide area, the networked multiple sensors can make sense very well. In our assumption, these sensors has only ranging function because of low cost and simple components. The position of the targets needs to be estimated by only range information. Such networked systems are often called as radar network systems.

The error of the estimation depended on the arrangement of the radars. The straight arrangement of the radars is very useful because of easy setting and installing. However, the radars arranged in a straight line would generate a large positioning error in the same direction of the line, that is the horizontal error. In this paper, for the reduction of this error, we proposed a novel estimation algorithm using the directivity information of the transmitter.



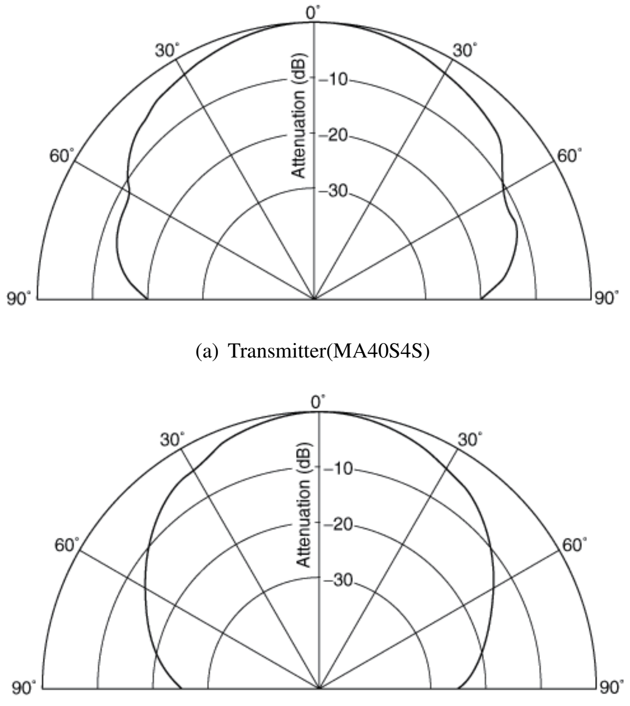
(a) Conventional

(b) Proposed EPEMD

Figure 17. Distribution of the estimated positions(Case3)

In the first half on the paper, to point out the problem clearly, we firstly describe the conventional system model and some estimation performance. After the recognition about the error problem, we introduced our proposed estimation method EPEMD theoretically.

Our system model used the transmitter which had the array component for changing directivity electrically. So the proposed EPEMD algorithm was effectively able to use not only the target existence probability which is calculated based on range but also the directivity information of the transmitter. Later in the paper, we tried various simulations which are different about the type of radars and the target positions for the demonstration and evaluation about our



(a) Transmitter(MA40S4S)

(b) Receiver(MA40S4R)

Figure 18. Directivity of real devices [18]

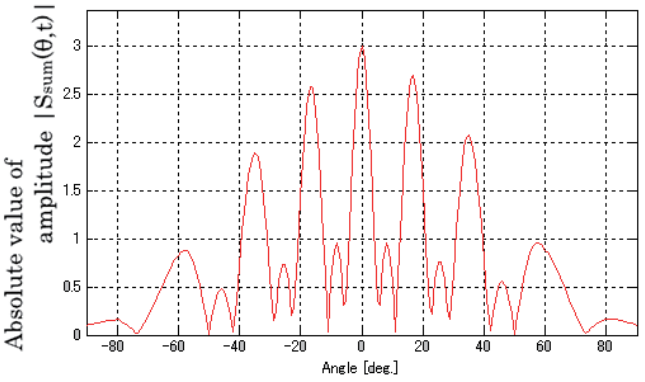


Figure 19. Designed directivity response pattern of the array (Simulation II)

estimation performance. By the computer simulations, we presented the reduction effect. That is, the proposal can reduce the horizontal errors compared to the conventional method. Moreover, the error in the direction of the receivers arrangement was effectively reduced as intended. However, the nearer the target exists, the smaller the reduction effect becomes.

As presented by the results of the computer simulations such as Table V, the position of y-direction can be estimated with very low variance, that is very high accuracy. This means that the radar network systems have significant

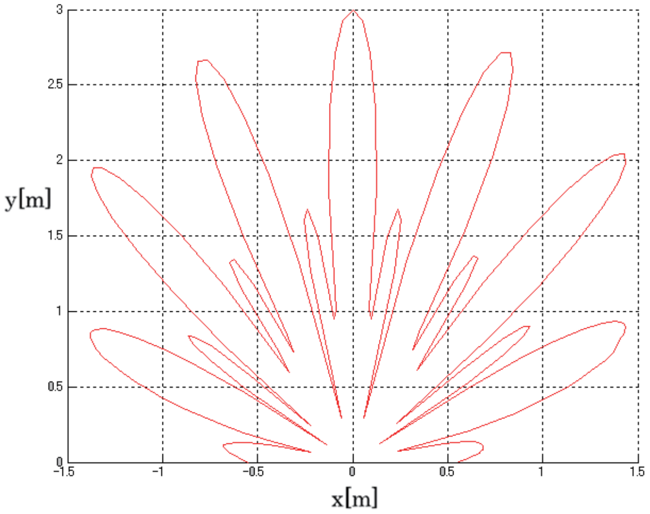


Figure 20. Reflectable area (Simulation II)

potential for various applications. However the error in the x-direction is still relatively large compared to the small error in the y-direction. This is still the problem.

As the future work, we will continue to solve this problem. Firstly, we will research suitable directivity pattern of the EPEMD algorithm. This is because the sharper beam will be able to reduce the horizontal error. However, the sharper beam maybe generates the large components and complicated signal processing of the transmitter. So we need to find the suitable directivity pattern. As other challenges for the error reduction, we will apply the reflected signals, which are often dealt with as multipath signals, to the EPEMD algorithm. The reason is that the multipath can surround the target even if the radars cannot surround the target, that is the radars are set as a straight line.

ACKNOWLEDGMENTS

The present study was supported in part by a Grant-in-Aid for Young Scientists (B) and by the Telecommunications Advancement Foundation.

REFERENCES

[1] H. Hatano, T. Mizutani, and Y. Kuwahara, “An error reduction algorithm for position estimation systems using transmitted directivity information,” *IARIA International Conference on Networks (ICN)*, pp. 267–272, Jan. 2011.

[2] M. Klotz and H. Rohling, “A high range resolution radar system network for parking aid applications,” *International Conference on Radar Systems*, May 1999.

[3] H. Rohling, A. Hoess, U. Luebbert, and M. Schiementz, “Multistatic radar principles for automotive radarnet applications,” *German Radar symposium 2002*, Sep. 2002.

[4] H. Hatano, T. Yamazato, and M. Katayama, “Automotive ultrasonic array emitter for short-range targets detection,” *IEEE international symposium on wireless communication systems*, pp. 355–359, Sep. 2007.

[5] V. Ramadurai and M. L. Sichitiu, “Localization in wireless sensor networks: A probabilistic approach,” *Proceedings of the International Conference on Wireless Networks, ICWN ’03*, pp. 275–281, June 2003.

[6] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, “Localization systems for wireless sensor networks,” *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 6 –12, dec. 2007.

[7] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, “The cricket compass for context-aware mobile applications,” *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 1–14, July 2001.

[8] S. Simic and S. S. Sastry, “Distributed localization in wireless ad hoc networks,” no. UCB/ERL M02/26, 2002.

[9] D. Niculescu and B. Nath, “Ad hoc positioning system (aps) using aoa,” *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, pp. 1734–1743, April 2003.

[10] Y. Chengyou, X. Shanjia, and W. Dongjin, “Location accuracy of multistatic radars (TRn) based on ranging information,” *CIE International Conference of Radar*, pp. 34–38, oct. 1996.

[11] R. Mende, “A multifunctional automotive short range radar system,” *German Radar Symposium 2000*, Oct. 2000.

[12] A. Hoess, H. Rohling, W. Hosp, R. Doerfler, and M. Brandt, “Multistatic 77GHz radar network for automotive applications,” *ITS World congress 2003*, Nov. 2003.

[13] H. Hatano, T. Yamazato, H. Okada, and M. Katayama, “Target position estimation using MMSE for UWB IPCP receiver,” *The 5th international conference on intelligent transportation systems telecommunication*, no. 45-3963283501, Jun. 2005.

[14] D. Oprisan and H. Rohling, “Tracking system for automotive radar networks,” *RADAR 2002*, pp. 339–343, Oct. 2002.

[15] M. Klotz and H. Rohling, “24 GHz radar sensors for automotive applications,” *International Conference on Microwaves, Radar and Wireless Communications*, vol. 1, pp. 359–362, Sep. 2000.

[16] H. Hatano, T. Yamazato, H. Okada, and M. Katayama, “A simple estimator of multiple target positions for automotive short range radar networks,” *IEEE vehicular technology conference 2007-spring*, pp. 2511–2515, Apr. 2007.

[17] M. E. Russell, C. A. Drubin, A. S. Marinilli, W. G. Woodington, and M. J. D. Checcolo, “Integrated automotive sensors,” *IEEE Trans. Microwave Theory and Techniques*, vol. 50, no. 3, pp. 674–677, Mar. 2002.

[18] Murata Manufacturing Co., Ltd., “Ultrasonic sensor application manual,” http://www.murata.com/products/catalog/pdf/s15e.pdf, retrieved: Jan. 2012.

Media Connectivity in SIP Infrastructures: Provider Awareness, Approaches, Consequences, and Applicability

Stefan Gasterstädt
adesso AG
Berlin, Germany
stefan.gasterstaedt@adesso.de

Markus Gusowski
Center for Information Services
and High Performance Computing
Technische Universität Dresden
Dresden, Germany
markus.gusowski@tu-dresden.de

***Abstract*—In SIP-based Voice over IP infrastructures, media data is usually exchanged directly between the endpoints using RTP without provider interaction. In contrast to the Public Switched Telephone Network where the delivery of all messages is the provider's responsibility, a SIP provider is not aware of media connectivity, i. e., whether a call was successful or not. This may lead to incorrect behavior when a Voice over IP provider offers services beyond signaling (for example, payment, prevention of Spam over Internet Telephony). Most existing mechanisms relating to media connectivity only aim at increasing the chance for connectivity or are endpoint centric and cannot achieve media connectivity awareness for the provider. We present and compare several approaches solving this problem that use both implicit and explict connectivity detection and notification mechanisms. Our favoured approach uses a set of behavioral rules for the user agents and implicit connectivity notification to achieve connectivity awareness. We also suggest SCTP for media transport and as an efficient connectivity detection mechanism. Besides conforming to the existing SIP standards and minimizing protocol changes, our solution is able to tolerate "lying" user agents. Measurements with our prototype SIP proxy implementation show that the impact on provider side call processing performance is negligible.**

***Keywords*-Voice over IP (VoIP), Session Initiation Protocol (SIP), Media Connectivity, Connectivity Awareness, SCTP.**

I. Introduction

The Session Initiation Protocol (SIP) [38] has become a majorly used protocol in Voice over IP (VoIP) communication. Often, SIP is used synonymously for VoIP infrastructures but it is actually one component of many. In particular, the request/response messages of SIP provide signaling (set up, modification and tear down of multimedia sessions), whereas the media data is nearly always transported directly between the user agents (UAs) using a separate media transport protocol. Only if two non-interoperable networks need to be connected, some specific Application Layer Gateway (ALG) will be involved. In terms of SIP, a gateway is just a special type of a user agent terminating the signaling path and, in this case, terminating the media path as well.

In most cases, the Real-time Transport Protocol (RTP) [40] is used for media transport between the endpoints. Usually, SIP and RTP use the User Datagram Protocol (UDP) [28] as the underlying transport protocol, while SIP sometimes utilizes the Transmission Control Protocol (TCP) [30] or the Transport Layer Security Protocol (TLS) [10] to secure the signaling. The RTP transport addresses and capabilities are specified and exchanged using the Session Description Protocol (SDP) [14] and its offer-/answer mechanism. SDP itself is carried in a SIP message body. All of these protocols belong to the application layer; their classification and the underlying, majorly used protocols are shown in the internet multimedia protocol stack, Figure 1.

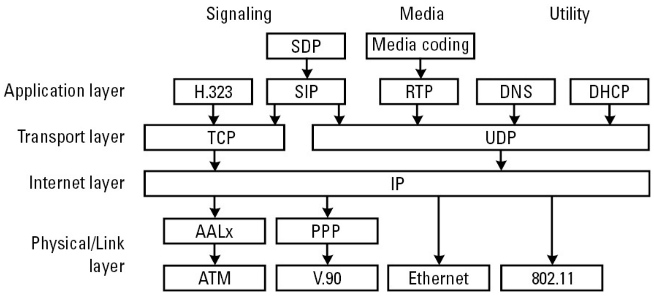


Figure 1: Internet Multimedia Protocol Stack [19, Fig. 1.1]

The SIP messages exchanged to set up and tear down a normal call (see Figure 3) and an example of a SIP invitation (see Figure 2) illustrate the separation and the interaction of these protocols. In this example figure, one can see the caller's invitation (*INVITE*) and the respective ringing/acceptance responses (*180 RINGING*, *200 OK*) send by the callee. Due to the fact that these messages contain all necessary information (e. g., current Internet Protocol (IP) adresses, port numbers, negotiated codecs and further media parameters), the subsequent acknowledgement (*ACK*), the media session itself, and the tear-down of the session (*BYE*, *OK*) can be send directly between both UAs.

SIP utilizes the Uniform Resource Identifier (URI) schema [5] to address users, single devices or end points and resolves these URIs to IP addresses [29] by using SIP proxy

```
INVITE sip:19@10.3.8.20:5060 SIP/2.0                                              % SIP request is an invitation
Via: SIP/2.0/UDP 10.3.8.18:5060;branch=z9hG4bK_000FC9022702_T664769F9             % Route of the message
Session-Expires: 1800                                                             %
From: "SIP Telefon 18" <sip:18@10.3.8.20:5060>;tag=000FC9022702_T634233581        % Caller information
To: <sip:19@10.3.8.20:5060>                                                       % Callee information
Call-ID: CALL_ID11_000FC9022702_T907830378@10.3.8.18                              %
CSeq: 589933214 INVITE                                                            %
Contact: <sip:18@10.3.8.18:5060>                                                  % information to potentially send
Max-Forwards: 70                                                                  %           SIP messages directly
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,REFER,DO,UPDATE,OPTIONS,SUBSCRIBE,PRACK,INFO  %
Supported: 100rel,timer,replaces                                                  %
User-Agent: ALL7950 02.09.31                                                      %
Content-Type: application/sdp                                                     % SIP body will contain a session
Content-Length: 231                                                               %                     description
                                                                                  %
v=0                                                                               % The body of this message
o=18 212024437 212024437 IN IP4 10.3.8.18                                         % contains the description of
s=ALL7950 02.09.31                                                                % the session offered by the
c=IN IP4 10.3.8.18                                                                % caller.
t=0 0                                                                             %
m=audio 41000 RTP/AVP 0 18 4                                                      % It contains information about
a=rtpmap:0 PCMU/8000/1                                                            % the media type, codec, ip
a=rtpmap:18 G729/8000/1                                                           % address and port number and
a=fmtp:18 annexb=no                                                               % further.
a=rtpmap:4 G723/8000/1                                                            %
a=sendrecv                                                                        %
```

Figure 2: Example of a SIP Invitation

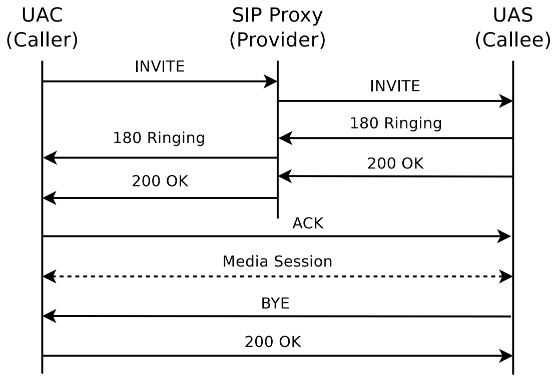


Figure 3: SIP Dialog of a Call

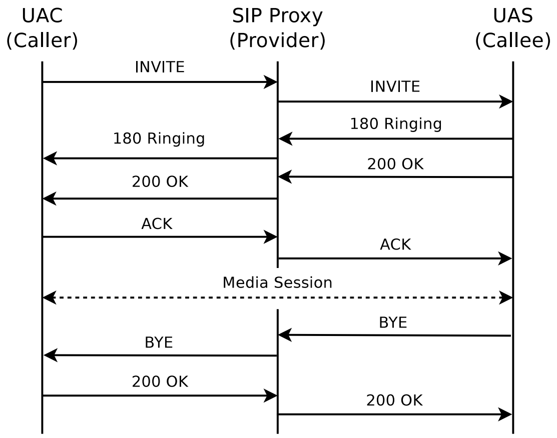


Figure 4: SIP Dialog of a Call with In-Route Proxy

servers and Domain Name Service (DNS) lookups [23], [24]. Users can call others without knowing their current IP address, because session invitations are routed to the SIP proxy that is responsible for the callee's URI domain; and as a next step, this proxy uses its location service to locate the callee and forwards the *INVITE* request to the addressed user. The location bindings can be updated by each respective user sending a *REGISTER* request to its SIP provider's registrar. Depending on its configuration, a SIP proxy may or may not request to stay in the route of any further SIP signaling (see Figures 3, 4). Independently, in most cases the media transmission is done directly between the UAs via RTP.

It is a known problem that the basic SIP infrastructure does not conform to the Network Address Translator (NAT) friendly application design guidelines described in RFC 3235 [41]. As a consequence, NATs and firewalls cause serious problems for SIP message delivery and media connectivity in conjunction with the separation of signaling and media delivery, dynamic port allocation, or RTP's "$x + 1$" port schema. In contrast to the UA-to-UA media connection, there are solutions for SIP messages; for example, by simply traversing NAT using symmetric response routing [37]. Examples of NAT and firewall traversal for SIP are given in [34].

The explicit separation between the session signaling and media delivery comes along with a significant implication: VoIP providers offering SIP services are unaware of whether or not the media stream is actually received by the endpoint(s), i. e., whether there is *connectivity* or not. SIP does not check for connectivity, and the condition is not signaled in any way. Therefore, a SIP provider cannot know if two users will actually be able to communicate, even if a SIP session was successfully established. There are several reasons why media streams negotiated between the UAs may be blocked in one or both directions, mainly because of NATs and/or firewalls [16], [44], but other network problems like the lack of a network route, node crash, configuration problems, or codec mismatch could be responsible as well [2]. This is in contrast to the traditional Public Switched Telephone Network (PSTN), where there is always connectivity once signaling completes successfully. Admittedly, there are some rare cases where people cannot talk to each other allthough there has been a successful ringing and call acceptance before. However, the PSTN phone provider will be aware of this failure.

There are, however, important scenarios where it is desirable for the provider to know the media connectivity status between the endpoints.

Payment: In some cases, the callee or the caller request some fee in order to accept or initiate a call. Examples include duration-based fees (similar to the PSTN); (fixed) fees relating to the (voice based) service a callee is offering, such as a support hotline; fees for calls a callee subscribed for, such as severe thunderstorm warning; or, in the case of Spam over Internet Telephony (SPIT) prevention, where a caller may be confronted with a small fee if its sincerity is in doubt [18], [20].

For whatever reason a session involves payment by at least one party, it is desirable to delay finalizing the payment transaction until connectivity is assured.

Reputation: Some approaches to detect and prevent SPIT use a reputation score in order to help determine the caller's nature [4], [20], [32]. Each user's reputation is related to its behavior and is calculated from several metrics that are collected by the providers. For examples, a short call duration may indicate an unsolicited call that prompted that callee to hang up immediately. Unfortunately, it may also indicate that at least one participant could not hear the other due to a lack of (bidirectional) media connectivity. In this case, the caller's reputation would falsely be reduced.

Forensics: In the area of law enforcement, reliable evidence is crucial. Regarding the question of whether or not a call took place, SIP can only provide information about signaling – if the phone rang, if the phone was picked up, and if the phone was hung up. This may not be sufficient: The information may be required as to whether or not the two parties in a call were actually able to communicate.

Call Detail Record Analysis: Call Detail Records (CDRs) are collected and analyzed for several reasons. These records contain information about each call, for example, the caller's and callee's IDs, the invitation time, the duration, and how the call terminated. This data can be used to conduct statistical analysis, to profile users' behavior, to reduce traffic congestion or, in general, to detect any kind of anomaly. It is not sufficient if the CDRs are based on the SIP messages only, without knowing whether or not there was media connectivity. This might result in contra-productive network configuration, misinterpretation of someone's reputation or, even worse, will black-list a participant.

In this paper we present a solution for the *VoIP Media Connectivity Awareness Problem*, which fulfills the following requirements:

1) Focus: It is the *SIP Provider* who needs to obtain knowledge about the connectivity status.

2) Multiple (bi-directional) streams: It is important to consider *all media streams* negotiated between the calling parties. Any single uni-directional stream that is not established successfully might be the reason for one of the participant to end the call prematurely. Thus, the provider needs to determine at least the connectivity status for the stream aggregate. For example, if any single media stream in any direction lacks connectivity, the stream aggregate is considered to have no connectivity.

3) Genuineness: In order to prevent false conclusions (and subsequent actions), the connectivity status gathered by the provider should be *genuine*.

4) Compatibility: The number of changes introduced into the SIP message sequences should be as small as possible. Ideally, neither extra SIP messages nor additional SIP headers should be required.

Parts of this paper have earlier been published in the Proceedings of the Tenth International Conference on Networks (ICN 2011) [1]. The focus was only on a single solution, dealing with the SIP providers' media connectivity awareness. In this article, we have extended the work presented in [1] in several ways. In detail, this article considers an extended range of related work, introduces and compares alternative approaches to solving the connectivity awareness problem, contains additional and improved call scenarios depicting the SIP message flows, presents specific protocol extensions to SDP for using the Stream Control Transmission Protocol (SCTP) as media transport, shows implementation details of the SIP Proxy message routing, and elaborates on measurement results.

This paper is structured as follows: In Section II, we discuss several existing approaches that have some relation to the awareness of media connectivity. In Section III, our favoured approach is presented. The section includes detailed scenarios, a preliminary investigation of using the SCTP for connectivity detection, a description of the proposed protocol extensions to SIP and SDP, and outlines several other possible approaches. Finally, Section IV presents a prototype SIP Proxy implementation of our solution and contains measurements of the performance overhead our solution introduces.

II. Related Work

There are some approaches that relate to the awareness of media connectivity, but which are motivated by different goals.

A. Dealing with the NAT

One possibility to solve the connectivity problem is the use of an ALG in addition to the NAT. In reality, however, ALGs are deployed in the fewest scenarios, even though most users manage their own private home networks. Furthermore, an ALG might increase the chance to achieve media connectivity, but the SIP provider still does not know about it.

In contrast to the UA-to-UA media connection, there is a very high chance to deliver all SIP messages by, e. g., traversing NAT using symmetric response routing [37]. Traversing the NAT for the media streams can be done using Interactive Connectivity Establishment (ICE) [33]. ICE describes NAT traversal for multimedia signaling protocols like SIP, and it extends the SDP [14] to convey additional data. In order to operate, ICE utilizes the protocols Session Traversal Utilities for NAT (STUN) [35] and Traversal Using Relays around NAT (TURN) [22].

The goal of ICE is to *establish* connectivity, but not to require it or to inform a third party of the connectivity status. This process of connectivity establishment is in principle independent of session establishment – a SIP session is allowed to be established successfully, even if there is no media connectivity.

