A Practical Approach to Speech Processor for Auditory Prosthesis using DSP and FPGA

V.Bhujanga Rao
CC(R&D), DRDO HQ
Ministry of Defence,
Govt of Inida, Delhi
919866441074
vepcrew1@rediffmail.com

P.Seetha Ramaiah
Andhra University
Dept. of Computer Science and
Systems Engg,Vishakhapatnam, India
919848224505
psrama@gmail.com

K.Raja Kumar
Andhra University
Dept. of Computer Science and
Systems Engg, Visakhapatnam, India
919490132765
krajakumar@yahoo.com

ABSTRACT

Auditory prosthesis (AP) is a widely used electronic device for patients suffering with severe hearing loss by electrically stimulating the auditory nerve using an electrode array surgically placed inside the inner ear. The AP also known as cochlear implant (CI) mainly contains external Body worn Speech Processor (BWSP) and internal Implantable Receiver Stimulator (IRS). BWSP receives an external sound or speech and generates encoded speech data bits for transmission to IRS via Radio Frequency transcutaneous link for excitation of electrode array placed in the inner ear. Development of BWSP and IRS involves normally the use of either standard microprocessor or microcontroller or digital signal processor (DSP) or FPGA or ASIC devices. Sometimes the performance of the AP system using the standard processors cannot meet the requirement of the intended application. As the selected DSP processor (ADSP2185) from Analog Devices Inc. solely cannot perform the purpose of the speech processor for auditory prostheses, the Xilinx FPGA is added to fulfill the requirement. Combination of a standard processor such as DSP and FPGA may lead to the solution in both prototyping and target operational system. The ADSP2185 processor is used to realize the Continuous Interleaved Sampling (CIS) algorithm for speech signal processing and FPGA is used to realize the speech data encoding algorithm. This paper introduces practical implementations of digital speech processor for use in AP based on DSP ADSP-2185 and Xilinx's FPGA Spartan 3 with the description of practical data. The combination of ADSP2185 and FPGA is used to develop Speech Processor for an auditory prosthesis. FPGA implementation of speech data encoder is initially simulated using ModelSim and interfaced with ADSP2185. The entire embedded application is tested with real time speech signals by using laboratory model IRS and satisfactory results are observed.

Keywords: Auditory prosthesis/cochlear implant, Speech Processor, DSP, FPGA.

1. Introduction

Development of embedded applications normally requires the use of either standard microprocessor or microcontroller or digital signal processor (DSP). Design of embedded system involves the partition of target system into hardware and software implementation parts. The price and performance requirements represent major criteria to choose between hardware and software implementation of the solution. Standard single-chip microcomputers or DSP's often sufficient for application requirement, and only software has to be developed for a given application. If additional hardware interfacing is required, it is usually implemented using standard specialized chips, or using general MSI/SSI chips. This leads to a static solution and the PCB is to be redesigned for any alterations. Field-Programmable static RAM based Gate Arrays offer easy reprogrammability and change of the function without change of the PCB design using simple downloading of a new bit stream representing new circuit design. The feature of dynamic reconfigurability is used to change the dynamically the function of an FPGA in a time-multiplexed manner [1]. FPGA is a programmable logic device that supports implementation of relatively large logic circuits. DSPs are a type of specialized processor with customized architectures to achieve high performance in signal processing applications. DSPs usually include hardware support for fast arithmetic, including single cycle multiply instructions (often both fixed-point and floating-point),

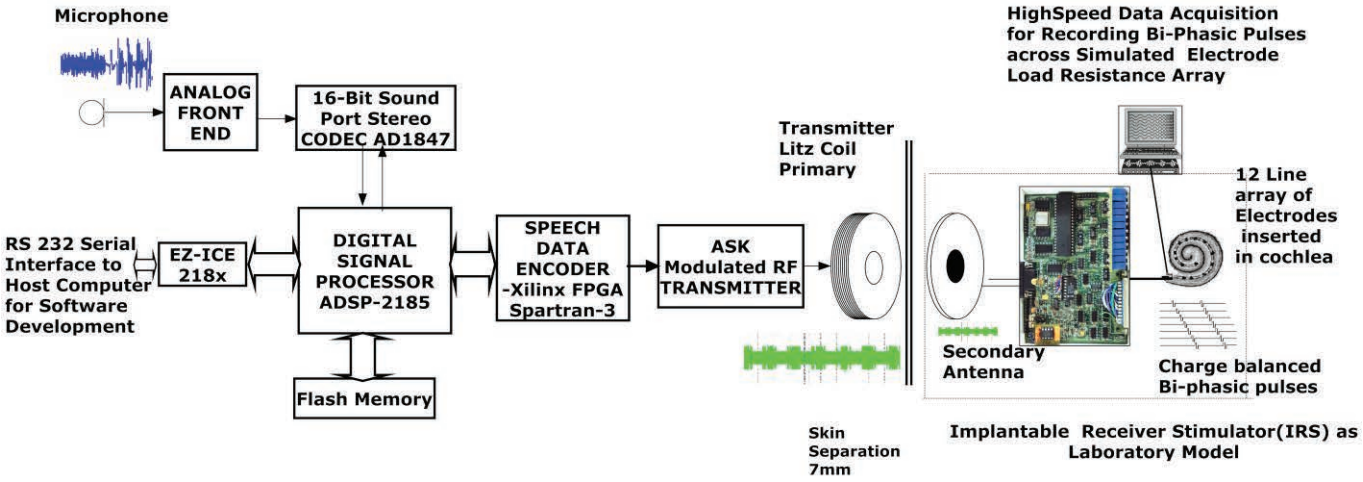


Figure 1: Functional block diagram of Speech Processor for Auditory Prosthesis.

large (extra wide) accumulators, and hardware support for highly pipelined, parallel computation and data movement. To keep a steady flow of operands available, DSP systems usually rely on specialized, high bandwidth memory subsystems. The complementary capabilities of DSPs and FPGAs integrated into high-density systems will continue to evolve to meet the growing challenges of high-complexity signal processing applications.

The auditory prosthesis has recently emerged as clinically acceptable prosthesis for aiding people who suffers from a profound to total sensorineural hearing loss. The multi-channel auditory prosthesis system, based on research undertaken at the Andhra University and Naval Science and Technological Laboratory, Visakhapatnam [2-5], uses an external BWSP and surgically implanted prosthesis to stimulate auditory neurons with biphasic pulses via an electrode array placed in the scala tympani of the cochlea (inner ear). The performance of the AP depends on the various parameters such as number of electrodes, placement of electrodes, types of electrodes, stimulation, and the speech processing strategies to deliver the important acoustic features to understand the sound under noisy environment. Development of high performance speech processor for auditory prostheses for AP involves careful selection of the parameters.

The development of auditory prostheses involves the potpourri of electronic systems, signal processing, mechanical engineering, physiology, electronics engineering and computer science and engineering [6-9]. Signal processing plays an important role in the development of different techniques for deriving electrical stimuli from the speech signal. Developing speech or sound signal processing algorithms that would help in mimicking the function of a normal cochlea in inner ear is the biggest challenge for the signal processing engineers. The function of Auditory Prostheses is an artificial replacement of damaged inner ear using external BWSP and IRS for stimulating auditory nerve via electrode array that enables understanding the speech by brain. The BWSP receives an external sound or speech and generates encoded speech data bits for transmission to receiver-stimulator via an inductive Radio Frequency (RF) transcutaneous link. The IRS receives the encoded speech data bits via RF receiver, decodes the speech data and electrically stimulates the corresponding electrode of line electrode array.

The working principle of the external BWSP is as follows: the microphone is used for picking up sound wave for converting into an electrical signal by a device – CODEC (CODER-DECODER). CODEC is basically a combination of a high speed ADC-DAC, which picks up analog sound signal from microphone, converts into digital samples and sent to the DSP for speech processing using CIS algorithm. The incoming sound signal is divided into 8 frequency bands (or channels) with center frequencies ranging from 250 to 7500 Hz. The output of each filter is rectified and low pass filtered with a cutoff frequency of 200 Hz. After computing all 8-filter outputs, maximum value of each output is logarithmically compressed to fit the patient's electrical dynamic range. The compressed information is sent to the Speech Data Encoder(SDE) where it receives the data bytes in parallel, encodes the stimulation parameters and sends data serially bit by bit to the RF transmitter. The RF transmitter is based on ASK modulation which modulates the incoming signal form SDE and transmitted to the prosthetic implant (IRS) through the RF inductive coils. The laboratory model for IRS consists of RF receiver, speech data decoder, current stimulator, switch matrix and simulated resistance electrode array. The RF receiver receives the ASK modulated serial speech data through RF Coils, demodulates and recovers the serial data. The recovered data signal is given to the speech data decoder in receiver-stimulator where data is decoded and the desired electrodes are stimulated in accordance with the stimulation parameters to activate the remaining auditory neurons in the inner ear, restoring hearing sensations partially.