B. Connectivity Preconditions

UAs may use Connectivity Preconditions as defined in RFC 5898 [3] to *verify* whether there is connectivity or not. Based on the concept of a SDP precondition in SIP as specified by RFC 3312 [8] (generalized by RFC 4032 [7]), the connectivity precondition defined by RFC 5898 tries to ensure that session progress is delayed (including suppression of alerting the called party) until media stream connectivity has been verified.

This approach is motivated by the separation of signaling and media path and its implications. Similar to a part of the solution described in this paper (see Section III), it enables the UAs to delay the SIP session establishment until connectivity is ensured. RFC 5898 has been published in July 2010 and does contain similarities to this paper, which we worked on at the same time. In contrast to our approach, the provider cannot enforce the UAs to make use of this extension. In addition, it does not inform a third party (such as the provider) of the connectivity status – neither implicitly nor explicitly. In particular, the provider is not aware of the media connectivity status.

Furthermore, RFC 5898 does not guarantee that session establishment comes along with media connectivity. In RFC 3312 (which is referenced by RFC 5898), alerting the user until all the mandatory preconditions are met has a “SHOULD NOT” semantics. According to the definition in RFC 2119 [6], this means that suspending session establishment is *not* guaranteed since the UA may have “*[…] valid reasons in particular circumstances when the particular behavior is acceptable or even useful […]*” [6, Sec. 4]. Even though the intentions of RFC 5898 and RFC 3312 are clear, the question remains if a provider interested in the connectivity status can rely on the information obtained from using Connectivity Preconditions.

C. Receiving RTCP information

RTP, the protocol used to transport the media data comes along with its own RTP Control Protocol (RTCP). This protocol is used between the RTP endpoints to send and receive statistical data about the quality of the received RTP streams. If the provider received these quality metrics, they could be used to *derive* a connectivity status for the corresponding RTP stream.

As a first possibility, a provider could “misuse” the solution suggested in [17] that tries to solve the “$x + 1$” RTCP port number problem with NATs. Due to the fact, that “it is even possible that the RTP and the RTCP ports may be mapped to different addresses” [17, p. 2] the RTCP streams could be redirected to the SIP provider who can analyze the incoming information and then forward the RTCP packets to the other endpoint.

As a second option, a provider can use the SIP Event Package for Voice Quality Reporting [27] to receive reports about the call’s quality metrics. The metrics are derived from the RTCP Extended Reports [12] and are reported to an interested third party using the SIP-specific event notification [31]. Using this mechanism, a provider can subscribe to the event with a UA in order to receive metric information periodically. This is done using the SIP request messages *SUBSCRIBE* and *NOTIFY* respectively.

In contrast to ICE and Connectivity Preconditions, in both mechanisms, the media connection’s information (either redirected RTCP packets or explicit signaling) is separate from the session establishment. Thus, a SIP session is established regardless of connectivity status, and the provider can then derive the connectivity status directly from the RTCP packets or event notifications. In addition, obtaining the connectivity status is not only separate from session establishment, but can only be done *after* the SIP session is established and *after* the media streams are set up. In fact, media must be sent first since RTCP packets (being a prerequisite) are not exchanged between the endpoints any earlier.

The dependence on RTCP introduces further issues. First of all, RTCP packets/RTCP Extended Reports may not arrive because there is no connectivity for the RTCP stream. However, this does not imply a lack of connectivity for the media stream since RTCP uses a different UDP port number (i. e., a different *transport address*) than the media stream. Packets may also not arrive because the other endpoint does not send them for some reason, even though there might be connectivity.

In addition, the quality metrics sent to the provider may be wrong because an endpoint deliberately falsified the information. Thus, when a provider uses the connectivity status to draw further conclusions (reputation, payment rollback, etc.), it needs to consider the trustworthiness of the information used to determine the connectivity status.

D. Disconnection Tolerance

Ott and Xiaojun [26] present mechanisms for detection of and recovery from temporary service failures for mobile SIP users.

For detection of connectivity loss, they suggest a media-based approach: Missing RTP packets, RTCP packets, or STUN packets along with some additional criteria are used as indicators that connectivity has been lost. If the connectivity loss persists longer ("call interruptions"), the UAs will automatically try to re-establish the session after locally terminating the session. For this purpose, the authors introduce the new SIP *Recovery* header field, which is set to `true` in the *INVITE* message used to re-establish the session.

By observing this header field, an in-route SIP proxy (and therefore the provider) has a way to know about connectivity loss in the previous session. However, the field contains no information about when the connectivity loss occurred. Finally, if the lack of connectivity persists even longer and automatic re-establishment fails ("call termination"), the system reverts to voice mail or instant messaging.

The focus of this paper is on obtaining the connectivity status during an ongoing session *after* the session has been established. Implicitly, it assumes that there was connectivity at the beginning of the session.

E. Conclusion

In all solutions presented except for the SIP Event Package for Voice Quality Reporting, the focus is always on the endpoints. Whether the main goal is to establish connectivity, ensure connectivity, detect/monitor connectivity status, or recover from connectivity loss, the assumption is always that the *endpoints* are the entities which are interested in the goal.

Hence, the provider is not aware of the media connectivity; and even when the connectivity information can be obtained, its validity and genuineness may be questionable.

III. Implicit Connectivity Detection and Notification

One major difference between the approaches presented above is *when* information pertaining to connectivity status is obtained. Three distinct cases can be identified: before session establishment (ICE, Connectivity Preconditions), after session establishment (Disconnection Tolerance [detection only], SIP Event Package for Voice Quality Reporting, RTCP attribute in SDP), and at the end of the conversation (Disconnection Tolerance [signaled through *Recovery* header field]). In the second case, the information can also be obtained continually during the ongoing session.

Another difference is found in the direction of a media stream for which connectivity status is determined and whether media streams are considered separately or jointly on a "session level." Most mechanisms distinguish between individual streams and, as streams are usually considered uni-directional, also between receiving and sending direction. Connectivity Preconditions distinguish both direction and individual streams, but the consequence (suspension of session establishment) is affected by the aggregate of the streams for which the precondition was requested. The Disconnection Tolerance solution disregards direction as symmetric connectivity is assumed; it also disregards individual streams because the existence of only one audio stream is assumed (point-to-point audio conversation).

In our approach, connectivity notification (Section III-A) and detection (Section III-B) is done before session establishment. Further, our solution regards both, different streams and directions. In addition, it ensures the genuineness of the connectivity status by considering missbehaving UAs (Section III-C).

A. Implicit Connectivity Notification

SIP itself already offers several possibilities to modify the message routing. For example, a SIP proxy can request to stay in the route of any SIP messages beyond those belonging the first SIP request. In order to achieve this, any UA sending a new SIP request needs to insert corresponding routing information. Thus, in contrast to a "normal" SIP session establishment, a proxy can become a mandatory node of the last SIP 3-way-handshake message, i. e., the *ACK* request (compare Section I, Figures 3, 4). Furthermore, the user agent server (UAS) does not necessarily need to send a *180 Ringing* response and notify the called person. Instead, it can respond with a *183 Session Progress* message to indicate further action prior to continued call processing.

This response message and the modified message routing can be combined with a modified UA behavior. By using the *183* response's payload, the callee can answer the caller's SDP offer. Thus, both parties know the parameters of all media sessions that will usually be established *after* the SIP session invitation has been accepted. Since the *183* now contains important information about the media session (SDP answer), it is crucial to ensure message reception at the caller's side. Fortunately with the Reliability of Provisional Responses [36], a mechanism exists that enables user agents to detect lost provisional responses and ensures their delivery by using special acknowledgements (*PRACK*) and retransmissions. We will not consider *PRACK* messages during the further investigation of our solution as this is beyond the scope of this paper. However, even without using

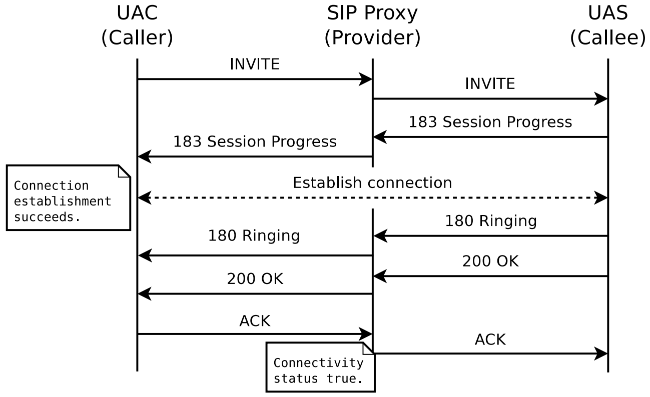


Figure 5: Accepted Call with Prechecked Media Connectivity

this mechanism, our solution will still work correctly with respect to determining the media connectivity status.

In our solution, the media sessions are established *beforehand*, and both parties **must hold back** the *180 Ringing*, *200 OK*, and the *ACK* messages until this has happened. Establishing the media sessions must involve some kind of connectivity detection mechanism, which will be considered in the next section. For simplicity, we refer to the establishment of all media sessions in their entirety as "connection establishment," where the connection establishment is considered successful when connectivity detection was positive for all media streams. With those rules, the *200 OK* and *ACK* messages act as connectivity confirmations by the UAS and user agent client (UAC), respectively. Furthermore, each UA **must ignore** any incoming media packets and **must not send** any media packets as long as the other endpoint did not confirm connectivity. This restriction enforces the connectivity status. It ensures that the reported connectivity status always matches the actual connectivity status as experienced by the user (genuineness requirement).

In result, the provider can conclude the media connectivity status by simply analyzing the messages it is routing (focus requirement). Therefore, we call the approach *implicit*. The provider will **conclude that there is connectivity if and only if** the UAS has sent a *200 OK* and then the UAC has sent an *ACK*.

In case the media connection could be established successfully, there will be a notification (*180 Ringing*), acceptance (*200 OK*) and acknowledgement (*ACK*) (see Figure 5). In result, the provider concludes that there is media connectivity.

If the UAS notices that establishing the media connection failed, it will reject the call by sending a *418* error response (see Figure 6). The latter is a new response code further explained in Section III-D. Since it is a final error response in the *4xx* category, even a UA who did not know about the new response code would consider SIP session establishment

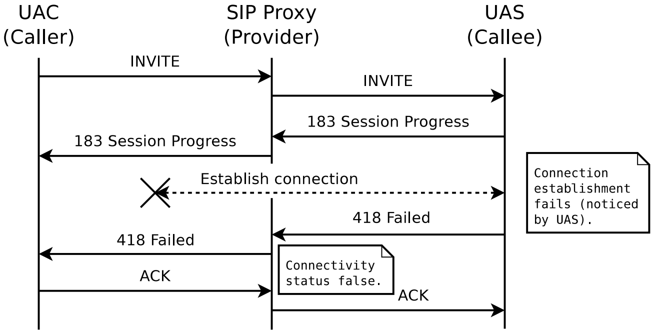


Figure 6: Call Abortion in the Case of no Media Connectivity, detected by the UAS

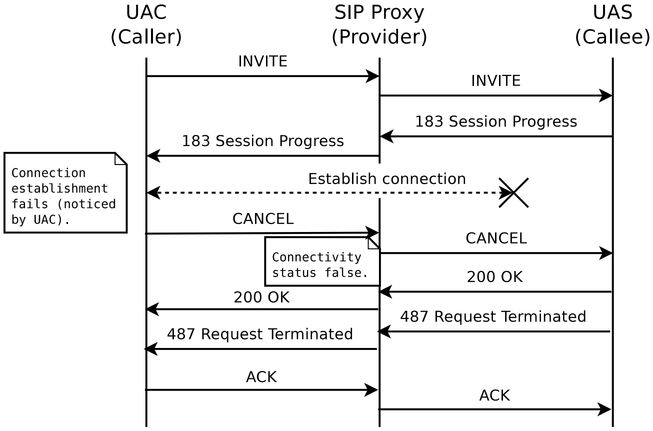


Figure 7: Call Abortion in the Case of no Media Connectivity, detected by the UAC

as failed and act appropriately. If the failure is detected by the UAC (see Figure 7), it will cancel the call using the *CANCEL* request causing the UAS to respond to the invitation with *487 Request Terminated*. In both cases, the provider concludes that there is no media connectivity.

In Figures 8 and 9, the media connection has been established successfully but the callee is unavailable. Thus, either the caller will *CANCEL* the call when a timeout appeared or the callee will response with a corresponding *408 Request Timeout* message. Again, the provider concludes lack of media connectivity.

B. Connectivity Detection

Due to the fact that the provider is simply analyzing the messages it is routing, it is up to the clients to verify the connectivity status. In detail, they need to check every single media stream for connectivity (multiple streams requirement), for example, by using STUN messages similar to the connectivity checks in ICE. This can be complex and time consuming.

In order to limit this overhead, we propose the use of the Stream Control Transmission Protocol (SCTP) [45] as the

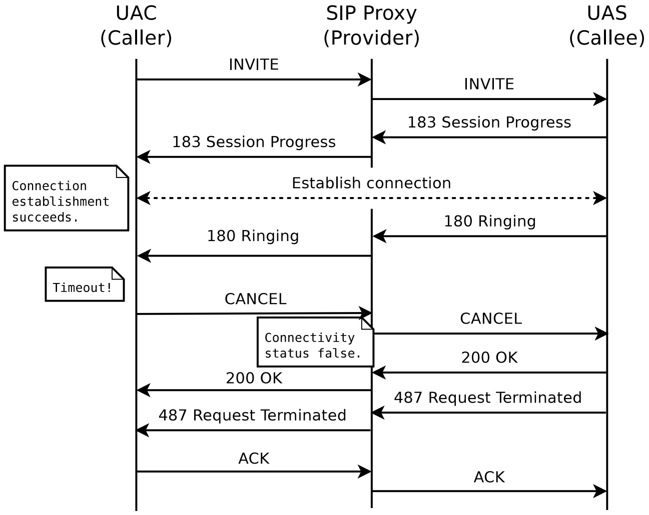


Figure 8: Timed-out Call with Prechecked Media Connectivity, detected by the UAC

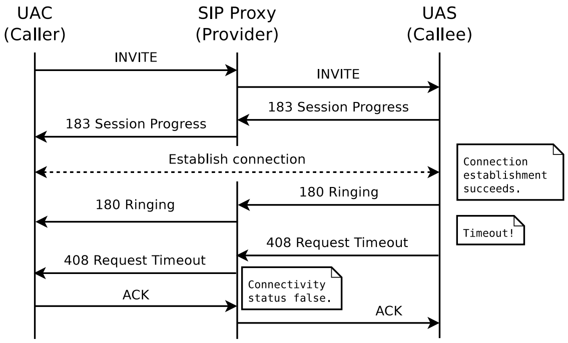


Figure 9: Timed-out Call with Prechecked Media Connectivity, detected by the UAS

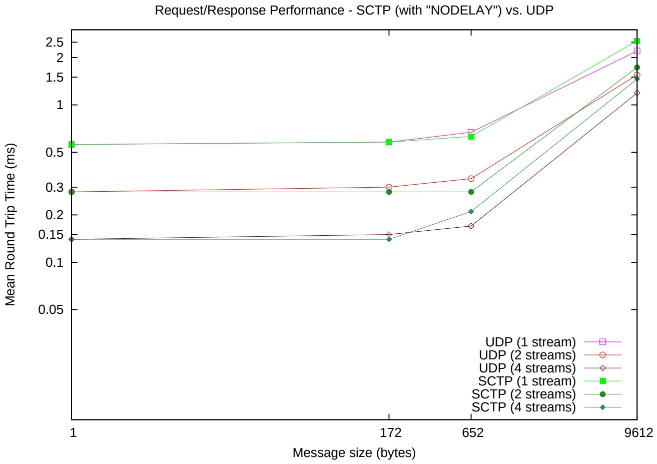


Figure 10: SCTP vs. UDP

media's underlying transport protocol. First of all, SCTP is connection oriented – SCTP's 4-way handshake at the beginning already ensures transport layer connectivity. In result, neither a media packet nor a notice of receipt need to be sent in order to check for connectivity.

Secondly, SCTP itself offers multiplexing; so there is no need for more than one connection, as every single RTP/RTCP stream can be sent using the same unique connection. In result, the time required to check each media stream (and media control stream) is reduced to a single check only. Last but not least, in contrast to TCP, SCTP offers unordered transport, meaning a lost packet does not delay delivery of succeeding packets. In addition, the partial reliable mode (SCTP Partial Reliability Extension [46]) can be used to improve the media quality in case a lost packet can be retransmitted immediately.

To confirm our proposal, we measured the SCTP performance in comparison to UDP. The environment consists of two machines with identical hardware and software running Debian GNU/Linux 5.0.3 (lenny) with kernel version 2.6.26 (i686). Both machines are equipped with an Intel Core 2 Duo E7500 dual core CPU running at 2.93 GHz and an Intel 82567LM-3 network adapter and connected via a FastEthernet switch (100 Mbps Full Duplex). The environment also determines the choice of UDP and SCTP implementations used – those of the Linux kernel. The benchmark itself is a ping-pong application that can send multiple messages at once, approximating multiple concurrent media streams. Figure 10 shows the mean round-trip time (RTT) in relation to the size of the messages. The sizes of 172 Bytes and 652 Bytes correlate to the RTP packet sizes produced by the G.711 codec using packet transmission cycles of 20 ms and 80 ms, respectively. One can see that the values of SCTP are very close to those of UDP, and hence, we expect no performance loss due to the use of SCTP.

C. *Missbehaving user agents*

In some cases, either the UAS or the UAC might try to falsify the information it tells about the connectivity status. Our solution requires sending a *200 OK* (UAS) or *ACK* (UAC) to convey "connectivity" or suppress those messages to convey that there is no connectivity.

For example, a caller might falsely announce "no connectivity" to the provider in order to send SPIT calls without consequences. In our solution, the UAC would have to suppress the *ACK* message. Fortunately, this would cause the callee to ignore any incoming media packets and to terminate the call by sending a *BYE* (see Figure 11). In case the UAS sends media packets instead of responding with a *200 OK*, the UAC is able to *CANCEL* the SIP session easily (see Figure 12).

In the payment example, the callee might say "connectivity" in order to receive his fee anyway. In this case, the caller receives a *200 OK* even though there is no

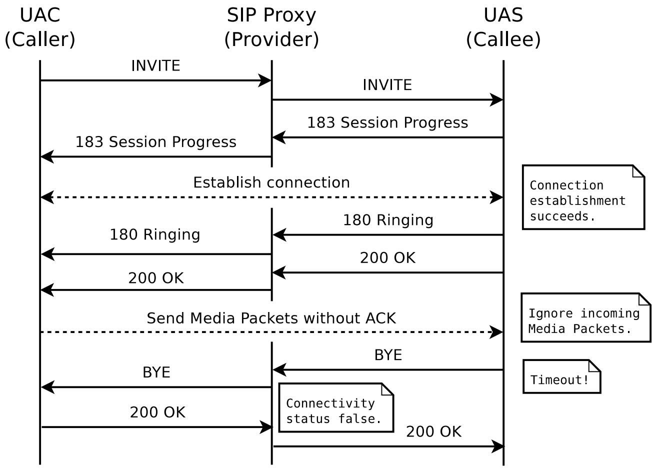


Figure 11: Media Delivery Without Prior *ACK*

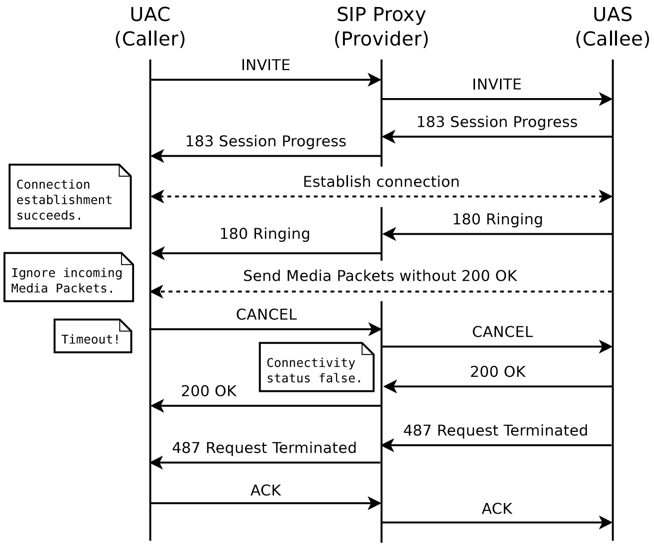


Figure 12: Media Delivery Without Prior *200 OK*

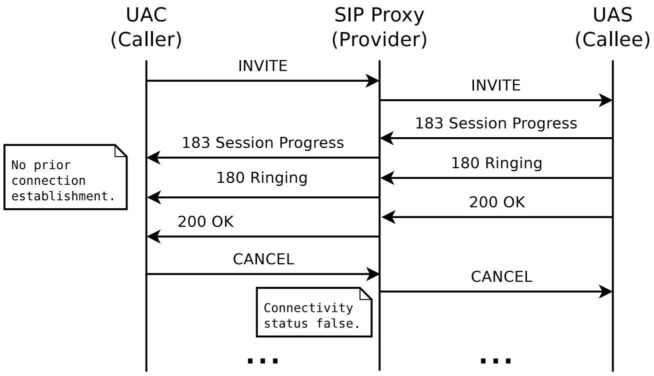


Figure 13: Immediate Session Acceptance Without Prior Connection Establishment

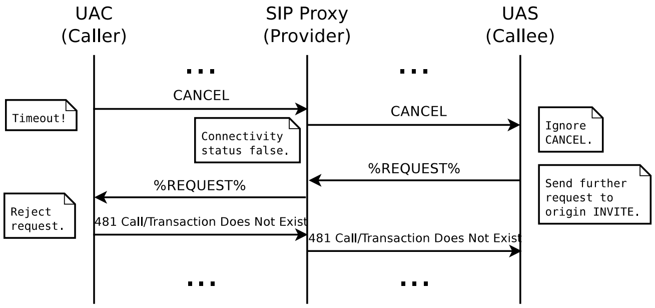

UAC
(Caller)
SIP Proxy
(Provider)
UAS
(Callee)
...
...
CANCEL
Timeout!
CANCEL
Connectivity status false.
Ignore CANCEL.
%REQUEST%
%REQUEST%
Send further request to origin INVITE.
Reject request.
481 Call/Transaction Does Not Exist
481 Call/Transaction Does Not Exist
...
...

Figure 14: Ignored CANCEL, but further Requests

media connectivity. According to the SIP specification, this final response needs to be acknowledged. Unfortunately, the provider would conclude connectivity, and thus, the caller needs to terminate the call without sending an *ACK* first.This can be achieved by sending a *CANCEL* request, causing a situation that – from the viewpoint of all other SIP entities – looks like the race condition described in RFC 5407 [15, Example 3.1.2.]. In result, the provider can conclude lack of connectivity and the callee is informed that the caller will not participate in the call any longer. The rest of this SIP message sequence is not shown since it depends on the implementation details of the UAs. In case the UAS has not implemented the correction to SIP suggested in RFC 6026 [42], the UAC may receive a *481 Call/Transaction Does Not Exist* instead of a *200 OK* to the *CANCEL*. Independently, the callee will not acknowledge any retransmitted *200 OK* to the *INVITE* pretending it never received these responses. In any case (even if timeouts are provoked), the provider's conclusion about missing connectivity will be correct.

In summary, for whatever reason a UA might missbehave – our solution enables the opponent party to react appropriately, enabling the provider to know the actual connectivity status. The general rule is: **Connection establishment before sending call acknowledgment, receiving call aknowledgment before connection usage** – otherwise, the call has to be rejected by adequate SIP messages.

The case that both, caller and callee, are lying cooperatively cannot be detected with our approach, but this is only a problem in the forensics scenario. It is doubtful, however, that the calling partners would use a provider at all to communicate in a criminal scenario.

Furthermore, a missbehaving callee might ignore the cancellation of a SIP session invitation and still send further requests and/or responses. These messages, however, can be rejected or ignored by the receiving caller (see Figures 14, 15). Similarly (not shown seperately), a callee can easily deal with further incoming SIP messages relating to a session that has previously been responded to with a final *4xx* client error response.

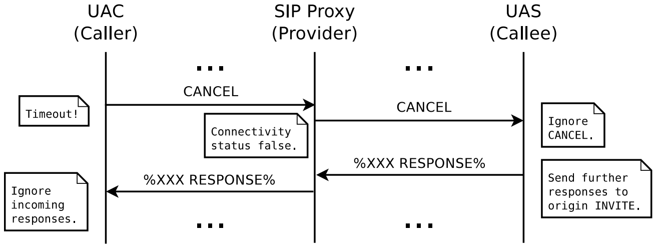


Figure 15: Ignored CANCEL, but further Responses

D. Protocol Extensions

There has been some work in the past for SCTP and SIP. Unfortunately the Internet-Draft by Fairlie-Cuninghame [11] is incomplete, inconsistent, and seems to have been abandoned. Another Internet-Draft by Loreto and Camarillo [21] tackles the same topic, but is very limited in its scope as it treats an SCTP association like a TCP connection in the sense that it completely ignores SCTP's multi-streaming feature. It is basically a one-to-one redefinition of RFC 4145 [47] for SCTP, specifying two additional protocol identifiers only. Similar to [11], it says nothing about how to use RTP over SCTP. This is different to our approach, since we do not use SCTP to deliver the SIP messages but to transport the media data. In order to simplify the media connectivity detection, all RTP streams are multiplexed by using a single SCTP connection.

In line with our compatibility requirement (see Section I), our solution only needs to slightly extend the abilities of SDP in order to specify the SCTP parameters. The use of the SCTP connection and the modified UA behavior can be indicated by identifying our extension – the SCTP Tunneling Extension for SIP – within a *Require* header by using a new option tag (”*sctp-tunnel´´*). If an incoming *INVITE* does not indicate usage of this extension the provider must reject this request by sending a *421 Extension Required* response. As described above, the extension just specifies the way the UAs must behave and the provider can draw conclusions; it does not specify any new SIP messages or headers – all of them have existed before.

We propose the following extensions to SDP in order to accomodate the use of SCTP as the transport protocol for media. First, we define the new “proto”-field value “SCTP/RTP/AVP”, which must be used in every session description “m=” line when the SCTP Tunneling Extension is used. “SCTP/RTP/AVP” denotes Real-time Transport Protocol (RTP) [40] used under the RTP Profile for Audio and Video Conferences with Minimal Control [39] running over SCTP [45]. We use the “c=” line to specify the IP address to which the SCTP tunnel should be established, which only has to be done once at the session level. Further, we define a new mandatory session-level attribute, “sctpPort”, which holds the port number to which the SCTP tunnel should be established. The syntax is defined in ABNF [9] as follows (cf. [11]):

```
sctpport-attribute = "sctpPort:" port
port               = 1*DIGIT
```

To accommodate multi-homed SCTP endpoints, we define a new optional session-level attribute, “sctpAddr”, that contains a list of IP addresses. It can be used to specify IP addresses that for establishing the SCTP tunnel in addition to the one specified in the “c=” line. The syntax in ABNF [9] can be defined as follows:

```
sctpaddr-attribute = "sctpAddr:"
    sctpaddr-elem *("," sctpaddr-elem)
sctpaddr-elem      =  nettype SP
    addrtype SP connection-address
```

In order to be able to specify SCTP stream numbers on which the endpoints expect to receive media packets for the various media streams, we redefine the notion of a “port” in SDP to mean “SCTP stream number.” The same rules how to assign port numbers for RTCP can be used for SCTP stream numbers. For example, even stream numbers are used for RTP and odd stream numbers for RTCP (cf. RFC 4566 [14, Sec. 5.14]). In general, whenever a SDP specification refers to a port number, this can simply be read as “SCTP stream number.”

To manage SCTP association establishment, the mechanims for TCP connection management defined in RFC 4145 [47] can be used analogously for SCTP. However, to allow for simultaneous association establishment, we extend the “setup” attribute with the value “simul”. Simultaneous association establishment – also called “initialization collision resolution” – can be useful to enable two endpoints that are behind different NATs to successfully establish an association.

The endpoints can now assume the following roles (defined in ABNF):

```
role  =  "active" / "passive" / "actpass"
         / "holdconn" / "simul"
```

where “simul” has the following semantics:

```
"simul": The endpoint is willing to
    accept an incoming connection, to
    initiate an outgoing connection,
    or to use simultaneous connection
    establishment (both endpoints will
    initiate the connection at the same
    time).
```

In the offer/answer model, “simul” gives the answering endpoint the option to choose among all options, so the answering endpoint can become active, passive, use simultaneous connection establishment, or the connection is not established for the time being.

Beyond SDP, the only syntactical extensions to SIP proposed in this paper (besides the behavioral definitions) are the additional option tag *sctp-tunnel* and the new response code number *418*.

Option tags define identifiers for SIP extensions and their use in the *Require* and *Supported* header fields. They are registered by the IANA under the “Session Initiation Protocol (SIP) Parameters” registry under the “Option Tags” sub-registry. The *sctp-tunnel* option tag can be defined as follows:

Name:

sctp-tunnel

Description:

This option tag is for tunneling all media streams between two endpoints of a SIP session through a single SCTP association as specified in the SCTP Tunneling Extension for SIP. When present in the *Supported* header field, it indicates that the UA is able to use the extension. When present in the *Require* header field, it indicates that UAC and UAS MUST use the SCTP Tunneling Extension and follow the rules specified therein.

Response codes are registered by the IANA under the “Session Initiation Protocol (SIP) Parameters” registry under the “Methods and Response Codes” sub-registry. The response code is defined as follows:

Response Code Number:

418

Default Reason Phrase:

SCTP Association Initialization Failed

The code can be used by the user agents to cause the *INVITE* request to fail when the SCTP tunnel cannot be established. Depending on which of the UAs “notices” that establishment failed (for example by using timeouts), the response code can be used directly as a failure response (by the UAS) or as a cause parameter in the *Reason* header field of a *CANCEL* request (by the UAC).

E. Further Approaches

Since it fulfills all requirements introduced at the beginning of this paper, we prefer the approach described above. Nevertheless, one might think of further approaches to solve the VoIP Media Connectivity Awareness Problem.

1) Media Gateway: Obviously, a provider could act as a media gateway, meaning all media data will be routed through a dedicated network component under its control. Since the gateway resides on the public Internet and media packets travel only between gateway and endpoint (not between endpoints directly), the typical connectivity restrictions caused by NATs and firewalls do not apply. Therefore – being an active part of all RTP streams – the provider can easily determine the media connectivity status. Setting up such a VoIP/SIP gateway can be done easily by using Asterisk [43], for example.

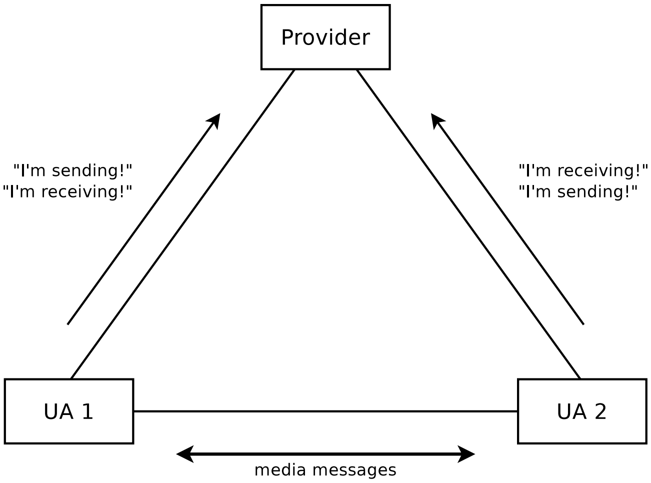


Figure 16: General View of Explicit Notification

Besides our “focus” requirement (see Section I), this solution fulfills the requirements multiple (bi-directional) streams, genuineness, and compatibility as well. On the other hand, this approach does not conform to the peer-to-peer architecture of VoIP infrastructures based on SIP. In addition, it requires a considerable amount of computational and network resources. In reality, VoIP services are often offered for free, and thus, providing these resources would not be economical.

2) Explicit Connectivity Notification: Besides an implicit notification, the UAs also can inform the provider about the connectivity status explicitly. In detail, the UAs send information about the connectivity status of every media stream they send and/or receive media packets on. A general view of this behaviour is depicted in Figure 16.

Such notifications can be implemented using *SIP-Specific Event Notification* [31], whereas, an *event package* needs to be defined that specifies the exact behavior of UAs subscribing to events and reporting events as well as syntax and semantics of the *NOTIFY* and *SUBSCRIBE* messages.

First, the provider subscribes to the *connectivity event* by sending a *SUBSCRIBE* message to the UAs. “The provider” could be the same SIP element as the SIP proxy, but it could also be a separate server belonging to the provider that communicates with the proxy. In the following section, the SIP element receiving and processing the connectivity notifications is called *connectivity server (CS)*.

For each stream where the UA is in the sender role (bi-directional and send-only streams), the UA notifies the CS as soon as it starts sending on that stream. Since this typically happens for most or all streams at the same time, the notifications for several streams can be sent in *one NOTIFY* message.

For each stream where the UA is in the receiver role (bi-directional and receive-only streams), the UA notifies the CS as soon as it receives the first packet on that stream. This

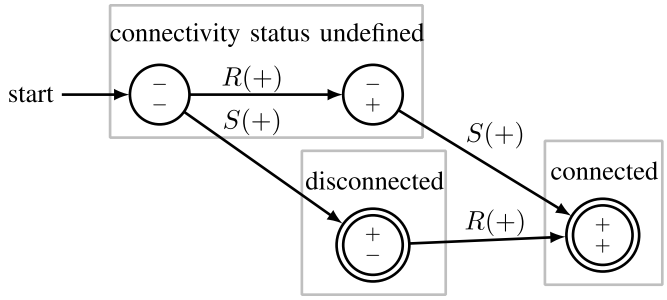


Figure 17: State Machine of Connectivity Status Calculation ($S(+)$ represents sender's sending notification, $R(+)$ represents receiver's receiving notification)

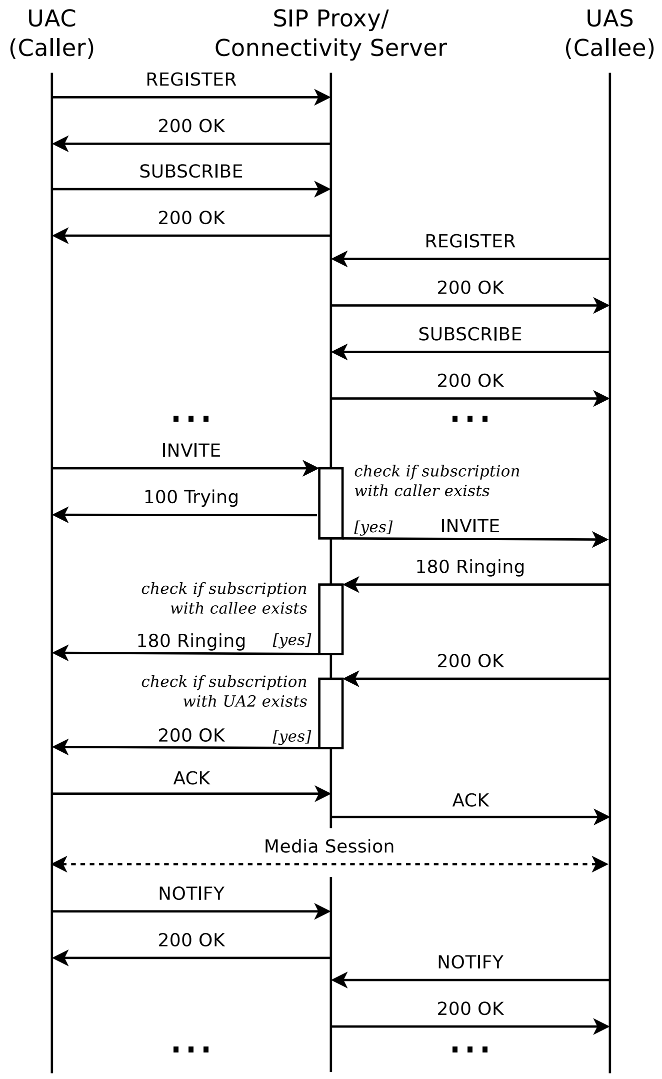


Figure 18: SIP Sequence of Explicit Notification

will potentially also happen roughly at the same time for most streams, so the UA can buffer the event for a short period of time in order to, again, include the notifications for several streams in *one NOTIFY* message.

The CS keeps a state table that holds the sender and receiver state (whether or not the sender is sending and whether or not the receiver is receiving) for each active media stream in the session (inactive and rejected streams are not relevant). Since the CS needs to know when a session is established and terminated, it has to communicate with the SIP proxy responsible for the call. It also needs to have a current description of the session, i. e., which streams with what directions have been successfully negotiated and which streams are inactive. In the end, the CS matches up the two notifications for each stream of the media connection and determines its connectivity status (see Figure 17).

In order to be able to determine the connectivity status, the CS needs to make sure that it will receive connectivity notifications from both endpoints involved in a call. To achieve this, the SIP proxy routing the call must delay forwarding any *INVITE* requests until it has successfully subscribed to the connectivity event with *both* UAs. If subscribing fails, the *INVITE* must be rejected. Once a subscription has been made, there should be little difficulty receiving the notifications: NATs or firewalls are unlikely to block notifications since the process of subscribing represents a two-way handshake (*SUBSCRIBE*, *200 OK*) and therefore ensures signaling connectivity between CS and UAs.

Figure 18 shows a message flow between two UAs belonging to the same provider. In this example, the connectivity server and the SIP proxy are the same SIP element. Both UAs register with the proxy first, after which the CS subscribes to the connectivity notifications. Then, a session is established successfully, after which both UAs send their connectivity notifications.

In case a callee is registered at a different SIP provider, it is considerably more complex to subscribe to the connectivity notification. In addition, provider and endpoints have a significant amount of additional work to do compared to a "regular" SIP session, especially the provider. Whereas each endpoint has to perform the subscription procedure, monitor incoming messages on all media streams, and compile and send the connectivity notification(s), the provider has to perform and manage subscriptions for a potentially great number of users, keep additional state for every SIP session (state machine), and do calculations for every media stream in every session. In addition, the provider has to keep an explicit history with the connectivity status for every completed SIP session in order to draw conclusions from the connectivity statuses later. This task is not necessary with our first approach (implicit connectivity notification), since there, every successful SIP session establishment implies that there was connectivity.

Furthermore, this approach does not fulfill the requirement of genuineness. The connectivity status is only correct under the assumption that both endpoints are generating the notifications truthfully. In contrast to the implicit approach, this explicit approach does not include a mechanism to enforce the connectivity status. It would be difficult to introduce such mechanism for two reasons: First, there is no feedback to the UAs about the connectivity status that would enable them to act appropriately (stop sending media or ignoring incoming media). Second, the nature of the connectivity detection (watching for incoming media packets) requires the active use of the media channel *before* the connectivity status is known. So essentially, a reaction would always be too late.

In addition, the assumption is made that notifications are not blocked or altered in transit to the CS and that they are coming from the correct endpoint – additional measures would be required to verify the authenticity of endpoints and the integrity of messages.

3) Connectivity Verification with Secret Tokens: Instead of concluding the connectivity status out of UAs notifications, a provider can test each media connection indirectly by using secret tokens initially known only to the provider. Modifying the messages of the signaling channel, the provider can send these tokens to the UAs. As a second step, the provider expects the UAs to relay these tokens using the media connections. Thus, the corresponding calling partner receives the tokens in case the media connection is set up correctly. Finally, the received tokens will be transmitted to the provider in order to enable each corresponding connectivity verification.

Figure 19 shows a typical message flow for the whole process of connectivity verification of a single bi-directional media stream. The *INFO* method is used for the UA to proxy communication. Note that the *200 OK* to the *INVITE*, the corresponding *ACK*, and further additional provisional responses are not shown in the diagram. Similar to the implicit approach, a *183 Session Progress* message is required to obtain the media parameters (used to send the tokens) before connection establishment will be acknowledged via *200 OK* and *ACK*.

Unfortunately, receiving a wrong or, even worse, no token does not necessarily imply lack of connectivity since a participant might lie by falsifying or suppressing the token.

Nevertheless, this approach fulfills all requirements. However, the mechanism would become very complex, in case the participants are registered at different providers and both providers want to verify connectivity, since the tokens need to be mapped to their respective creators.

Furthermore, the provider needs to identify every media stream to be used in order to generate the corresponding amount of tokens, timeouts have to be defined, the message format needs to be specified, a new SIP header is required to carry the information, and the UAs' behavior needs to be ensured.

It is questionable if this solution fully fulfills our compatibility requirement. On the media transport level, the RTP media streams are "misused" by transporting non-media data. On the signaling level, the SIP *INFO* request is not used in the protocol intended way, since it is actually an end-to-end message and thus needs to be "illegally" intercepted by the proxy in order to process it (instead of forwarding it to the calling partner). If the SIP-specific event notification was used instead, similar problems to the explicit notification approach would arise.

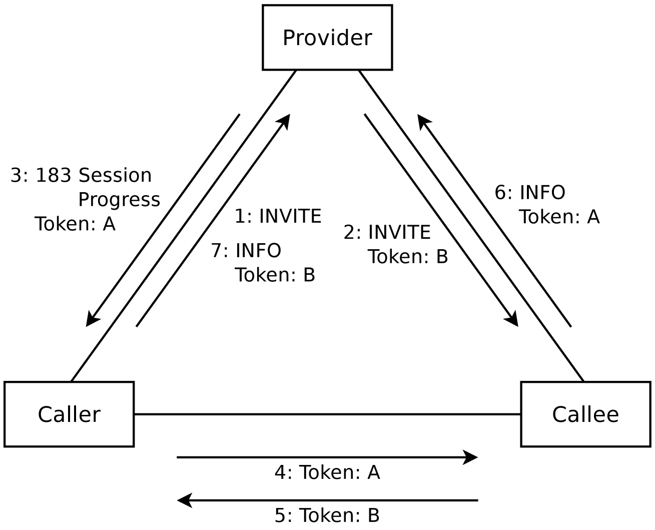


Figure 19: Connectivity Verification with Random Numbers

F. Summary

Compared to all other approaches, the explicit connectivity notification approach has one major drawback: It does not fulfill the genuineness requirement because it cannot guarantee that the connectivity status seen by the provider is genuine – an endpoint can lie. All other approaches satisfy this requirement in the sense that one endpoint alone can never falsify the connectivity status seen by the provider. The key to this feature is the concept of *enforcing* the connectivity status: Each endpoint knows the connectivity status claimed by the other endpoint and only exchanges media packets if and only if this claims states that there is connectivity.

All other requirements, on the other hand, are satisfied by all the approaches presented.

However, the alternative approaches in general require much more resources and efforts to put these solutions into practice. Considering each approach's costs added by the additional behavior, the way SIP and other involved protocols are extended or altered, the effort required for implementing the mechanism, and each applicability, the implicit notification approach presents itself as the most promising solution. Thus, this approach is the focus of our research.

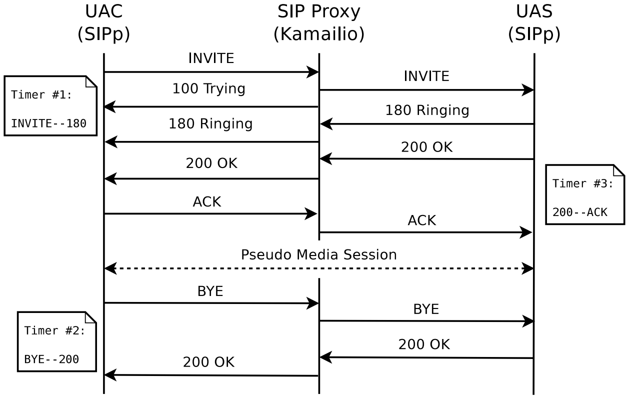


Figure 20: Measurement Scenario

IV. Measurements

Although we minimized the changes to the existing VoIP infrastructure, the provider still has to be aware of the *sctp-tunnel* extension indicated within the SIP messages. Whether the extension is stated or not, the provider has to use different message handling and routing. It is thus important to know how much network and computational overhead our extension creates on the provider's side and how much more UA-to-UA time the message routing takes.

Note that the following measurements do not consider the impact of SCTP. SCTP is used as the underlying transport protocol of the UA-to-UA media session only, whereas SIP messages still use UDP. In result, a SIP proxy does not need to be adapted to use another transport protocol. On the other hand, media gateways (not considered by the following measurements) need to be altered to conform to our approach.

A. Testbed, Scenarios

We used three nodes (each with 2 x AMD Opteron 244 CPU (1.8 GHz), 4 GB RAM, Gigabit Ethernet Interconnection) to setup one SIP proxy (Kamailio [25], v3.0.3) and two UAs (SIPp [13], v3.1) that generated and processed a various number of SIP calls. Kamailio has been configured to use 1024 MB of memory, to create four processes, and its log level was set to zero.

In general, we measured three scenarios: a) default behavior of the proxy, b) modified behavior of the proxy where the UAs already indicated the use of the *sctp-tunnel* extension, and c) the modified behavior of the proxy without initial indication by the UAs. The third scenario is the most expensive one since the provider needs to reject incoming invitations first, and then has to deal with the reformulated ones. In addition, we measured d) the SIPp-SIPp-interconnectivity to determine the overhead of Kamailio in general.

In scenarios a) and b), the UAC and the UAS send and receive SIP messages according to Figure 20. According to Figure 4, the proxy stays in the route for the whole call.

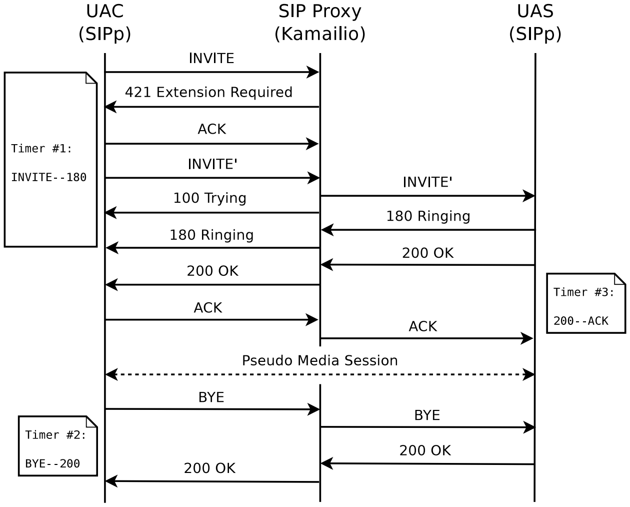


Figure 21: Measurement Scenario with Enforced Use of Extension

Scenario c) requires three more messages at the beginning: the first *INVITE* will be rejected with a *421* response that has to be *ACK*ed (see Figure 21).

In order to implement the modified proxy behavior, we used a prototypical approach that only required modification of the Kamilio routing logic that is defined in the Kamilio configuration file (`kamailio.cfg`). The relevant excerpts from the file are shown in Listing 1. The implementation checks if the *Require* header is present and searches for the option tag that identifies the SCTP Tunneling Extension. If it is not present (Figure 21), it sends the *421* response attaching the *Require* header with the appropriate option tag indicating which extension is required. If the *Require* header with the correct option tag is present, routing proceeds as usual (Figure 20; see the first **return** statement in Listing 1).

Each call generates three round-trip time values: RTT #1 represents the delay of a UAS's response including Kamailio action (such as lookup and extension verification); RTT #2 represents the delay of a UAS's response in case the request can be forwarded immediately; RTT #3 represents the delay of a UAC's feedback. Each series lasted five minutes, using a constant call frequency (between 1 and 1000 calls per second). The proxy and the UAs were restarted for each frequency.

B. Results

For all scenarios and each frequency, we calculated the corresponding median and quartile values for each RTT. As expected, in scenarios a)–c), the values of RTT #2 and RTT #3 are nearly the same (see Figures 23, 24). The SIPp-SIPp interconnection's second and third RTT are ~0.25–0.55 ms lower only.

The comparison of RTT #1 is shown in Figure 22. Again, one can see the additional time required (~0.5–0.6 ms) when

```
#!KAMAILIO
#!define WITH_SCTP_TUNNELING

/* other defines, parameters,
 * and module configuration */

####### Routing Logic ########

# main request routing logic

route{
  /* processing of related requests */

  # make sure UAC is using the
  # SCTP tunneling extension
  if (is_method("INVITE")) {
        route(REQUIRE);
  }

  /* processing of initial requests */

  /* other processing */
}

/* other route blocks */

route[REQUIRE] {
#!ifdef WITH_SCTP_TUNNELING
  if (is_present_hf("Require")) {
    if (search("^Require:.*sctp-tunnel.*")) {
      return;
    }
  }
  append_to_reply("Require:␣sctp-tunnel\r\n");
  send_reply("421", "Extension␣Required");
  exit;
#!endif
  return;
}
```

Listing 1: SIP Proxy Implementation: Kamailio Configuration File (kamailio.cfg)

Kamailio is put between the SIPp instances. Furthermore, we expected the overhead of the header verification to be very small since we only slightly modified the routing logic of Kamailio. This small RTT increase can be seen when comparing the values of scenarios a) and b).

In scenario c), where the proxy had to enforce the use of the SIP extension, RTT #1 increases a little more. This happens because Kamailio is involved one more time and three more messages are sent until the first callee's response is received by the inviting caller.

V. Conclusion and Future Work

In this paper, we presented several scenarios motivating the need for media connectivity awareness for SIP providers, including payment, reputation, forensics, and call detail record analysis. We identified specific requirements a solution must fulfill. Two requirements should be emphasized. The derived connectivity status must be genuine. This is important because the provider uses the obtained information

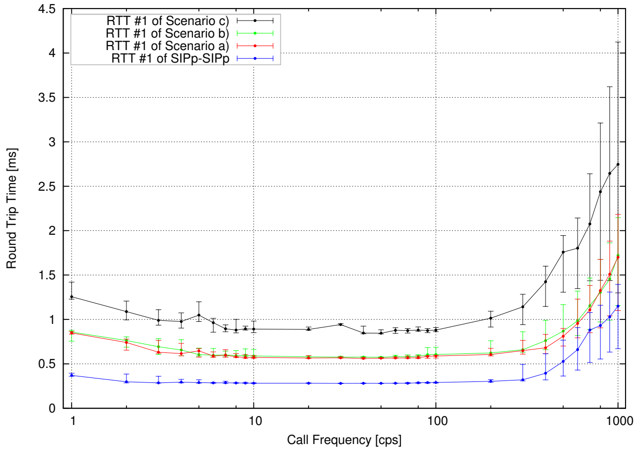


Figure 22: Comparison of RTT #1

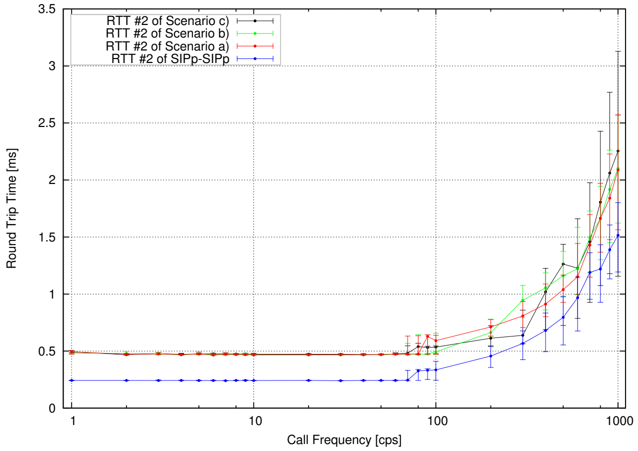


Figure 23: Comparison of RTT #2

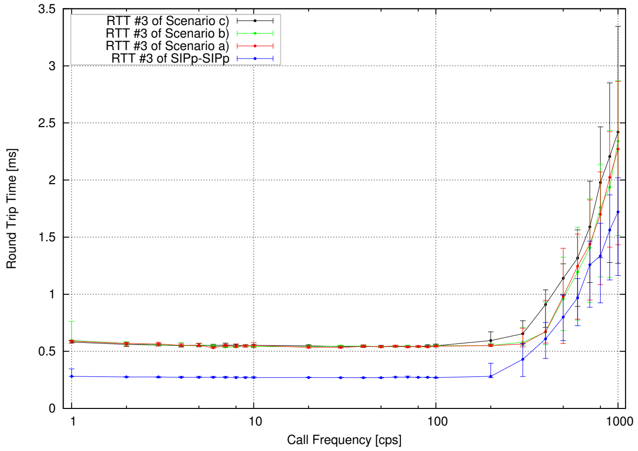


Figure 24: Comparison of RTT #3

to draw futher conclusions that potentially impact the user directly, like demoting a user's reputation regarding SPIT. It is similarily important that the solution is compatible with existing protocols as it would otherwise not be applicable in existing VoIP infrastructures.

In our solution, the provider is *implicitly* informed about the media connectivity: The SIP provider can draw genuine conclusions by simply analyzing the messages it is routing. The UA, however, needs to alter its behavior. This behavior is specified by way of a new SIP extension and its usage can be enforced by the provider.

To reduce the overhead of media connectivity detection, we propose to use SCTP for media transport. This requires a slight extension of SDP.

The measurements showed that the overhead introduced by our solution is negligible, as long as the UAs indicate the use of our extension from the beginning. In addition, our approach can easily be integrated into existing VoIP infrastructures as it fully conforms to existing protocols. If a UA is not aware of our extension it is at the discretion of the provider to proceed with the call (without the ability to conclude media connectivity) or to reject it.

Several other approaches to the connectivity awareness problem are presented and contrasted in this paper. We favour the implicit approach as it requires the least changes to the involved protocols, minimizes protocol overhead, and ensures that the connectivity status is genuine even if a user agent lies.

Even though none of the related work presented in Section II by itself enables a SIP provider to gain awareness of the media connectivity status, one work – Connectivity Preconditions [3] – could achieve this with a slight modification and combination with parts of our approach. The semantics of the Connectivity Preconditions SIP extension would have to be altered from "SHOULD" to "MUST" and the SIP Proxy would have to be able to enforce the use of the precondition. In addition, the behavioral rules for the user agents introduced in Section III would have to be observed. Lastly, each UA would have to verify connectivity of every single media stream in both directions.

Future work will deal with Quality of Service (QoS) aspects. Besides a lack of connectivity, low quality can also cause a call to be aborted prematurely by one of the participants. We therefore need to conduct further investigation in order to deal with this problem.

In addition, as the use of reliable provisional responses would be beneficial to our solution (cf. Section III-A), protocol interactions and possible implications need to be investigated.

References

[1] Stefan Gasterstädt, Markus Gusowski, and Bettina Schnor. SIP Providers' Awareness of Media Connectivity. In Pascal Lorenz, Tibor Gyires, and Iwona Pozniak-Koszalka, editors, *10th International Conference on Networks (ICN 2011)*, pages 157–163. IARIA, January 23rd–28th, 2011.

[2] Alessandro Amirante, Simon Pietro Romano, Kyung Hwa Kim, and Henning Schulzrinne. Online Non-Intrusive Diagnosis of One-Way RTP Faults in VoIP Networks Using Cooperation. In Georg Carle, Helmut Reiser, Gonzallo Camarillo, and Vijay K. Gurbani, editors, *Proceedings of the 4th International Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm 2010)*, pages 153–160, Munich, Germany, August 2nd–3rd, 2010. Technical University Munich.

[3] F. Andreasen, G. Camarillo, D. Oran, and D. Wing. Connectivity Preconditions for Session Description Protocol (SDP) Media Streams. RFC 5898 (Proposed Standard), July 2010.

[4] Vijay A. Balasubramaniyan, Mustaque Ahamad, and Haesun Park. CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation. In *Proceedings of the 4th Conference on Email and AntiSpam, CEAS 2007*, August 2nd–3rd, 2007.

[5] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard), January 2005.

[6] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119 (Best Current Practice), March 1997.

[7] G. Camarillo and P. Kyzivat. Update to the Session Initiation Protocol (SIP) Preconditions Framework. RFC 4032 (Proposed Standard), March 2005.

[8] G. Camarillo, W. Marshall, and J. Rosenberg. Integration of Resource Management and Session Initiation Protocol (SIP). RFC 3312 (Proposed Standard), October 2002. Updated by RFCs 4032, 5027.

[9] D. Crocker and P. Overell. Augmented BNF for Syntax Specifications: ABNF. RFC 5234 (Standard), January 2008.

[10] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176.

[11] R. Fairlie-Cuninghame. Guidelines for specifying SCTP-based media transport using SDP. Internet-Draft draft-fairlie-mmusic-sdp-sctp-00, Internet Engineering Task Force, May 2001. Work in progress.

[12] T. Friedman, R. Caceres, and A. Clark. RTP Control Protocol Extended Reports (RTCP XR). RFC 3611 (Proposed Standard), November 2003.

[13] Richard Gayraud, Olivier Jacques, et al. SIPp: An Open Source Performance Testing Tool for SIP [v3.1]. http://sipp.sourceforge.net, retrieved: January 22nd, 2012.

[14] M. Handley, V. Jacobson, and C. Perkins. SDP: Session Description Protocol. RFC 4566 (Proposed Standard), July 2006.

[15] M. Hasebe, J. Koshiko, Y. Suzuki, T. Yoshikawa, and P. Kyzivat. Example Call Flows of Race Conditions in the Session Initiation Protocol (SIP). RFC 5407 (Best Current Practice), December 2008.

[16] M. Holdrege and P. Srisuresh. Protocol Complications with the IP Network Address Translator. RFC 3027 (Informational), January 2001.

[17] C. Huitema. Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP). RFC 3605 (Proposed Standard), October 2003.

[18] C. Jennings, J. Fischl, H. Tschofenig, and G. Jun. Payment for Services in Session Initiation Protocol (SIP). Internet-Draft draft-jennings-sipping-pay-06, Internet Engineering Task Force, July 2007. Work in progress.

[19] Alan B. Johnston. *SIP: Understanding the Session Initiation Protocol*. Artech House, 2^{nd} edition, 2004.

[20] Stefan Liske, Klaus Rebensburg, and Bettina Schnor. SPIT-Erkennung, -Bekanntgabe und -Abwehr in SIP-Netzwerken. In David Buchmann and Ulrich Ultes-Nitsche, editors, *15. ITG/GI-Fachtagung "Kommunikation in Verteilten Systemen" (KiVS 2007) – Workshop "Secure Network Configuration" (NetSec 2007)*, pages 33–38, Fribourg, February 26th–March 2nd, 2007. DIUF, Universität Fribourg.

[21] S. Loreto and G. Camarillo. Stream Control Transmission Protocol (SCTP)-Based Media Transport in the Session Description Protocol (SDP). Internet-Draft draft-loreto-mmusic-sctp-sdp-05, Internet Engineering Task Force, February 2010. Work in progress.

[22] R. Mahy, P. Matthews, and J. Rosenberg. Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). RFC 5766 (Proposed Standard), April 2010.

[23] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936.

[24] P.V. Mockapetris. Domain names - implementation and specification. RFC 1035 (Standard), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966.

[25] Ramona-Elena Modroiu, Bogdan Andrei Iancu, Daniel-Constantin Mierla, et al. Kamailio (OpenSER) [v3.0.3]. http://www.kamailio.org/, retrieved: January 22^{nd}, 2012.

[26] Jörg Ott and Lu Xiaojun. Disconnection tolerance for SIP-based real-time media sessions. In *MUM '07: Proceedings of the 6^{th} international conference on mobile and ubiquitous multimedia*, pages 14–23, New York, NY, USA, 2007. ACM.

[27] A. Pendleton, A. Clark, A. Johnston, and H. Sinnreich. Session Initiation Protocol Event Package for Voice Quality Reporting. Internet-draft, Internet Engineering Task Force, Mar. 2010. Work in progress.

[28] J. Postel. User Datagram Protocol. RFC 768 (Standard), August 1980.

[29] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[30] J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFCs 1122, 3168, 6093.

[31] A. B. Roach. Session Initiation Protocol (SIP)-Specific Event Notification. RFC 3265 (Proposed Standard), June 2002. Updated by RFCs 5367, 5727.

[32] J. Rosenberg. The Session Initiation Protocol (SIP) and Spam. Internet-Draft draft-ietf-sipping-spam-03, Internet Engineering Task Force, October 2006. Work in progress.

[33] J. Rosenberg. Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols. RFC 5245 (Proposed Standard), April 2010.

[34] J. Rosenberg and G. Camarillo. Examples of Network Address Translation (NAT) and Firewall Traversal for the Session Initiation Protocol (SIP). Internet-Draft draft-rosenberg-sipping-nat-scenarios-03, Internet Engineering Task Force, July 2004. Work in progress.

[35] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing. Session Traversal Utilities for NAT (STUN). RFC 5389 (Proposed Standard), October 2008.

[36] J. Rosenberg and H. Schulzrinne. Reliability of Provisional Responses in Session Initiation Protocol (SIP). RFC 3262 (Proposed Standard), June 2002.

[37] J. Rosenberg and H. Schulzrinne. An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing. RFC 3581 (Proposed Standard), August 2003.

[38] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141.

[39] H. Schulzrinne and S. Casner. RTP Profile for Audio and Video Conferences with Minimal Control. RFC 3551 (Standard), July 2003. Updated by RFC 5761.

[40] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761, 6051, 6222.

[41] D. Senie. Network Address Translator (NAT)-Friendly Application Design Guidelines. RFC 3235 (Informational), January 2002.

[42] R. Sparks and T. Zourzouvillys. Correct Transaction Handling for 2xx Responses to Session Initiation Protocol (SIP) INVITE Requests. RFC 6026 (Proposed Standard), September 2010.

[43] Mark Spencer et al. Asterisk – The Open Source Telephony Projects. http://www.asterisk.org/, retrieved: January 22^{nd}, 2012.

[44] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT). RFC 3022 (Informational), January 2001.

[45] R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Proposed Standard), September 2007. Updated by RFC 6096.

[46] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad. Stream Control Transmission Protocol (SCTP) Partial Reliability Extension. RFC 3758 (Proposed Standard), May 2004.

[47] D. Yon and G. Camarillo. TCP-Based Media Transport in the Session Description Protocol (SDP). RFC 4145 (Proposed Standard), September 2005. Updated by RFC 4572.

A Didactic Platform for Practical Study of Real Time Embedded Operating Systems

Adam Kaliszan
Chair of Communication and Computer Networks
Poznan University of Technology
ul. Polanka 3, 60-965 Poznań, Poland
Email: adam.kaliszan@gmail.com
http://www.adam.kaliszan.yum.pl

Mariusz Głąbowski
Chair of Communication and Computer Networks
Poznan University of Technology
ul. Polanka 3, 60-965 Poznań, Poland
Email: mariusz.glabowski@put.poznan.pl
http:// glabowski.eu

***Abstract*—The article proposes a new didactic platform for practical study of embedded Real Rime Operating Systems (RTOSs). Three fundamental parts that are included in the platform are discussed in detail: the hardware part, the firmware part and the software tools. In the description of the hardware part the following parts are addressed: main controller, input/output module, executing module and programmer module. The project of the hardware part is distributed according to GPLv2 license. The firmware of the platform is based on FreeRTOS distributed according to the modified GPL license, ported by the authors on the microcontrollers not originally supported, i.e., Atmega128 and Atmega168. The firmware part of the platform proposed and described in the article implements: the command line interpreter, file system, the protocol for communication between main controller and executing modules, TCP/IP stack and xModem protocol, among others. All the software tools work on the Linux operating system and are free of charge; most of them have open source code. Particular attention is given to a presentation of laboratory exercises that have been worked out in the process. These exercises are designed to facilitate the learning process in the study of embedded operating systems with the application of the proposed didactic platform. The proposed platform is not expensive and is easy to assemble. Most students can afford to build or modify it on their own.**

***Keywords*-Embedded systems; Real Time Operating System; Multitasking; Interprocess communication; Intelligent home.**

I. Introduction

Practice is an important addendum to any embedded operating systems theory course [1]. The practical part of the course is often conducted with the help of one of the existing operating systems, usually Linux or Windows. Linux has certain advantages, such as its versatility, ranging from small embedded devices to powerful supercomputers. Thanks to Linux open source code, there are many written kernel modules [2] supporting new devices, which ensures such a great versatility of the system and makes it applicable in many embedded systems. Microsoft, in turn, offers different versions of its own operating system, ranging from Windows CE or Windows Mobile that are working on mobile phones, PDA devices and car navigation, to Windows Server [3]. On account of Microsoft .NET framework, it is possible to write software in a very easy way. However, it should be noted that the software produced by Microsoft is not free. Additionally, the fact that its code is closed complicates porting the operating system to new, not particularly common, hardware devices. Hence, its application is limited to a few basic CPU architectures.

Irrespective of a chosen operating system, the practical part of an operating systems course is often limited to learning the basis of operating systems, i.e., learning Linux fundamental commands such as creating and removing files or directories, changing file attributes and launching applications. Such laboratory classes do not introduce the subject of embedded systems, nor do they have any connection to the operating system theory, since most laboratories do not cover topics such as multitasking, interprocess communication and its synchronization or operations on file systems. Furthermore, as it is often the case, proposed laboratory exercises in operating systems have little relevance to practical implementations.

The mentioned difficulties are caused by the absence of a proper platform with a simplified programming interface that is capable of building (compiling) in a short amount of time. In the Linux case, the complication results mostly from a required compatibility with various standards, e.g., Linux is compatible with posix and sysV standards [4]. In order to provide the compatibility with each of these standards, separate interfaces have been introduced. Consequently, it takes a lot of time to get familiar with the whole programming interface and, finally, students getting prepared to their laboratories are generally focused on studying the documentation instead of understanding the essence of presented mechanisms of the operating systems. Additionally, the build time of the embedded Linux requires about one hour, while laboratory classes usually last 90 minutes (at Polish technical universities).

In view of the above-mentioned difficulties the authors felt encouraged to develop a new didactic platform, including hardware, firmware and software tools. In the proposed platform, the handling of mechanisms such as files, multitasking, interprocess communication and process synchronization, have been simplified. The platform was initially presented at AICT 2011 [1]. Due to page limitations, the conference paper includes only the most important assumptions and a general description of the operation of the proposed platform. This article extends [1], presenting below a detailed description of all component elements of the platform. We also propose an extensive set of new laboratory exercises that make it possible for students to carry on with practical exercises in embedded real time operating systems unaided and on their own. In particular, we draw the reader's attention to the fact that the proposed platform can be used in controlling the intelligent home (smart home, eHouse) [5][6]. The firmware for the platform can be modified while being implemented in laboratory exercises according to one's needs and wishes, which secures easy and fast expansion of its functionality.

The remainder of the article is organized as follows. Section II presents state of the art. Section III presents the hardware of the proposed platform. In Section IV, the software architecture is described, including the programming software (software development kit) used in the process. In Section V, exemplary exercises conducted with the help of the proposed platform are presented. Section VI concludes the article.

II. Related work

One of the first operating systems developed for educational purposes was Mach system [7][8]. A group of systems represented by the Mach system was developed in academic circles in the years 1985–1994. The solutions delivered have been further adopted to numerous commercial operating systems, such as NeXTSTEP or Mac OS X [9].

The rapid development of software, especially of operating systems, started actually when the idea of GNU open source appeared in 1985 [10][11]. The operating systems, developed in accordance with the GNU idea, such as Linux [12] or FreeBSD [13], came into general use and became so attractive that they have been competing with the commercial solutions since then. The open source (GNU) systems combine the advantages of both the systems developed for educational needs and the systems used commercially.

Unfortunately, from the standpoint of teaching, these systems have become more and more advanced, thus preventing their use in the classes and teaching materials on the basics of operating systems. Simultaneously, students interested in practical issues and applications were not motivated enough to learn typical, education-oriented systems. The operating systems (with an open source) of real-time, dedicated to support the embedded systems [1], were indicated as much easier and optimal type of operating systems, possible to apply into the teaching process. The chapter presents further an analysis on potential use of existing real-time operating systems, programming environments and libraries in the teaching process. The analysis is limited to the systems comprising a support for the embedded systems. The following criteria were taken into account during the overview of existing solutions: an open source written in C, a support for the AVR architecture (because of rich microcontroller equipment, a simple and functional set of instructions, a free set of tools including C language compiler and a common presence in the projects for beginner constructors), a liberal license granting system, a support for the controller handling the Ethernet interface, an ability to be embedded on any microcontroller. Particular attention was paid to the latter criterion. Once met, it helps to replace a program written in a single thread, where a complicated loop implementing many tasks is made, with a multi-threaded program, where each step is implemented with a separate thread. Such approach increases the readability of the program (in each loop only one step is performed) and facilitates the division of work.

The first operating system to be considered in the article is Ethernat Nut/OS. The project of Nut/OS operating system is free of charge, open, BSD-licensed. It is a real-time operating system with a stack of TCP/IP network protocols, which supports the AVR, ARM7, ARM9 microcontrollers [14]. There are many projects of evaluation boards for this system. It should be noted, however, that these are complex devices that support, e.g., the embedding of Linux system. There is no option to embed the Nut/OS system on the simplest microcontrollers. The system features are: sustainable use of resources, support for multi-threading, dynamic memory management, but above all an implementation of TCP/IPv4 stack. The code is written in C. The software tools are prepared for Linux, Windows and MacOS. The system requires 32 kB memory – in case of the AVR microcontroller, an external bus is then required in order to add an external memory. The memory bus is used also to work with different types of peripherals, such as Ethernet controller. Particular emphasis in the project was placed on the optimization of supporting the peripherals, with the use of memory bus. This approach caused the appearance of difficulties with the servicing of devices applying the serial bus.

Arduino [15] can also be considered as an interesting didactic platform. It was designed to build, develop software for and be applied in simple devices. It includes both hardware elements and software tools. The hardware elements were made using the AVR microcontroller (without an external memory bus), and the processors of AtMega88 family (AtMega168, AtMega328), which differ in program memory capacity. There were numerous modules containing peripheral devices, i.a., Ethernet ENC28J60 controller and SD card reader, developed for the Arduino platform. The programming language is Arduino Programming Language [16] based on Wiring [17]. This language has the same syntax as C and contains a series of macros and functions for hardware abstraction. The applied abstraction helps with hiding some configuration details, e.g., the registers controlling entry/exit ports. Unfortunately, this platform is not a good element facilitating the teaching of operational systems basics, because a major part of its functionality, typical for operating systems, is omitted therein, such as multitasking and mechanisms of inter-process communication.

One of the most interesting current real-time operating systems, from the standpoint of the teaching of the basics of operating systems and embedded systems, is FreeRTOS [18] system. It was written in C and is licensed in accordance with a modified GNU GPL [19] license. It supports many microprocessor families (27 processors architectures). This system has been designed for minimal requirements. Its kernel takes 4 kB of program memory and it needs 0.5 kB of data memory to be embedded. With such small requirements the system may be applied to almost any microcontroller. FreeRTOS system supports pre-emption and multi-threading; additionally, the co-routines have been implemented therein. One of the advantages, deciding its selection for the purposes of the teaching platform discussed in the article, are good documentation and a large group of users and developers, which provides a good, long-term support for the platform. The free version is devoid of libraries designed to handle FAT 32 file system. Universal libraries working on all platforms to support network interfaces are not added either. Simultaneously, getting familiar with the FreeRTOS system enables the learners to use also the commercial versions of this system. An interesting extension of the FreeRTOS system is, e.g., OpenRTOS. OpenRTOS is FreeRTOS, provided under a commercial license that makes no reference to the GPL and includes fully featured professional grade USB, file system and TCP/IP components [18].

III. Hardware

Figure 1 shows a schematic diagram of the didactic platform described in the article, whereas Figure 2 shows a photo of the platform. The system is distributed and consists of the main controller, optional input/output (I/O) module and executing modules. The main controller and the executing module are programmed using a universal programmer designed and custom-developed for the platform's purposes. The solid line rectangles belong to the platform's hardware. The solid lines indicate communication interfaces or buses and the dotted lines indicate the programmer interfaces. The main controller is connected with the executing modules by an RS 485 bus. Additionally, the input/output module (I/O module) is connected to the main module by the SPI bus. The programmer module also has RS 485 interface in order to facilitate debugging or controlling the executing module if the main controller is disabled.

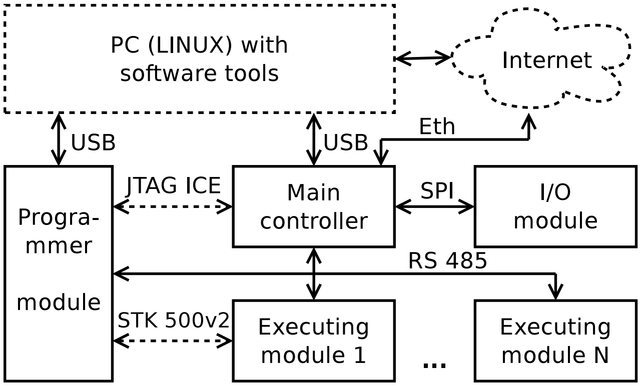


Figure 1. Modular schematic of the platform's hardware

The hardware part was designed with the help of a freeware version of Eagle [20] CAD software. The dimensions of the PCB board were limited to 10 by 8 centimeters (i.e., the maximum dimensions of PCB board allowed by freeware version of Eagle CAD software). The complete project of the hardware is available at svn repository http://rtosOnAvr.yum.pl/hardware/ssw [21], where the login and the password is "student". In order to download the project, the following command must be executed in the shell prompt: `svn co http://akme.yum.pl/eagle/ssw`. The limited dimensions of the board allow students to modify the project using freeware version of Eagle CAD.

The hardware was designed in a user friendly manner: it uses a common interface and does not need any external power supply. The platform is connected to a PC via USB, since RS 232 is not very common in modern personal computers. There is a place on the main controller for a power converter. It allows the platform to work as a stand-alone device that does not require power supply from the USB port. The hardware project is based on AVR microcontrollers [22][23]. This reduced instruction set computing CPU architecture is preferred by students because of its simplicity, freeware C compiler (avrgcc) and high performance in comparison with other 8-bit microcontroller architectures.

A. Main controller

The main controller is responsible for controlling the executing modules connected to the RS 485 bus and the I/O module, storing logs in its memory, and communicating with users via a USB or Ethernet interface. The modular schematic of the main controller is presented in Figure 3. The functional modules are presented as solid line rectangles and connectors or jacks are presented as dotted line rectangles. The main controller consists of: microcontroller Atmega128, 64 kB of data external memory, USB interface (Ft232Rl chip [24]), RS 485 interface (Max481 chip [25]), Ethernet interface (Enc28j60 chip [26]) and Secure Digital card reader. In order to communicate with external devices, sensors and modules, the controller uses the following buses: SPI, I2C and RS 485. All buses have their own connectors. A diagram of PCB board of the main controller that includes its most important elements and connectors is shown in Figure 4.

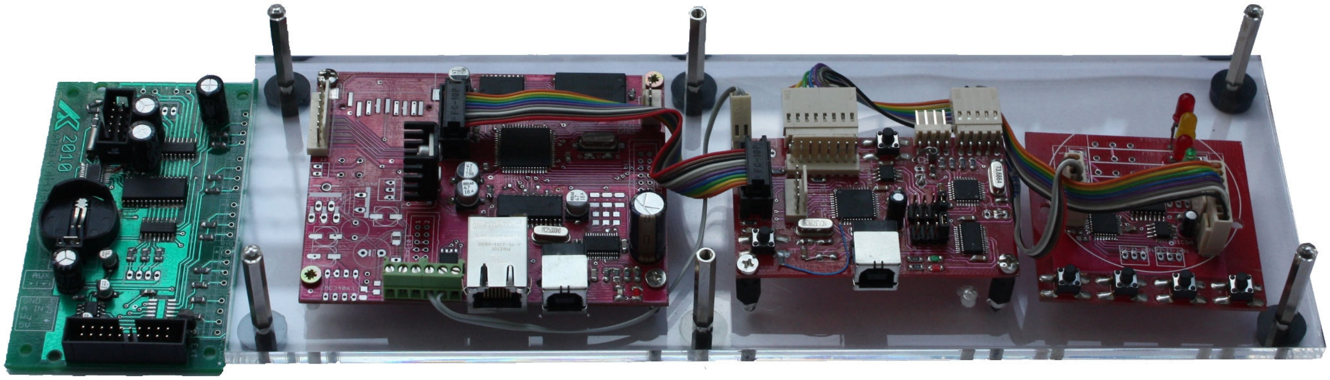


Figure 2. Didactic embedded platform: I/O module, main controller, programmer execution module.

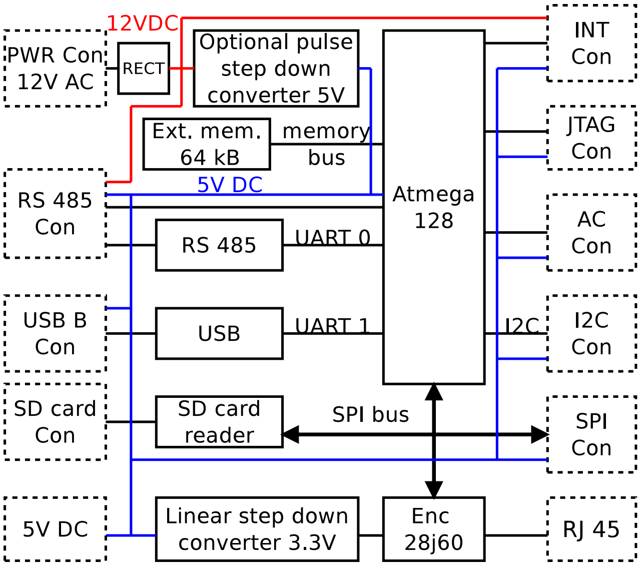


Figure 3. Ideological schematic of the main controller

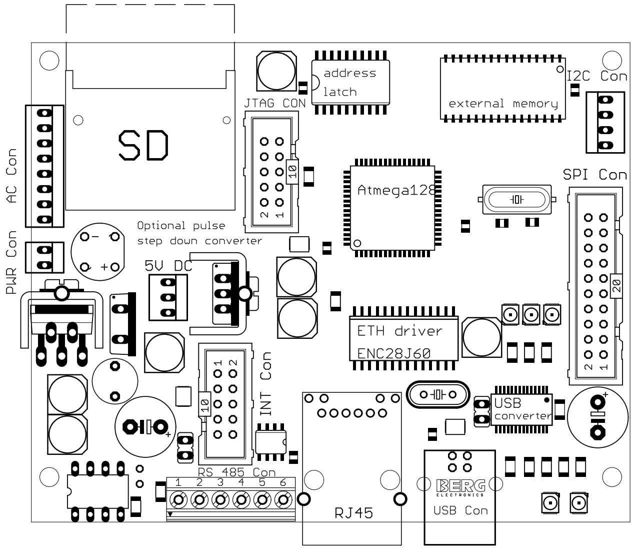


Figure 4. Main controller PCB with connectors

The microcontroller uses the SPI bus to communicate with the Ethernet controller and the SD card reader. It is also possible to connect 8 additional devices to this bus through an SPI connector placed on the main controller. The SPI connector can also be used in hooking up the input/output (I/O) module, as it is presented in Figure 1. Only memory is connected to the memory bus. The bus has no connector led out of its housing, i.e., no additional system can be connected to it.

The controller also has the optional 5 V pulse step down converter and a rectifying bridge. These elements of the controller can be useful if we want to use an external power supply because both the processor and other systems of the platform are powered by a 5 V voltage source. This power supply can be supplied either by a pulse converter or an USB port. In Figure 4, the external 12 V power supply line that leads from the rectifying bridge is denoted with the colour red and the 5 V power supply with the colour blue. The main controller provides power supply to all modules that are attached to it and, hence, each connector (or jack), to which any module can be hooked up, has its connection for power supply of its own.

As it was mentioned earlier, when a 5 V pulse converter is not available, the system can be powered from an USB port. With the application of this type of power supply, however, the A/D converter in the input/output module does not operate properly. Its analogue part is powered by a 5 V voltage source from it own linear converter that, at its output, requires at least 8 V. When it is powered by 5 V voltage (from an USB port), it will give output voltage lower than 5 V, and thus the A/D converter will be operating improperly. For didactic needs, it is possible to change the characteristics of the Input/Output module and make the analogue part of the converter powered by the 5 V voltage from the USB port. The execution module requires the power supply of 12 V to switch its own relays. Instead of 12 V, it is possible to supply 5 V (from the USB port) and introduce relays that operate under 5 V. Despite certain inconveniences, the system powered from the USB port is fully operational and functional in didactic situations. Lack of external power supply makes it easier to hook up and work with the set during lab classes (it is sufficient to supply power to the device from an USB port of the computer).

In order to reduce costs, the user communicates with the controller via console (VTY100 protocol). Access to the console is available both via the USB interface and the Ethernet interface. The main controller has neither display nor keyboard. The CPU is programmed using the JTAG interface that allows the user to debug the software. Additionally, there is a connector (*AD Con*) with analogue inputs and a connector (*Int Con*) with inputs generating interrupts. The connector *RS 485* provides proper access to earth ground, and the voltage 5 V and 12 V that provides power supply to the executing modules. Additionally, an input to the microcontroller has been introduced that can generate interrupts. This allows for a modification of the protocol operating on the RS485 bus so that the devices connected to the bus could impose service demands. This, in turn, makes it possible to eliminate the necessity for continuous checking of executing modules. The main module has a limited number of lines that can operate as input/output. The number of inputs/outputs can be, alternatively, made higher as a result of the application of an I/O (input/output) module that is connected to the SPI connector.

B. I/O module

The schematic diagram for the I/O module proposed in the article is presented in Figure 5, while its printed circuit board (PCB), along with a description of its most important components and connectors, is shown in Figure 6. The input/output module is composed of a port expander, an 8-input A/D converter and a real-time clock (RTC). Individual elements of the I/O module are presented in Figure 5 as solid line rectangles. Connectors are presented as dotted line rectangles.

The inputs of the I/O module are led out in such a way as to make them capable of being used during laboratory classes, those considered in the article, and for solutions of the type "smart home" (intelligent house). The module is connected to the main controller by the connectors *SPI Con* and *Int Con*. Connector *SPI Con* addresses and communicates with individual elements of the I/O module, whereas connector *Int Con* provides 12 V power supply and receives interrupts from the I/O module.

In the I/O module, the port expander is implemented with a MPC23S17 chip [27], which is connected to the SPI bus and the address line 7. The address line determines whether the system can use the SPI bus. This system has two 8-bit ports. Each line of the port can operate as input or output. In the I/O module seven lines from each of the ports operate as output, while the last one is not used. Each port of the expander is connected to a separate line

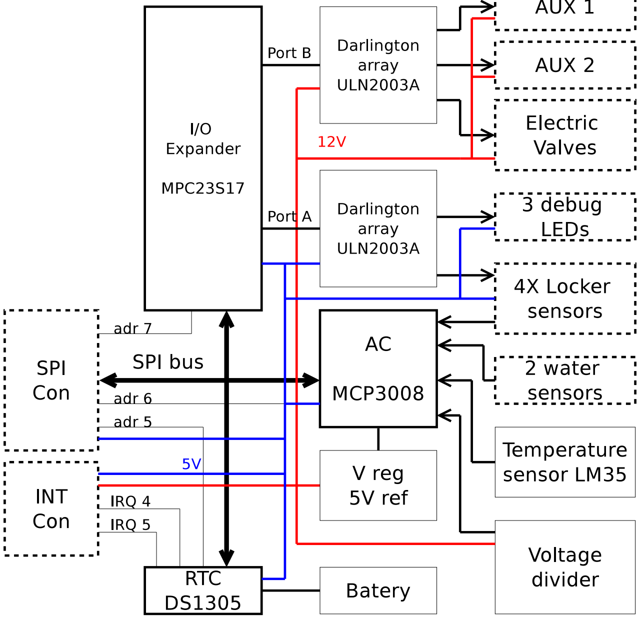


Figure 5. Ideological schematic (schematic diagram) of the I/O module

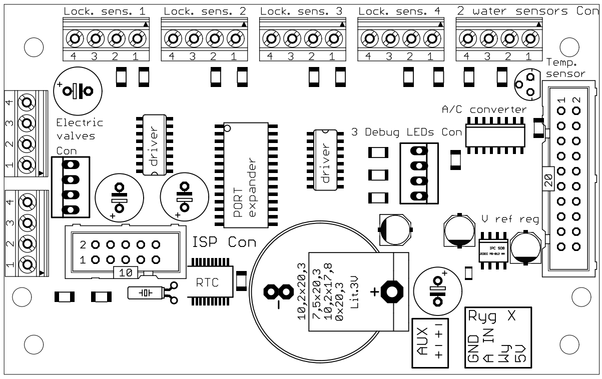


Figure 6. I/O module's PCB with connectors

of the (high-voltage) high-current controller implemented with the use of the high-voltage high-current Darlington transistor array (ULN2003A chip [28]). This system enables controlling devices whose consumed power exceeds 10 mA (current running though a single output of the expander cannot exceed 10 mA). The ports of the expander control different voltage. Port A controls devices that are powered by 5 V voltage, whereas port B controls devices that are powered by 12 V voltage. Despite the high current that the expander can control – after the application of transistors – one should not forget about the limited power of the power-supply unit and the limited current voltage that can run through the rectifying bridge in the main controller. In the case of an excessive load, the voltage on the power-supply line 12 V may drop.

For the convenience of the the didactic platform considered in the article, in the I/O module, some outputs are assigned to dedicated applications. Four outputs that control 5 V devices are designed to operate (flash) the diodes in lock bolt (valve) sensors and each of them is led out with a separate connector *Lock sensor*. The remaining three lines are led out with the application of the connector *3 debug LEDs*. Three outputs from port B, that control devices powered by 12 V voltage, are dedicated to control electro-valves and are led out through the connector *Electrics Valves*. The remaining four lines are led out with the application of two connectors: *AUX1* and *AUX2*. They can be used to control additional devices.

The A/D converter (MCP3008 chip [29]) located in the I/O module is connected to the SPI bus and to the address line 6 that allows it to occupy the SPI bus. The analogue part of the converter is powered from the output of the linear converter. In order to work properly, this converter requires an external 12 V power supply. In the case of a construction of a system that is to be powered from/by a USB port (without external power supply), the analogue part of the converter has to be connected directly to 5 V voltage (from the USB port). In the present didactic platform the converter is designed to check the state of valves (lock bolts) and makes it possible to verify whether the flat/house has been flooded. Checking the state of the valves (lock bolts) is based on measurement taking of the decrease in voltage (voltage drop) on the photo-transistor, after its LED diode is illuminated. If the doors are locked, then between the transistor and the diode there is a valve (lock bolt) that blocks admission of light to the transistor. Following this, the transistor does not transmit current and a voltage drop ensues. The flooding control is based, in turn, on a measurement of the voltage drop on the "flooding" sensor. If the sensor is dry, then it does not transmit current and a drop in voltage follows. In the case of flooding, the sensor transmits current and the voltage drop in it decreases. The A/D converter is supplemented with an analogue temperature sensor LM35 [30] and an output from the voltage divider (potential divider) of 12 V power supply. Thanks to the measurement takings in the voltage obtained from the divider, it is possible to determine whether the 12 V power supply line is overloaded or not.

The last system used in the I/O module is a real-time clock. This system is addressed through the address line 5. The real-time clock is placed in the I/O module instead of within the main controller due to the lack of space on the board of the main controller (no available place results from the limitations imposed by the free version of the eagle program). The applied DS1305 chip [31] is capable of generating interrupts. These interrupts are activated along with the alarm activation. Hour and date can be set in the RTC system. The system should remain operational even if power supply is not available, thus it is necessary to connect it to an additional battery cell that keeps its memory running and provide power to the clock system. In order to make use of alarms generated by the system, it has to be connected with the connector *INT Con* to the inputs of the microcontroller that generates IRQ 4 and IRQ 5 interrupts.

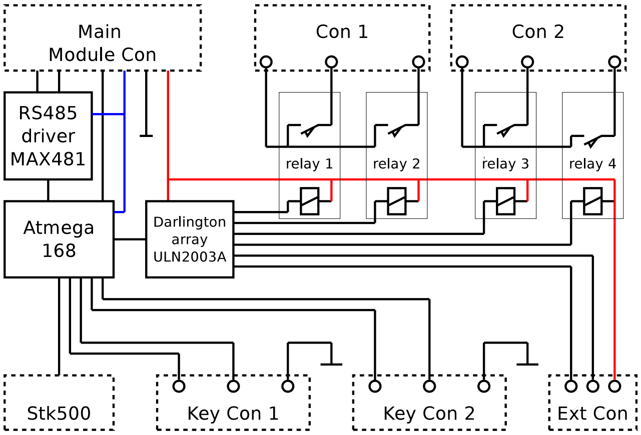


Figure 7. Ideological schematic of the executive module

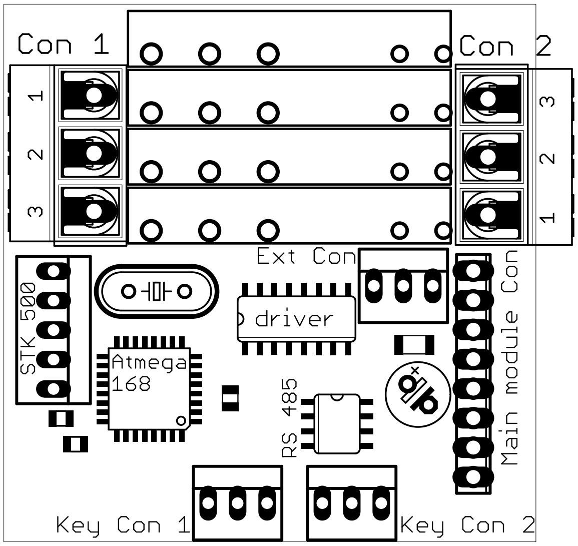


Figure 8. Execution module's PCB with connectors

C. Executing module

The executing module is responsible for switching on/off various devices, e.g., lights or roller shutters in an intelligent home. Figure 7 shows a schematic diagram of the executing module, whereas Figure 8 presents the arrangement (deployment) of its most important elements and connectors on the PCB board.

The executing module consists of: microcontroller Atmega168, RS 485 interface MAX481 [25], Darlington array ULN2003A and four relays. The module is equipped with a number of connectors that are shown in Figure 7 as dotted line rectangles. Connector *Main Module Con* is designed to connect the executing module to a common bus that serves the remaining modules and the main controller. The connector provides power supply and the connection to the RS 485 bus. Additionally, it has an additional line led out that can be used for different purposes in the future, e.g., to reset executing modules or to send interrupts to the main module. The module has two connectors (*Con 1* and *Con 2*) with relay joints. Two receivers (e.g., sources of light and their power supply) can be connected to each of these connectors. Additionally, the module is supplemented with the connector *Ext Con*, to which additional diodes or other low-power receivers can be connected.

The Atmega 168 microcontroller is programmed via SPI (connector *Spi Prog*). This means that there is no possibility to withhold from an execution of the program and to view the state of registers of the microcontroller. What may be useful then is to connect additional diodes to connector *Ext Con*. These will be instrumental in determining the current state of the device.

The executing module operates four buttons that control the devices. The outputs envisaged to accommodate the buttons are led out on the connectors *Key Con1* and *Key Con 2*. The executing module works as a slave device on the RS 485 bus. Its address can be set up with the help of five jumpers. Two of them are arranged at the stage of the preparation of the PCB board (by welding in resistors 0 Ω), whereas the remaining three jumpers are led out on the connector of the STK500 programmer.

The relays have independent power supply in order to avoid brownouts. Its voltage depends on the relays used (in the case of external 5 V power supply it is recommended to use special relays). Voltage is also supplied to connector *Ext Con* that can control low-power receivers. These receivers are powered with the same voltage as the relays. The executing module is realized with relays purposely, since this arrangement simplifies the preparation of the driver for didactic purposes. Working of the relay is audible and any device, including those that operate in low voltage, can be connected to it. In place of a relay, a LED diode can be appropriately connected. The LED diode provides information about the state of the output. The set presented in Figure 2 has been prepared in this way. In place of supplying power to the relay coil, a LED diode has been welded in and connected serially with a 330 Ω resistor.

Additionally, the repository [21] includes projects of other executing modules that have been realized using triacs with zero crossing circuit. An application of such a module exclusively for didactic purposes, however, would be impractical as this solution would require connecting devices that are powered by 230 V voltage and any work with the module would require special safety precautions to avoid electric shocks. The application of the executing module with triacs enables students to apply learned skills in practice because they are in a position to apply the presented set to control devices in an e-House. For this particular purpose, the module has been designed in such a way as to be easily accommodated in a standard recessed box (flush-mounted box). Its inputs have been additionally secured by adding zener diodes, while triacs have been secured by adding an appropriate RC circuit. Power supply to the external devices is secured with a varistor in such a way as to avoid any damage to triacs in case of power (voltage) surge in the mains. One should not forget, however, that there are certain differences in a module that is supported by relays and the one supported by triacs (the latter lacks appropriate place for the connector *Ext Con*), which necessitates a slight modification to the software.

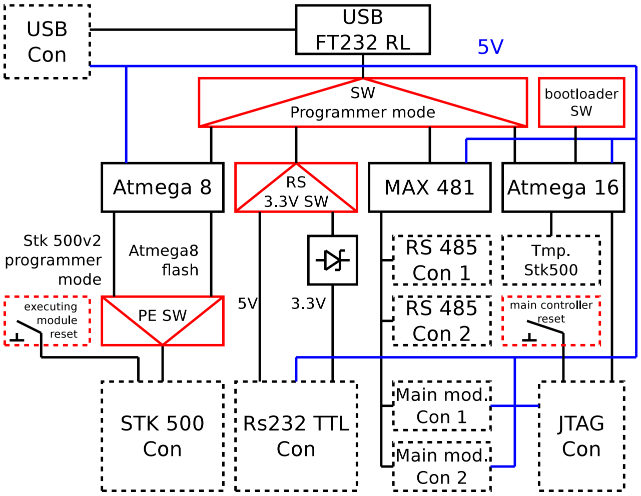


Figure 9. Ideological schematic of the programmer

The executing module can be programmed using the SPI bus (STK 500v2 programmer) or RS 485 bus (bootloader with xModem protocol). In the case of improper operation of the device (deadlock), unlocking the bootloader mode can be impractical. To make programming with the help of a bootloader possible in such a situation, it is sufficient to connect an additional line on the connector *Main Module Con* to the input reset of the microcontroller.

D. Programmer module

The programmer module has been designed to provide extensive functionality with a simultaneous reduction of costs. The programmer module uses the USB interface and, therefore, it does not require additional power supply. Its main function is flashing firmware to the main controller or executing modules. Both devices (main controller and executing module) have different programming interfaces (JTAG and SPI). The constructed programmer provides additional RS 485 and RS 232 TTL interfaces. The JTAG programmer bases on Atmega16 microcontroller and Atmel JTAG ICE firmware, therefore it is compatible with AVR

Studio. The archetype of the SPI programmer is an open source project [32] and it bases on Atmega8 microcontroller. The hardware has been slightly modified but the firmware has remained unchanged. The SPI programmer uses STK 500v2 protocol and is compatible with AVR Studio.

Figure 9 shows a schematic diagram of the programmer. Individual modules of the programmer are presented as solid line rectangles, whereas the connectors as dotted line rectangles. Additionally, the control switches are shown as rectangles marked with solid red line, while astable switches as rectangles marked with dotted red line.

The mode of operation of the programmer is selected (set) by two control switches. Each of them has to be set (positioned) in the same position for a given mode of operation. These control switches perform the function of a multiplexer. Connector *STK500 Con* is a connector of the STK 500v2 programmer. In addition, it can be used to embed (install) the firmware into the target Atmega8 processor so that the latter could operate as a programmer. For this purpose, the control switch *PE Sw* should be set into (Program Enable SW).

Additionally, the programmer system employs a control switch that allows the voltage of the transmitter of the RS232 port to be reduced to 3.3 V on the output of the connector *RS232 TTL Con*. This option is very useful when the programmer module is used to connect itself to the router console, e.g., Edimax 6104KP. The work on the console and the modifications to the firmware for this device is not, however, the subject for the present article and thus will be omitted.

Leads of the serial port can be used to establish connection with other devices that have a serial port led out with the TTL voltage level or 3.3 V voltage level. The RS485 communication bus is led out on four connectors. This allows for a number of sets to be connected serially so that, ultimately, a network of distributed devices can be built up. While implementing such a solution one has to remember, however, that the system thus executed has only one main controller working or, alternatively, a change of the protocol has to be implemented (e.g., apply a protocol of the type Token Ring, in which main controllers will be passing on tokens to one another that allow them to work on the bus in the master mode and checking the availability of the remaining modules or main controllers operating at the time in the slave mode). Two connectors (*RS 485 Con 1* and *RS 485 Con 2*) have only the RS 485 bus led out. The remaining connectors (*Main controller 1* and *Main controller 2*) have additionally 5 V voltage led out. By selecting the RS485 operating mode, we can monitor and communicate with the executing modules through the RS 485 bus. The solution that has been applied to the didactic platform makes it possible to control the executing modules directly from the computer. The only conditioning element is then to switch off the option of transmission on the bus by the main controller, or an application of some other protocol (e.g., a protocol of the type Token Ring).

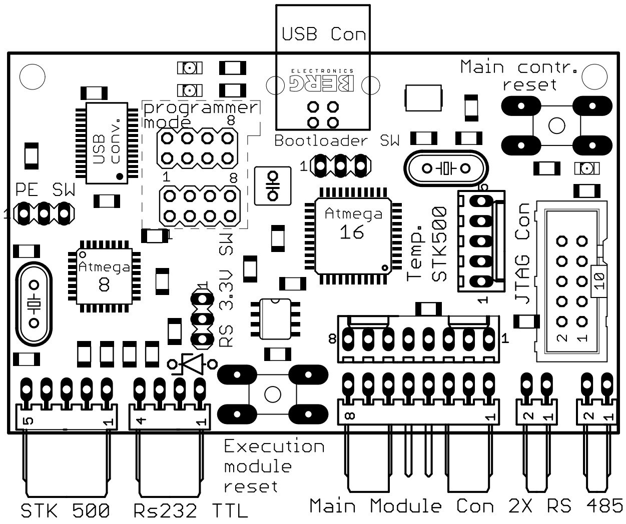


Figure 10. Programmer's PCB with connectors

The main module can be programmed by a JTAG programmer. It is connected to the connector *JTAG Con*. The JTAG programmer is activated by the connector *Temp. STK500*. The bootloader is installed through the connector. The control switch *Bootloader SW* [33] is used for the activation of the bootloader. With the bootloader, it is possible to download the newest version of the firmware of the JTAG programmer from the Internet. When this is the case, installation of the AvrStudio studio is then required.

The executing module and the main controller can be restarted with the buttons *executing module reset* and *main controller reset* that are placed in the programmer.

IV. Firmware

The firmware was written in C language. The complete source code is available at svn repository http://rtosOnAvr.yum.pl/software/FreeRtos [34], where the login and the password is "student". The firmware part of the presented didactic platform consists of two basic parts: the firmware for the main controller and the firmware for the executing modules. Each device has a different microcontroller and has different functions, therefore it needs specialized firmware. There is an embedded RTOS on both modules. The authors have chosen FreeRTOS as the RTOS because it is distributed under a modified GPLv2 license [19]. FreeRTOS uses two methods of providing multitasking: tasks and coroutines. Its kernel needs 4 kB of program memory, hence it is possible to use FreeRTOS on microcontrollers with 8 kB of program memory. Originally, FreeRTOS was ported to the Atmega32. In the case of the proposed platform, it has been necessary to make a port for Atmega168 and Atmega128 microcontrollers.

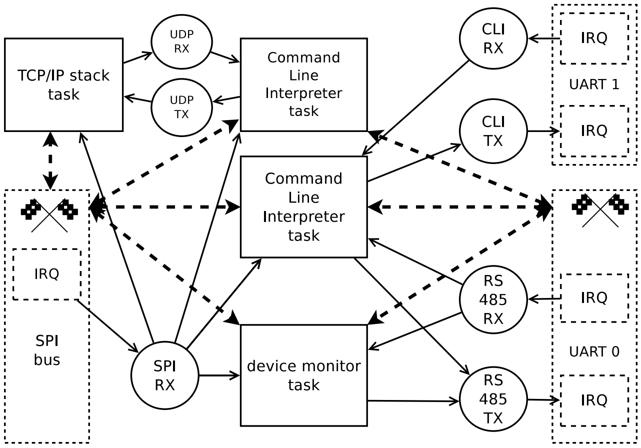


Figure 11. Architecture of main controller firmware

A. Main controller

The main controller is responsible for controlling the I/O module, executing modules and communication with users. It stores logs and allows the scheduling of some operation, e.g., moving up the roller shutters. The main modules of the main controller firmware are the following: kernel, command line interpreter, file system, communication protocol, TCP/IP stack and xModem protocol.

1) Kernel: Multitasking in the main controller is provided with the help of tasks without preemption. Such an approach has numerous and significant advantages. Tasks are simple, have no restrictions on use and support full preemption (not used in case of labs excercises). Moreover, they are fully prioritized [35]. The firmware has been written without preemption, so re-entry to the task does not need to be carefully considered. The main disadvantage is that each task has its own stack. The Atmega128 has 128 kB of program memory and 4 kB of internal data memory, extended by external chip to 64 kB, and allows us to use FreeRTOS with tasks. It is recommended to place stacks of the tasks in internal memory, hence there are 4 kB available for stacks. There are four tasks: two Command Line Interpreter tasks, a device monitor task and a TCP/IP stack task. 4 kB is enough for four stacks. In order to save internal memory, buffers and other structures have been moved to two times slower external memory. Constant strings and constant structs are stored in flash (program) memory. In Figure 11 the firmware architecture of the main controller is presented. It bases on the mentioned four tasks.

The system supports two simultaneous console sessions. Each session is serviced by separate Command Line Interpreter task. The first task (at the top of Figure 11) is responsible for the communication with the user according to the TCP/IP protocol stack. It reads out the sequence of characters (signs) given by the user from the UDP RX buffer and transmits the reply to the UDP TX buffer. The second task that services the command interpreter operates in a similar way. This task receives data from the UART1 serial port through the CLI RX buffer and transmits data using the CLI TX buffer. This task uses serial port UART 1 for its exclusive use. This simplifies the implementation since the introduction of synchronization is not necessary. In addition, the CLI tasks make use of co-shared resources such as the SPI bus and the UART0 serial port. Since only one task can use the co-shared resources at a given time, it is thus necessary to introduce certain synchronization that enables exclusive access to be implemented. Synchronization can be effected with the help of the mechanisms made available by the FreeRTOS system, such as, e.g., semaphores.

The semaphore blocks simultaneous access to one of the resources by more than one task. In Figure 11 the semaphores are marked by a racing checkered flag symbol. When the task is attempting to enter the critical section (e.g., read or write to serial port UART 0), it has to pass through the semaphore. If the semaphore is locked, the task is suspended as long as the semaphore is locked. Once the semaphore is unlocked, the task is released automatically and the semaphore is locked again by this task. The task unlocks the semaphore again after leaving the critical section. FreeRTOS provides a special API for handling semaphores. The task is suspended as long as the semaphore is locked, or until its optionally specified timeout.

FreeRTOS supports an API for buffer handling in order to simplify the implementation of the main controller firmware. There is a special function for writing to the buffer. If the buffer is full, the task is suspended as long as the buffer is full and optional specified timeout is not exceeded. The function informs (returns the result) if the operation was successful or not. Similarly there is a function for reading the buffer. If the buffer is empty, the task is suspended. The task is released when data is available in the buffer or timeout is exceeded. All the mentioned FreeRTOS API functions are non-blocking functions. If the task is suspended, the microcontroller is executing other, not suspended, tasks. The developer has to care about avoiding deadlocks. Programming tasks is thus complementary to the operating systems theory within the range of topics related to deadlocks.

The task of the device monitor is to check the state of modules connected to the RS 485 bus or the SPI bus. This includes polling all devices connected to the RS 485 bus, reading analogue inputs values and communicating with devices connected to the SPI bus (e.g., RTC clock). The task uses the resources such as SPI BUS or serial port UART 0. The task is synchronized with other tasks by semaphores.

The TCP/IP stack task is responsible for listening and establishing new connections and handling them. Currently work is being carried out on a full implementation of the TCP protocol. Remote access to the console is effected through the UDP protocol. The task uses the SPI bus and is also synchronized. This tasks has a lower priority than the two other tasks.

The proposed didactic platform does not provide support for preemption. Excluding the preemption allows to error notification – the errors would stay unperceived if the preemption was used. The students deal with the preemption and race condition at high-level programming language courses.

2) Command Line Interpreter: The main controller provides interactive communication with a user via a Command Line Interpreter. Initially, the CLI was taken from the AVRlib project [36]. The original CLI was not designed for a multitask environment: only one instance of the CLI was available and, furthermore, it was working on global variables. The original CLI was not ready to cooperate with `stdio` C library. As a result, for the purpose of the proposed platform, most of the code of the original CLI has been rewritten. Now, it is possible to use many independent instances of CLI. Each CLI has the history of its last four commands and works on a new engine. The proposed CLI is compatible with the `stdio` library and it is possible to use `fprintf` functions in order to make a print.

The new CLI API is user-friendly (it allows users to add new commands easily) and communication with the main controller is simple. The command `help` displays all available commands and its description. In the next section, the method for adding new commands to the interpreter will be discussed.

3) File system: An important part of operating system theory is devoted to file systems. For the purpose of the didactic platform, a simple file system, the so-called FAT 8, has been written. It can address up to 256 clusters. Each cluster, contrary to the CP/M operating system, has 256 bytes instead of 128, which has simplified the file system implementation. The whole implementation takes about 500 lines of code and is compatible with the avr-libc [37] API. The file is visible as a stream. Writing to a file is possible using the `fprintf` function.

4) Communication protocol: The main controller and the executing modules are connected to a common medium – the RS 485 bus. The communication model looks as follows. The main controller (master) starts the transmission on the bus. Each frame sent by the master main controller has an address of a slave device (an executing module) – the receiver of the message. The slave device can answer to the message. The frame format is Type Length Value. The frame fields are the following: synchronization sequence, address, type of message, message length and message data. Two bytes with CRC sum end the frame.

5) TCP/IP stack: The TCP/IP stack implemented in the presented didactic platform is based on the stack proposed within *HTTP/TCP with the Atmega88 microcontroller (AVR web server)* [38] project. For the purpose of our project, the TCP/IP working on Atmega88 with 8 kB of program memory was adopted for multitasking system. The TCP/IP stack is supported in the presented didactic platform only partially. At the current stage, only the ICMP protocol and UDP socket are implemented. The next releases of the didactic platform will also include an implementation of IPv6, servicing several TCP connections and WWW server.

6) Xmodem protocol: This protocol allows to send or receive files. It cooperates with the `stdio` library and input/output stream. This protocol is useful for bootloader handling. It allows to flash the executing module by a new firmware image. Implementation of the TFTP protocol is much more complicated.

7) The operation of specific devices that are connected to the SPI bus: The I/O module does not include systems that need software downloading to operate. The systems of the I/O module require, however, appropriate control. Software the controls them is already built into the main controller as firmware.

The following libraries have been created for the specific needs of the platform: A/D converter (MCP3008 system), port expander (MPC23S17 system) and the real-time systems (DS1305). All these systems communicate with the microcontroller through the SPI bus. Each system is connected to a different address line. Addresses of individual systems can be set up in the configuration file of the project (design).

In order to simplify the communication process, a library to handle the SPI bus has also been prepared. This library introduces the possibility of checking the state of the semaphore before an attempt is made to occupy this communication bus by a given process. After entering the critical section, before communication commences, the operating mode for the SPI bus is configured to adjust its configurations to, individual to a given device, processing speed of sending/receiving data. The library responsible for servicing the SPI communication bus also provides two commands for concurrent writing and reading to/from the SPI bus (*spiSendSpinBlock* and *SpiSend*). The first is a blocking operation. The process performs busy waiting until termination of data sending on the SPI bus, whereas the other version of the command is a non-blocking operation. During data sending on the SPI bus the process is suspended (excluding instances of busy waiting), and, within the time offered, the operation system can perform another task. After termination of the sending operation, the task can be resumed. In the case of the work with systems that can communicate with great speed through the SPI communication bus, blocking operations should be applied because switching of a context occupies more time than the operation of sending one byte. If the device works at a low speed and the duration of sending data on the SPI bus takes more than switching of the context twice, then non-blocking operations should be used. After termination of the use of the communication bus, the process must release it.

The libraries responsible for the service of particular

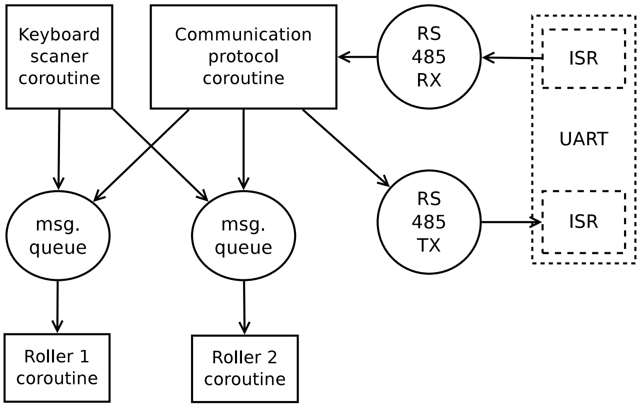


Figure 12. Architecture of executing module firmware

systems, e.g., the RTC clock, make use of the API for the service of the SPI communication bus. This simplifies the implementation of services rendered to other remaining devices. More details related to the control of the devices of the module are included in the following section.

B. Executing module

The executing module controls four relays and reads four inputs. It is suitable for controlling, e.g., two roller shutters or four light sources. Some controlling functions can be fulfilled automatically, e.g., after pressing the button the relay is switched on. The relay state may be changed after receiving special command from main controller.

1) Kernel: The executing module has no complex configuration. Its microcontroller has only 16 kB of program memory and 1 kB of data memory. In order to save data memory, the FreeRTOS is using coroutines. The coroutines share a common stack. The coroutines in FreeRTOS are automatically restored by the scheduler and a developer does not need to focus on them. Moreover, they are very portable across other architectures [35]. The disadvantage of the application of coroutines requires special consideration. The lack of stack causes data stored in local variables to be destroyed after the restoration of a coroutine, which complicates the use of coroutines. The coroutine API functions can be called only inside the main coroutine function. In FreeRTOS, the cooperative operation is only allowed among coroutines, not between coroutines and tasks. For this reason, there are only coroutines and no tasks in the firmware of the executing module.

Figure 12 shows the architecture of the executing module that controls two roller shutters. For driving a single roller shutter two relays are required as one executing module can coordinate two roller shutters. The firmware consists of four coroutines, presented in Figure 12 as solid line rectangles. Two coroutines drive the rollers, additionally there is a coroutine that scans the keyboard connected to the executing module and another one responsible for communication within the RS 485 bus. The coroutines communicate with each other by two buffers presented in Figure 12 as circles. The coroutine responsible for communication with the RS 485 bus can send appropriate commands to the driving roller shutter coroutine with the help of the buffer. The same buffer can be used by the scanning keyboard coroutine to send a message. The messages sent by the buffer includes information on relays (its number), which should be switched on or off at a specified time.

2) Communication protocol: Executing modules work as slave devices. The communication is always started by a master device by sending a message with a slave device's address (destination address). All slave devices check the destination address of the received messages. If the slave device's address matches the message's destination address, the slave device answers and executes the command issued by the main controller. In most cases, messages with not matching addresses are ignored. There is only one exception to this rule, which is presented in the next section.

The coroutine that services the communication protocol communicates with the RS485 bus via the buffers: RS485 RX and TX. These buffers are also used by interrupt handlers such as "Receive Complete" and "Data Register Empty". If a serial port receives a new sign (name), then the Receive Complete interrupt is initiated. In the implementation of the software for the executing module, the service for this interrupt involves placing this sign in the RS485 RX buffer. This is a buffer that is realized in a programmable way with the capacity of 16 bytes. Apart from program buffers, AVR microcontrollers are equipped with sending and receiving buffers for serial ports with the capacity of two bytes. If a sending buffer is available (at least 1 byte is free), then "Data Register Empty" interrupt appears. Handling of this interrupt involves checking the state of the RS 485 TX buffer. If certain data are in the buffer, then they are retrieved and stored in the sending buffer of the device. When they are missing, "Data Register Empty" interrupt handling is activated. This interrupt must be activated after new data are stored in the RS 485 TX buffer. The next section will include a description of the API of the FreeRTOS system designed to handle the buffer by coroutines and the functions handling interrupts.

The initial design for the didactic platform envisages a possibility of an expansion to the communication protocol for the communication bus has an additional line, with which devices of the slave type can generate interrupts. In addition, slave devices can send and read the information on the state of the bus concurrently. This makes them capable of detecting conflicts when a number of devices sends data along a common medium – the RS 485 bus.

3) Bootloader: The bootloader is mainly used when a STK 500v2 programmer is not available or when it is not connected. The main controller can flash firmware to the executing module. With the help of the xModem protocol the firmware image is first uploaded to the main controller and stored in a file. Next, the main controller sends a restart command to the executing module and if the address is matched, the device restarts. Otherwise, the device disconnects from the RS 485 bus for 60 seconds – this is enough to write firmware to the executing module. After restart of the executing module the bootloader code is executed. The bootloader waits 30 seconds for the flash command. After receiving it, the executing module is trying to download firmware using the xModem protocol. The main controller sends firmware according to the xModem protocol.

4) Keyboard scanner: There is a coroutine described in Section IV responsible for keyboard scanning. It can distinguish a key press from stick bouncing on keyboard.

C. Software tools

The prepared toolset for the platform purposes works on Linux and consists of an editor (Integrated Development Environment – IDE), compiler, repository and programmer software.

Software programs worked out for the purpose of the didactic platform are configured in such a way as to be handled by a freeware Kdevelop 4 editor. For this purpose, the file cmake.txt had to have been written separately and appropriately for each of the projects. It provides an instruction, based on which the Makefile file (in the case of the Linux system) will be generated. In addition, the file cmake.txt informs the editor about the names of files that are included in the project.

Figure 13 presents a screen shot of the editor. The screen shows a project called CLI. Files of this project have been grouped thematically based on the information within a cmake.txt file. Kdevelop, since its version four, stores information on the project in the cmake.txt file (earlier versions used GNU autotools [39], and information about project files was stored in the makefile.am file). Using the program cmake [40], the Makefile file is generated that provides instructions to the program make [41] concerning the method for a compilation of individual files and the way they should be compiled to create the image of the system (hex file) that would be ready to be installed into the microcontroller.

Some sample Makefile files are attached to the projects included in the repository [34]. They can be accessed and used directly without the cmake tool. To do this, it is sufficient to activate the `make` program to compile the project and the `make program` to transfer the image to the microcontroller. This solution is, however, rather inconvenient, especially when errors occur. When this is the case, it is necessary to access the information on errors and then open a given file and find the indicated code line to remedy or eliminate the error.

The addition of the `cmake.txt` file considerably improves code writing. The project can be compiled by pushing the button *Build Section* in in the Kdevelop editor that is framed in red in Figure 13. In the case of an error occurrence, the editor displays an appropriate message and a single click takes us to the erroneous code fragment.

In addition, the Kdevelop environment collects information on all data structures defined in the project and facilitates browsing and managing them. These structures are made accessible after a special bookmark is activated, which is shown in Figure 14.

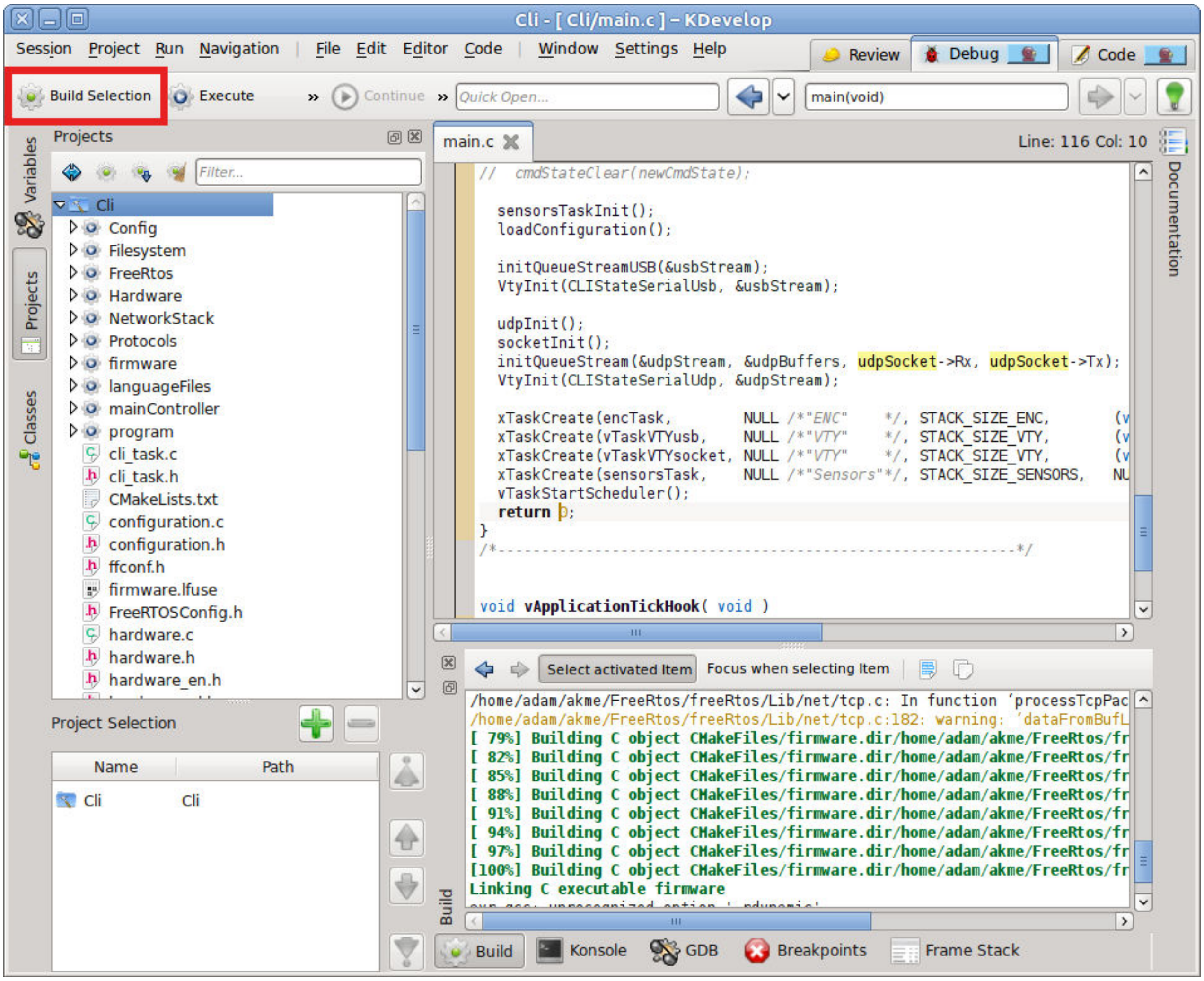
In the Ubuntu distribution, all of the required programs are available in its repositories and can be installed using the `apt-get install` command. Thanks to this advantage it is very easy to write instructions for students, how to prepare the system to be up and running.

V. Laboratory exercises

The presented platform allows users to prepare an extensive number of exercises, both in operating systems and in embedded systems. The laboratory exercises prepared for the platform can successfully replace exercises that are usually carried out with the help of the Linux system. The two sample laboratory classes presented further on in the section include basic issues related to the theory of operating systems, i.e., multitasking, synchronization of processes, inter-processor communication and the interpreter of commands. At the same time, the proposed platform can also be used as a didactic support to classes in network embedded systems. Students can both design a protocol of their own that would be operative on the RS485 bus, and can modify network protocols and (in the future) a www server.

For the purpose of the teaching process during classes, templates, provided in the repository available at http://rtosOnAvr.yum.pl/software/FreeRTOS [34] in the directory Lab, have been worked out. The project templates are provided in the `templateProjects` directory. The library functions described in the previous section, have been placed in the `freeRtos/Lib`. It is recommended that the contents of this directory should not be modified during laboratory classes. Each of the projects included in the platform has a *makefile* added. The file allows the user to quickly and easily construct a project: all that is needed is to simply type the command: `make` and `make program` to upload the constructed firmware image into the microcontroller. Additionally, the projects include files `cmake.txt` that enable integration of a project with the KDevelop editor.

In the remainder of this section, two sample laboratory exercises that employ the proposed platform are presented, i.e., the exercises "CLI Interface" and "Coroutines FreeRTOS API".



```
//   cmdStateClear(newCmdState);

  sensorsTaskInit();
  loadConfiguration();

  initQueueStreamUSB(&usbStream);
  VtyInit(CLIStateSerialUsb, &usbStream);

  udpInit();
  socketInit();
  initQueueStream(&udpStream, &udpBuffers, udpSocket->Rx, udpSocket->Tx);
  VtyInit(CLIStateSerialUdp, &udpStream);

  xTaskCreate(encTask,        NULL /*"ENC"    */, STACK_SIZE_ENC,        (v
  xTaskCreate(vTaskVTYusb,    NULL /*"VTY"    */, STACK_SIZE_VTY,        (v
  xTaskCreate(vTaskVTYsocket, NULL /*"VTY"    */, STACK_SIZE_VTY,        (v
  xTaskCreate(sensorsTask,    NULL /*"Sensors"*/, STACK_SIZE_SENSORS,    NU
  vTaskStartScheduler();
  return 0;
}
/*-----------------------------------------------------------*/


void vApplicationTickHook( void )
```

```
/home/adam/akme/FreeRtos/freeRtos/Lib/net/tcp.c: In function 'processTcpPac
/home/adam/akme/FreeRtos/freeRtos/Lib/net/tcp.c:182: warning: 'dataFromBufL
[ 79%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[ 82%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[ 85%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[ 88%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[ 91%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[ 94%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[ 97%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
[100%] Building C object CMakeFiles/firmware.dir/home/adam/akme/FreeRtos/fr
Linking C executable firmware
```

Figure 13. Kdevelop as IDE

A. CLI Interface

During the first laboratory classes students learn in more detail about the option of adding new commands. The approach adopted in the laboratory classes is similar to the approach adopted in programming teaching: during the classes a simple command will be created that, after prompting, will cause the “Hello World !!!” message to be displayed on the screen. The base project template, to which a new command is to be added, is in the directory `Labs/cli`.

The addition of a new command does not require any extensive knowledge of the code structure for the whole of the software for the main controller. In order to achieve the main goal of the exercise it is sufficient to perform the following operations: writing a function that will be executed after the appropriate command and defining the name of the command and complementing it with its description. Each command is written in the `command` structure.

```
struct command
{
  prog_char *commandStr;
  prog_char *commandHelpStr;
  CmdlineFuncPtrType commandFun;
};
```

The structure includes all elements that are necessary in the process of adding a new command. The expression type `prog_char *` defines the index for a string stored in the flash memory of the program. Such strings are handled in a different way than strings (`char *`) stored in data memory. The type `CmdlineFuncPtrType` is an index for the function that executes a command. Each such function accepts the index for the installation of the command interpreter as argument and returns the result that provides information whether the command has been properly executed or not. The declaration of the index for the function is presented below.

```
typedef cliExRes_t
  (*CmdlineFuncPtrType)(cmdState_t *state);
```

The result of the function is defined in the enumerated (enum) type cliRes_t.

```
enum cliExecuteResult
{
  OK_SILENT =0,
  OK_INFORM,
  SYNTAX_ERROR,
  ERROR_SILENT,
  ERROR_INFORM,
  ERROR_OPERATION_NOT_ALLOWED
};
```

```
typedef enum cliExecuteResult
cliExRes_t;
```

Type cliExecuteResult includes six feasible results. When the command is properly formed, the value OK_SILENT or OK_INFORM is returned. The latter value is used to inform overtly about the proper execution of the command. Additional values of the enumerated type make it possible to, e.g., provide information on the lack of required parameters or on parameters that have been given inaccurately, in the case of commands that require additional parameters to be furnished. When this is the case, the execution function will return the value SYNTAX_ERROR. With the instance of an error occurrence during the execution of a command, the command interpreter can inform overtly with the message (ERROR_INFORM) or, alternatively, it can leave out the information on the error (ERROR_SILENT). If the execution of a given command is not possible, then the last value of the enum type under consideration will be returned, i.e., the value ERROR_OPERATION_NOT_ALLOWED. In the case of the considered command that prompts the "Hello World" welcome message, information on a properly executed command may not necessarily appear on the screen, therefore the value OK_SILENT will be returned by the function as shown in Figure 14.

In the case of the proposed didactic platform, messages are written similar to the way they are written in the C language, i.e., with the help of the function fprintf_P. The sequence _P means that the text chain is stored in program memory and not in data memory. The index for the output stream is in the structure that stores information on the instance of the command interpreter cmdState_t. This structure will be discussed in more detail in the later part of this section. The command interpreter has been designed in such a way as to make it capable of handling many languages. Hence, all commands and their descriptions are written in separate files, e.g., vty_en.h for the English language, or vty_pl.h for the Polish language. At the stage of adding a command, it is recommended to add in each of these files an appropriate chain so that, after a change in the language, the project could be immediately compiled. Variables that define text chains are labelled according to the following convention: variables that include the name of the command will start with cmd_, whereas variables that include the name of the command along with a description of the command will start with cmd_help_. Thus, for the sample "hello" command under consideration:

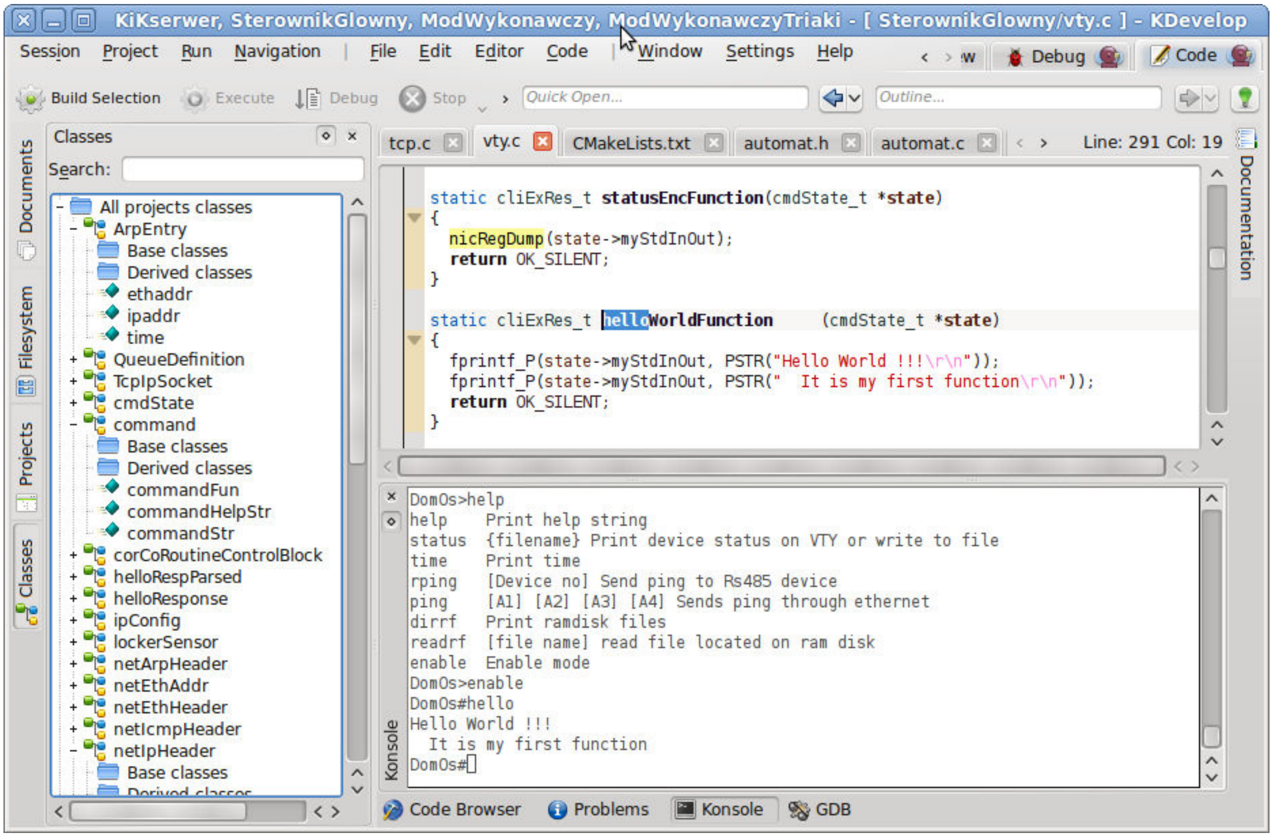
```
 prog_char cmd_hello[] = "hello";
 prog_char cmd_help_hello[]
   = "Writes␣hello";
```

The screen shot from the Kdevelop program that includes the function executed after the "hello" command has been enabled is presented in Figure 14. The name of the function, in line with the adopted convention, ends with a suffix Function. Note that the text is written onto the screen with the help of the function fprintf. This function adopts as the first argument the index for the output stream. At this point, the application of the PSTR macro as the second argument needs certain explanation here. This macro imposes an inclusion of stings in the memory of the program instead of, as it is adopted conventionally, in data memory.

The last element related to the addition of a command is to place the structure with the added command in an appropriate command table. In the example presented in Figure 14, the command has been placed in the menu at the privileged level. Therefore, the table cmdListEnable included in the vty.c file should be completed with yet another type command_t.

```
command_t __ATTR_PROGMEM__
  cmdListEnable[] =
{
  {cmd_help, cmd_help_help, helpFunction},
  ...
  {NULL, NULL, NULL}
};
```

The table cmdListEnable is placed in the memory of the program (attribute __ATTR_PROGMEM__). This mean that an increase in the number of commands will not lead to a decrease in the available data memory. A reduction in the available data memory could eventually lead to a situation where the system simply hangs. In addition, a special command status has been added to the system. The command returns information on available memory. If the obtained value has a negative number, then it is necessary to decrease one of the stacks of tasks being serviced. Otherwise, the stack or the buffer cache will overlap a section of the memory cache that is occupied by global variables, which, in consequence, will block the operation of the system or will render the operation of the system unstable.



```
static cliExRes_t statusEncFunction(cmdState_t *state)
{
  nicRegDump(state->myStdInOut);
  return OK_SILENT;
}

static cliExRes_t helloWorldFunction     (cmdState_t *state)
{
  fprintf_P(state->myStdInOut, PSTR("Hello World !!!\r\n"));
  fprintf_P(state->myStdInOut, PSTR("  It is my first function\r\n"));
  return OK_SILENT;
}
```

```
DomOs>help
help    Print help string
status  {filename} Print device status on VTY or write to file
time    Print time
rping   [Device no] Send ping to Rs485 device
ping    [A1] [A2] [A3] [A4] Sends ping through ethernet
dirrf   Print ramdisk files
readrf  [file name] read file located on ram disk
enable  Enable mode
DomOs>enable
DomOs#hello
Hello World !!!
  It is my first function
DomOs#
```

Figure 14. IDE and Hello world function

Further active participation in the set of the laboratory classes requires students to know the file system developed in the project. There are the following files in the project directory: main.c(h), cli_tasks.c(h), netstack_task.c(h), sensors_task.c(h), hardware.c(h), serial.c(h), vty.c(h), configuration.c(h). A device is initiated in the file "main", then tasks are created. Functions targeted by appropriate tasks are in the files: cli_tasks.c(h), netstack_task.c(h) and sensors_task.c(h). The functions of these tasks make use of the module libraries of appropriate modules. The knowledge of their implementation is not necessary to carry on with the exercises during the classes. Basic knowledge of the API is sufficient. The "hardware" file includes appropriate functions that handle the devices included in the evaluation set. The configure.c file includes functions that handle writing and reading of the configuration, e.g., the IP address and the mask. The serial.c file is responsible for handling serial ports. These ports send and receive data through buffers that are also used by tasks operating in the system. In the vty.c file, functions to be performed after an appropriate command has been written to the interpreter are defined.

The next proposed laboratory task related to the usage of the command interpreter is to add a command that controls the output of the MCP 23S17 expander connected to the microcontroller through the SPI bus. The controlling functions for the expander are in the Lib directory and students do not have to know its precise implementation. The implementation itself has been realized in such a way as to show some interesting aspects of the C language that are not necessarily discussed during lectures.

The library functions used during the lab classes have been prepared in such a way as to be employed in an all-purpose manner, i.e., the address of a device connected to the SPI bus has not been specified. This device (the port expander in the discussed case) is prompted with the help of the functions included in the hardware.c file. The file includes functions that are appropriately adapted to a specific set, where each of the devices involved is always connected to the same address line. These settings are written in the hardware.h file. This saves time for students as they do not have to learn the construction of the main controller in detail. Figure 3 shows that each of the modules connected to the SPI bus has a different address and, thus, its address line is connected to a separate output in a port of the microcontroller. The functions included in the hardware.c file automatically control the available ports of the microcontroller and ensure that no more than one device is addressed at a time. Some devices are addressed by high state, others by low state. Using appropriate macros, the user does not have to know all these implementation details, which allows at the same time to maintain high efficiency in controlling the modules. A sample way of addressing the MPC23S17 port expander is presented in the following code.

```
void enableSpiMPC23S17(void)
{
#if MCP23S17_SPI_CS_EN_MASK_OR != 0
  MCP23S17_SPI_CS_PORT |=
    MCP23S17_SPI_CS_EN_MASK_OR;
#endif
#if MCP23S17_SPI_CS_EN_MASK_AND != 0xFF
  MPC23S17_SPI_CS_PORT &=
    MPC23S17_SPI_CS_EN_MASK_AND;
#endif
}
```

In a similar way, releasing the address for the same device can be implemented in the following way.

```
void disableSpiMPC23S17(void)
{
#if MCP23S17_SPI_CS_EN_MASK_OR != 0
  MCP23S17_SPI_CS_PORT &=
    (~MCP23S17_SPI_CS_EN_MASK_OR);
#endif
#if MCP23S17_SPI_CS_EN_MASK_AND != 0xFF
  MPC23S17_SPI_CS_PORT |=
    (~MPC23S17_SPI_CS_EN_MASK_AND);
#endif
}
```

The following constants have been defined in the hardware.h file: MCP23S17_SPI_CS_PORT, MCP23S17_SPI_CS_EN_MASK_OR, MCP23S17_SPI_CS_EN_MASK_AND. The first constant defines the port of the microcontroller, to which the address line that gives the expander access to the bus is connected. The constant MCP23S17_SPI_CS_EN_MASK_OR determines, which outputs of the port are to be in logical 1 state to address the device, whereas the negative constant MCP23S17_SPI_CS_EN_MASK_AND determines, which bits are to be in logical 0 state to address the device. If a system operating on the SPI bus is addressed by logical 0 state, then the constant ..._SPI_CS_EN_MASK_OR has a zero value, therefore the logical sum bit operation does not change its value. In order to avoid superfluous operations, the conditional compilation directive has been applied. Similarly, if a device is addressed in high state, then the constant ..._SPI_CS_MASK_AND has the value 0xFF and the product bit operation gives no results. To omit such an operation, conditional compilation has also been applied. When changing the way of connecting a given system to the SPI bus, it is sufficient to modify the hardware.h file. Note that some of the functions have been implemented twice in the project (e.g., function enableSpiMPC23S17): in the library files and in the directory of the project itself. Such an approach is possible when the WEAK attribute is applied. When the definition of the function reappears (without WEAK attribute), then it replaces the earlier function with the WEAK attribute. The application of the WEAK attribute is a more efficient alternative as compared to indexes to functions or virtual functions available in the C++ language.

After getting to know the expander's API, a new requirement emerges – the reading of the line number that has to be set to either high or low state. The CLI mechanism provided allows the user to read additional arguments furnished along with the command. Using the attribute argc in type cmdState_t, it is possible to read the index of the last argument. The argument with the index 0 is the name of the command, while arguments with the consecutive indexes are the parameters, with which the command is executed. In SPI, the functions cmdlineGetArgStr, cmdlineGetArgInt and cmdlineGetArgHex are given for CLI. These functions return respectively: sign chain, integer number determined on the basis of the conversion of the sign chain written in decimal format into a numerical value, and integer number determined on the basis of the sign chain written in hexadecimal format/number. Therefore, in order to write a function that sets a given line in port A in the port expander into high state, the number of that line has to be read first. Then, using the logical sum resulting from the state of port A and the bit left-shift of the number 1 by the value of the line number, sets a new state of port A of the expander. The fragment of the code below shows the function that sets the state for port A.

```
static cliExRes_t
  setPortExtAFunction(cmdState_t *state)
{
  if (state->argc < 2)
    return SYNTAX_ERROR;
  uint8_t newState =
    cmdlineGetArgInt(1, state);
  MPC23s17SetDirA(0x00, 0);
  MPC23s17SetPortA(newState, 0);
  return OK_SILENT;
}
```

The static parameter before the type returned by the function denotes that the function is available only in the file, in which it has been declared. This facilitates maintaining order in the code. The aim of the task is to add an additional function that sets a given output line into high or low state according to its number within the port. To execute this task,

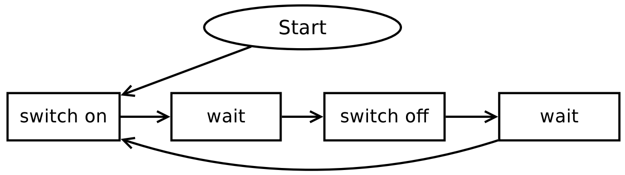


Figure 15. Flash light algorithm

the code of the function presented above can be used.

B. Coroutines FreeRTOS API

Subsequent laboratory classes introduce the issues of cooperative multitasking. Cooperative multitasking is executed with the help of coroutines, which, when used, reduce requirements in system resources. The application of coroutines is followed, however, by certain limitations. The understanding of the idea of coroutines and the specificity of their usage will be facilitated by the laboratory exercise presented below. The exercise involves the introduction of a modification to the software for the executing module so that it will be capable of controlling four light sources. The template for the project is in the directory: `Lab/Coroutines`. The first task is to control the light in such a way as to make the light flash. The algorithm for controlling a single light source is presented in Figure 15.

The algorithm is to execute the following: switching the light on, waiting for requested time, switching the light off and waiting for requested time again. Flashing of three light sources is easy to execute with the help of timers (Atmega168 microcontroller has three timers). The situation is more complicated when the number of light sources to be controlled is greater than the available counters and when each light flashes with a different frequency, independent from other light sources. In such a situation, there are two available options: a suitable program that executes the task can be written, although is not very clear or readable; or, in the most preferable solution, to make use of multitasking offered by operating systems. The realization of the algorithm that controls four light sources and uses multitasking is presented in Figure 16. Each light source is controlled in a separate thread. In the executing modules multitasking is carried out with the help of coroutines, hence each coroutine controls a separate light source, whereas the control algorithm remains the same for all sources. This means that each coroutine can perform the same function. The API of the FreeRTOS system defines for each coroutine the index to the function that is later to be performed by the coroutine. The index to the function of a coroutine has the following form: `void` vACoRoutine(`xCoRoutineHandle` xHandle, `unsigned portBASE_TYPE` uxIndex).

The first argument of the function is a handle to the coroutine. It is used by API functions and macros of the FreeRTOS system that are executed within a function of the coroutine, e.g., to make a coroutine dormant during a given amount of time or to service a queue. The second argument is the index of the coroutine. With reference to the mentioned sample task, each light source has its own coroutine that performs the same function. Each light source has a different index for the coroutine, therefore by performing a common function, on the basis of the argument *uxIndex*, it is possible to determine the light source that the function has to control. For a coroutine to become dormant at a time t, the macro `crDELAY(`*`xCoRoutineHandle`*` xHandle, `*`portTickType`*` xTicksToDelay )` is used. The handle to the coroutine is the first argument, the second being the number of system (internal) clocks. System frequency is defined in the configuration file `FreeRTOSConfig.h`. One system clock includes many microcontroller's internal clocks. Their number is determined automatically by a special macro on the basis of the system frequency `configTICK_RATE_HZ` and the microprocessor clock frequency `configCPU_CLOCK_HZ`. FreeRTOS defines its own types that depend on the processor's architecture [35]. In the case of the Atmega processor, the type *portTickType* is a 16-bit indeterminate integer variable.

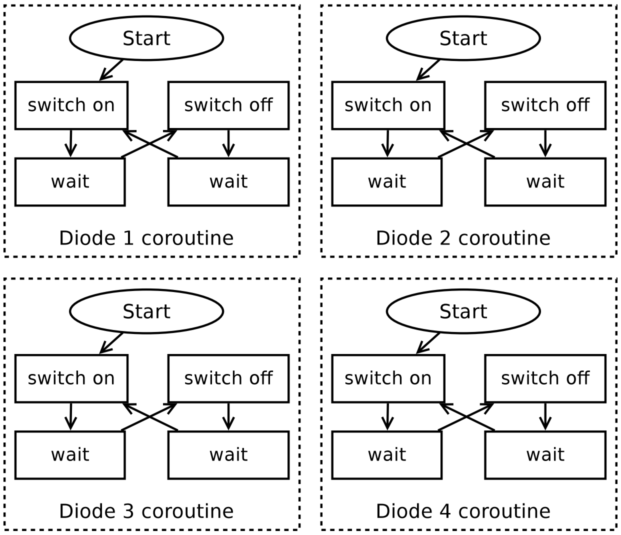


Figure 16. Algorithm for concurrent flashing of four light sources

The code for the coroutine that implements the flashing algorithm can be written in the following way:

```
void vLed(xCoRoutineHandle xHandle,
       unsigned portBASE_TYPE uxIndex )
{
  // This macro is required
  crSTART( xHandle );

  for( ;; )
  {
      ledOn(uxIndex);
      crDELAY( xHandle, tLedOn[uxIndex]);
      ledOff(uxIndex);
      crDELAY( xHandle, tLedOff[uxIndex]);
   }

   // This macro is required
   crEND();
}
```

In the presented code, tLedOn and tLedOff are global tables. Each element of either of the tables denotes the glow discharge time of the diode and the idle time of the diode, respectively. The functions ledOn and ledOff are API functions of the executing module. The function, in which the coroutine is executed conventionally starts with the macro crSTART(xHandle) and ends with the macro crEND. Between the macros there is an infinite loop, in which the appropriate algorithm is executed.

Note that the low memory demand is a distinct advantage of coroutines – they operate on a common stack. Such a solution has a substantial disadvantage as well, which leads to certain limitations in their usage. After the coroutine resumes operation, the values for the local variables defined within the function executing the coroutine may be changed. Thus, if we want the variable to remain stable, such a variable has to be declared as a static or global variable. Thus revealed, this problem makes students stop and think how the compiler operates and in what way it places variables in the memory. Another limitation involves the absence of the possibility of preemption. A coroutine has to decide for itself when it is to be switched, so the only possible option here is the so-called collective multitasking. Switching of coroutines is effected at the time of the execution of blocking calls such as putting the coroutine into a dormant state for a specified time cdDELAY or operations on the buffer (sending or retrieving information from the buffer). All the mentioned operations can be performed only within the block of functions that service the coroutine. These cannot be executed within some other function recalled by the function that services the coroutine. In addition, the functions mentioned cannot be performed within the switch construction.

At the later stage of the envisaged laboratory classes students are asked to perform a task that involves creation of a coroutine that would allow each of the light sources (diode) to falsh with its own frequency. For the coroutine to be created the function portBASE_TYPE xCoRoutineCreate(crCOROUTINE_CODE pxCoRoutineCode, unsigned portBASE_TYPE uxPriority, unsigned portBASE_TYPE uxIndex) is used. The first argument is the function that the coroutine will perform, the second is the priority of the coroutine and the third argument is the coroutine index mentioned earlier in the text. The FreeRTOS text executes tasks or coroutines according to their priority. In the configuration file the number the levels for priorities is set configMAX_CO_ROUTINE_PRIORITIES. The higher the number is, the more operating memory is required by the system. Coroutines themselves are executed within the task or in the idle task. We do not create any tasks in the executing module, thus coroutines are executed in the idle task. This means that it is necessary to add an appropriate function that is executed in the idle task.

```
void vApplicationIdleHook( void )
{
   for( ;; )
   {
      vCoRoutineSchedule();
   }
}
```

The template for the project of the executing controller already includes the code presented above. What is necessary, however, is to create coroutines. This should be done in the main function.

```
portSHORT main( void )
{
   // Initializes hardware,
   // sets ports directions.
   hardwareInit();

   uint8_t ledNo;
   for (ledNo = 0; ledNo<4; ledNo++)
      xCoRoutineCreate(vLed, 0, ledNo);

   vTaskStartScheduler();
   return 0;
}
```

The presented main function ends with the function vTaskStartScheduler. Within this function, a scheduler is activated that performs all the required tasks according to their priorities. In the case of the executing module, there are no tasks, so the scheduler is always set to the idle task, in which coroutines are serviced.

The next proposed laboratory task is to expand the functionality of the software for the executing module with handling of keys. For example, pressing one of the keys will result a diode glowing for some time until the diode goes off. The state of the keys will be checked by separate coroutines. The architecture of the firmware of the executing module is presented in Figure 17.

After pressing of the key is detected, the coroutine sends a relevant message to the coroutine that services the light source. Therefore, it is necessary to introduce communication between the coroutines. This communication can be carried out using message queues. Each coroutine has its

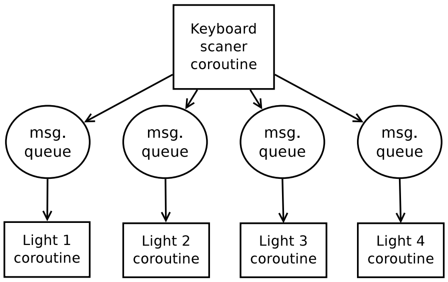


Figure 17. Architecture of executing module firmware controlling four light sources with keyboard

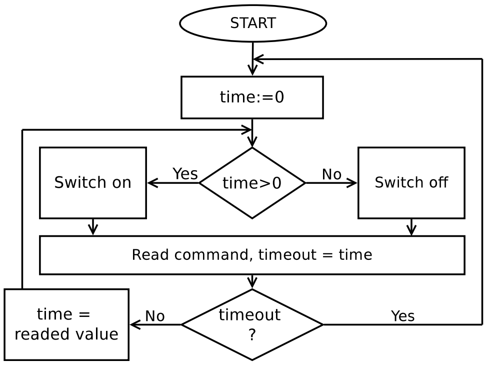


Figure 18. Algorithm for coroutine handling a single light source

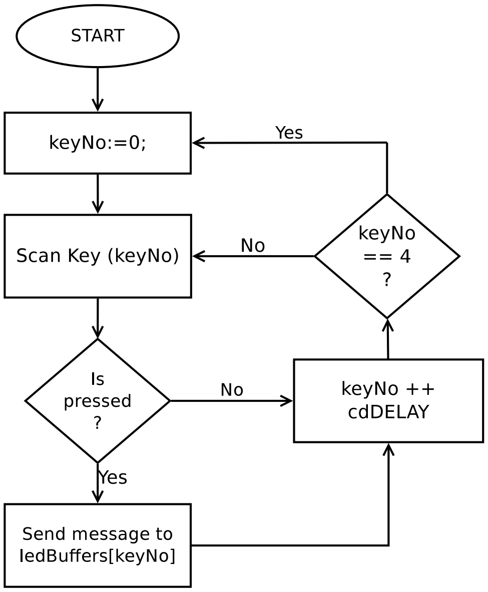


Figure 19. Algorithm for coroutine handling keyboard

own message queue, from which it receives messages. The information in the message includes information on how long the light has to be switched on. If the value is equal to zero, the light has to be switched off.

The algorithm of the coroutine that controls a single light source is shown in Figure 18. Initially – during the initialization phase – the light is switched off, therefore the variable *time* is equal to zero (analogous to message format). In the next step, the coroutine checks the value of the *time* variable. If it is greater than zero, the light is switched on. Otherwise, the light is switched off. Next, the coroutine is waiting for a new message in the buffer, not longer than the time of switching on the light. If the time-out is exceeded and there is no message, the algorithm goes back to the initialization phase and the light will be switched off in the next step. If there is a new message, the light is switched on for a time specified in the message.

FreeRTOS provides a special API for handling semaphores. To read messages the macro *void* crQUEUE_RECEIVE(*xCoRoutineHandle* xHandle, *xQueueHandle* pxQueue, *void* *pvBuffer, *portTickType* xTicksToWait, *portBASE_TYPE* *pxResult) is used. The macro allows the determination of the maximum time-out for a message with the help of the fourth argument. After invoking the macro, the coroutine is suspended for a specified time (optionally specified time-out) or until the message is received. The macro crQUEUE_RECEIVE is then capable of replacing the macro crDELAY.

The macro crQUEUE_RECEIVE requires additional arguments such as a handle to the coroutine (the first argument) and a handle to the queue, from which it will be reading a new message (the second argument). The third argument is the index to the memory, to which the received message will be written. The fourth argument defines the time dedicated for an operation to be performed, whereas the fifth argument is the index to the variable of the type *portBASE_TYPE*. The variable pointed to by pxResult will be set to pdPASS if data has been successfully retrieved from the queue, otherwise it will be set to an error code as defined within ProjDefs.h. Hence, if the variable has the value *pdPASS*, then the light source will be switched off. The light source can be switched off prior to the completion of the specified time-out when a successive message with the time value set to 0 is sent to the queue.

The keyboard coroutine, after detecting pressing of a key, sends a message to an appropriate coroutine through a message queue. For this purpose, the macro *void* crQUEUE_SEND(*xCoRoutineHandle* xHandle, *xQueueHandle* pxQueue, *void* * pvItemToQueue, *portTickType* xTicksToWait, *portBASE_TYPE* * pxResult) has to be used. The importance of individual arguments is the same as in the case of the macro crQUEUE_RECEIVE.

The algorithm for handling keyboard events is presented in Figure 19. The algorithm provides an opportunity to check successively the state of each of the keys. When pressing of a key is detected, then a message is sent to an appropriate message queue through its handle. Handles to message queues are stored in the global table *ledBuffers*. Each element of the table corresponds to a different coroutine that handles a separate light source. To check the state of a key the function *uint8_t* readKey(*uint8_t* keyNo) is used. This function returns zero, when the key is pressed down, and a non-zero result when the key is depressed. The argument keyNo defines the number of the key that is being checked.

To ensure appropriate handling of messages related to keyboard handling, it is necessary to create buffers in the function main with the function *xQueueHandle* xQueueCreate(*unsigned portBASE_TYPE* uxQueueLength, *unsigned portBASE_TYPE* uxItemSize) provided by API of the FreeRTOS system. The first argument determines the length of a single message, whereas the number of messages that the buffer can accommodate is defined by the second argument. It should be noted that in the case of the AVR architecture the type *portBASE_TYPE* is a 8-bit variable and, hence, the length of a queue and the length of a single message cannot exceed 255 bytes. It is also necessary to create in the function main a coroutine that will be responsible for checking the state of the keyboard.

The last task to be performed by students during their classes devoted to handling of coroutines is to secure communication between the executing module and the rest of the system using the RS485 bus. To achieve this, another coroutine that is responsible for handling of the communication protocol has to be added. In line with the architecture shown in Figure 20, the coroutine handling the communication protocol makes use of the queues RS 485 RX and RS 485 TX that, respectively, send and receive data onto and from the RS 485 bus. Each byte that is received on the bus is placed as an individual (separate) message in the RS 485 RX buffer. Similarly, each byte that has to be sent onto the bus is placed by the coroutine handling the communication protocol in the RS 485 TX bus. The coroutine that handles the communication protocol receives successive bytes from the queue RS 485 RX and consolidates them in a message frame. Then, the coroutine checks whether the message address corresponds to the address in the executing module and, with the help of the control code CRC16, whether the message is not a malfunction message. If the received message includes the switch on the light source command or the switch off command, then the coroutine sends a message to an appropriate message queue that services the coroutine handling light source data. The coroutine handling the communication protocol is more complex. Students get its ready-made implementation. The task they are expected to perform is to create queues RS 485 TX and RX, as well as the coroutine handling the communication protocol.

Messages are sent to the RS 485 RX buffer by the function handling Receive Complete interrupt that appears after a byte is received by the serial port. Functions handling interrupts have a spacial API provided by the FreeRTOS system. The macro crQUEUE_SEND_FROM_ISR is designed to send messages to the program buffer. Similarly, messages from the RS 485 TX buffer are read by the function handling the interrupt “Data Register Empty” that determines whether the sending buffer of the serial port can receive another sign to be successively sent on. For this purpose, the macro crQUEUE_RECEIVE_FROM_ISR is used. If the RS 485 TX buffer is empty, then the interrupt “Data register Empty” is switched off. Switching the interrupt off is activated by the macro vInterruptOff() and is effected automatically within this interrupt handling. If the RS 485 TX buffer included any data to be sent, then, before placing them in the hardware sending buffer of the serial port, the MAX481 transmitter is switched on. The transmitter enables sending data onto the RS 485 bus that operates in the half duplex mode. The transmitter is switched off within “Transmit Complete” interrupt handling. The interrupt occurs only when the transmitter has sent all the content of its hardware buffer.

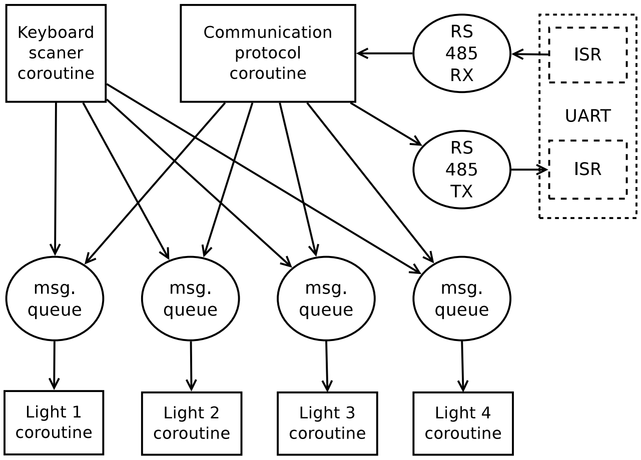


Figure 20. Architecture of executing module firmware controlling four light sources

After adding a new sign to the RS 485 TX sending buffer, interrupt handling “Data register Empty” should be switched on with the macro vInterruptOn(). The programmer must not forget about it. Such an approach has been adopted purposefully having in mind that the RS 485 bus should be blocked for as short a time as possible. A message should be sent only when it formed in full and after being placed in the buffer. Hence, after placing the whole of the message in the buffer, it is necessary to switch the interrupt “Data Register Empty” on. The function handling this interrupt will then switch on the transmitter and will send the message via the bus. On termination of the transmission, the transmitter will be automatically switched off (without the interference of the programmer).

VI. Conclusion and Future Work

The presented didactic system is a valuable addition to the theory of operating and embedded systems. It enables students to get familiarized with such aspects as multitasking, interprocess communication, and process synchronization. The platform has been designed in such a way as to facilitate its quick and easy implementation. For this effect, AVR microcontrollers, which are increasingly popular among students taking interest in electronics, have been used. The presented solution is inexpensive and most students can afford to build the presented platform and use it for didactic or practical purposes limited only by their imagination.

The article puts special attention to a detailed presentation of some selected laboratory exercises prepared for laboratory classes. These exercises familiarize students with practical aspects of issues related to the theory of operating and embedded systems. The presented exercises facilitate successful understanding of techniques of implementing multi-thread applications and the creation of commands and handling of file systems.

The presented didactic platform is still being developed. The future work is related to the implementation of SD card, as well as IPv6 or TCP protocols.

Simultaneously, the achieved platform's simplicity introduces some limitations, mainly concerning the size of random access memory, which cannot be extended. The limited size of random access memory hinders an implementation of the SSH protocol that supports the encrypted connections.

Acknowledgement

The authors would like to thank all the developers taking part in open source projects cited in the article.

References

[1] A. Kaliszan and M. Głąbowski, "Didactic embedded platform and software tools for developing real time operating system," in *Proceedings of the The Seventh Advanced International Conference on Telecommunications (AICT 2011)*, M. Głąbowski and D. K. Mynbaev, Eds. St. Maarten, The Netherlands Antilles: IARIA, Mar. 2011, pp. 77–82.

[2] R. Love, *Linux Kernel Development*, 3rd ed. Novell Press, Jul. 2010.

[3] "Microsoft Windows Server." [Online]. Available: http://www.microsoft.com/windowsserver2008/en/us/default.aspx <retrieved: Jan, 2012>

[4] W. R. Stevens and S. A. Rago, *Advanced Programming in the UNIX Environment*, 2nd ed. Addison-Wesley, 2005.

[5] J. M. Corchado, J. C. Augusto, and P. Novais, *Ambient Intelligence and Future Trends*, 1st ed., ser. Advances in Intelligent and Soft Computing. Springer, 2010, vol. 72. [Online]. Available: http://www.springer.com/engineering/computational+intelligence+and+complexity/book/978-3-642-13267-4 <retrieved: Jan, 2012>

[6] L. Sydell, "Chasing a habitable 'home of the future'," May 2006. [Online]. Available: http://www.npr.org/templates/story/story.php?storyId=5360871 <retrieved: Jan, 2012>

[7] "Mach homepage." [Online]. Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/mach/public/www/overview.html <retrieved: Jan, 2012>

[8] A. Tevanian, Jr., R. F. Rashid, D. B. Golub, D. L. Black, E. Cooper, and M. W. Young, "Mach threads and the unix kernel: The battle for control," in *in Proceedings of the USENIX Summer Conference, USENIX Association*, 1987, pp. 185–197.

[9] A. Singh, *A Technical History of Apple's Operating Systems*. osxbook.com, 2001.

[10] R. Stallman, "The GNU manifesto," *Dr. Dobb's Journal*, vol. 10, no. 3, p. 30, Mar. 1985.

[11] ——, "GNU manifesto." [Online]. Available: http://www.gnu.org/gnu/manifesto.html <retrieved: Jan, 2012>

[12] "100 of the most significant events in linux history," *Linux Journal*, Aug. 2001. [Online]. Available: http://www.linuxjournal.com/article/6000 <retrieved: Jan, 2012>

[13] "FreeBSD homepage." [Online]. Available: http://www.freebsd.org <retrieved: Jan, 2012>

[14] "Ethernut homepage." [Online]. Available: http://www.ethernut.de <retrieved: Jan, 2012>

[15] "Arduino homepage." [Online]. Available: http://arduino.cc <retrieved: Jan, 2012>

[16] "Arduino programming language." [Online]. Available: http://arduino.cc/en/Reference/HomePage <retrieved: Jan, 2012>

[17] "Wiring homepage." [Online]. Available: http://wiring.org.co <retrieved: Jan, 2012>

[18] "The FreeRTOS project homepage." [Online]. Available: http://www.freertos.org <retrieved: Jan, 2012>

[19] FreeRTOS, "Copyright notice," http://www.freertos.org/copyright.html <retrieved: Jan, 2012>

[20] "Eagle." [Online]. Available: http://www.cadsoftusa.com/ <retrieved: Jan, 2012>

[21] A. Kaliszan, "AtMega128 RTOS hardware repository." [Online]. Available: http://rtosOnAvr.yum.pl/hardware/ssw <retrieved: Jan, 2012>

[22] Atmel, "Atmega128 data sheet." [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf <retrieved: Jan, 2012>

[23] ——, "Atmega168 data sheet." [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf <retrieved: Jan, 2012>

[24] FTDI, "FT232RL data sheet." [Online]. Available: www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf <retrieved: Jan, 2012>

[25] Maxim, "MAX481 data sheet." [Online]. Available: http://datasheets.maxim-ic.com/en/ds/MAX1487-MAX491.pdf <retrieved: Jan, 2012>

[26] Microchip, "Enc28j60 data sheet." [Online]. Available: ww1.microchip.com/downloads/en/devicedoc/39662a.pdf <retrieved: Jan, 2012>

[27] ——, "MPC23S17 data sheet." [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf <retrieved: Jan, 2012>

[28] Texas Intruments, "ULN2003A data sheet." [Online]. Available: http://focus.ti.com/lit/ds/symlink/uln2003a.pdf <retrieved: Jan, 2012>

[29] Microchip, "MCP3008 data sheet." [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/21295d.pdf <retrieved: Jan, 2012>

[30] National, "LM35 data sheet." [Online]. Available: http://www.national.com/ds/LM/LM35.pdf <retrieved: Jan, 2012>

[31] Maxim, "DS1305 data sheet." [Online]. Available: http://datasheets.maxim-ic.com/en/ds/DS1305.pdf <retrieved: Jan, 2012>

[32] G. Socher, "AvrUsb500v2 – an open source Atmel AVR programmer, stk500 v2 compatible, with USB interface." [Online]. Available: http://tuxgraphics.org/electronics/200705/article07052.shtml <retrieved: Jan, 2012>

[33] "AVR JTAG ICE clone." [Online]. Available: http://www.scienceprog.com/build-your-own-avr-jtagice-clone <retrieved: Jan, 2012>

[34] A. Kaliszan, "AtMega128 RTOS firmware repository." [Online]. Available: http://rtosOnAvr.yum.pl/software/FreeRtos <retrieved: Jan, 2012>

[35] FreeRTOS, "FreeRTOS API reference." [Online]. Available: http://www.freertos.org/a00106.html <retrieved: Jan, 2012>

[36] P. Stang, "Procyon AVRlib API," 2006. [Online]. Available: http://www.procyonengineering.com/embedded/avr/avrlib/ <retrieved: Jan, 2012>

[37] "AVR-libc API," 2006. [Online]. Available: http://avr-libc.nongnu.org <retrieved: Jan, 2012>

[38] G. Socher, "HTTP/TCP with an Atmega88 microcontroller (AVR web server)," 2006. [Online]. Available: http://www.tuxgraphics.org/electronics/200611/embedded-webserver.shtml <retrieved: Jan, 2012>

[39] "GNU automake." [Online]. Available: http://www.gnu.org/software/automake <retrieved: Jan, 2012>

[40] "Cross platform make." [Online]. Available: http://www.cmake.org <retrieved: Jan, 2012>

[41] "GNU make." [Online]. Available: http://www.gnu.org/software/make <retrieved: Jan, 2012>

www.iariajournals.org

International Journal On Advances in Intelligent Systems
ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2679

International Journal On Advances in Internet Technology
ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
issn: 1942-2652

International Journal On Advances in Life Sciences
eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
issn: 1942-2660

International Journal On Advances in Networks and Services
ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
issn: 1942-2644

International Journal On Advances in Security
ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS
issn: 1942-2636

International Journal On Advances in Software
ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2628

International Journal On Advances in Systems and Measurements
ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
issn: 1942-261x

International Journal On Advances in Telecommunications
AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
issn: 1942-2601