Two DSP processors :Motorola DSP56001 operating at 32MHz and TMS32C50 operating at 40MHz are popularly used for implementing speech processing algorithms including communication protocol and speech data encoding without FPGA/CPLD[10-14]. This paper addresses the implementation of speech processor using ADSP2185 DSP processor and Xilinx Spartan 3 FPGA. FPGA implementation for communication protocol and speech data encoding is relatively easy with a flexibility to adapt for new receiver-stimulator that needs different data rates, protocol and encoding schemes. This may lead to improvement in performance.

Due to the processing limitations of the current selected DSP - ADSP2185, we add Xilinx based FPGA as a co-processor to perform the tasks that are not done by ADSP2185. The tasks that are less frequently changed are distributed to ADSP2185. The tasks that are frequently changed and the tasks that are to be modified regularly and for future additional functions are implemented in FPGA. The ADSP2185 performs the following functionalities on speech signal such as band pass filtering, rectification, low pass filtering and compression suitable for patient's dynamic range. The tasks of FPGA are to encode the compressed 8 BPF outputs and transmit the encoded information serially bit-by-bit via RF transmitter. In brief, the Speech processor performs the implementation of CIS algorithm and the FPGA performs the Speech Data Encoder functions. This paper offers a practical approach for embedded system design that builds upon the potential of Hardware-Software co-design methodology for FPGA based Speech Data Encoder. The hardware design issues for Speech Processor for Auditory Prosthesis (SPAP) are presented in section-2. Section 3 deals with the software design issues of SPAP. Finally, the experimental results are given in section 4.

2. Hardware Design Issues for Speech Processor.

The main functions of SPAP are speech signal processing, speech data encoding and transmission of encoded speech data to IRS via transcutaneous RF inductive link. The hardware design requirements of SPAP are : (i) to sample continuously speech or sound signals from environment by the microphone of analog front end that carries the speech or sound signal to the speech processor, (ii) Speech Processor processes the signal into n ($4 \leq n \leq 8$) different bands/channels corresponding to n active electrodes inserted in cochlea, based on CIS speech processing algorithm using CODEC and DSP, (iii) DSP generates 16 data bytes -(8 electrode number bytes with their respective 8 bytes of electrode current units that correspond to 8 spectral maximum values in 8 bands of sampled speech signal) continuously, (iv) 16 speech data bytes from DSP are transferred on interrupt basis to Speech Data Encoder(SDE) that encodes 16 speech data bytes as per the synchronous serial communication protocol format, (v) Encoded serial speech data bytes are used by ASK modulator for transmitting to the IRS via inductive RF communication link for

stimulating the electrodes inserted inside damaged cochlea. The hardware functional block diagram of BWSP is shown in Figure 1. The Analog Front End (AFE) is used to amplify the incoming low amplitude speech signal from microphone. SPAP receives the analog speech signal from AFE, extract the spectral and temporal information by implementing the most popularly used CIS Speech processing algorithm. The processed information is fed to the SDE where it is encoded for stimulation and transmitted serially to the ASK modulated RF transmitter. The RF transmitter is based on ASK modulation which modulates the incoming signal from speech data encoder and transmitted to the IRS of CI fabricated as laboratory model through inductively coupled coils.

2.1 Description of Hardware functional blocks

Analog Front End: Analog Front End consists of 2-stage pre-amplifier followed by Automatic Gain Control (AGC) and a last stage amplifier for amplifying the low amplitude incoming signal from microphone.

2.1.1 CODEC

Coder-Decoder chip AD1847 is used to sample and convert the incoming speech signal via AFE into a 16-bit digital value being processed by ADSP2185. The CODEC receives the amplified input signal from analog front end and samples the input sound signal with the sample rate of 11025 Hz specified in control register, and transmits serially to the ADSP2185 with the 16-bit mono format.

2.1.2 Speech Processor

ADSP-2185 receives the 16-bit sample of speech information from AD1847 CODEC via its serial port SPORT0, processes the sample based on 4 to 8 channel CIS speech processing algorithm (programmable 4 to 8 digital FIR band-pass filtering over incoming speech signal sample followed by full-wave rectification, FIR low-pass filtering and power law compression). The output of CIS algorithm is a sequence of 16 speech data bytes that corresponds to 8 channel/electrode number bytes with their respective 8 charge bytes and stored in data memory of ADSP-2185. Once 16 bytes are in data memory, ADSP-2185 interrupts SDE using the connection of active low IOMS output of ADSP2185 to active low input INT0 of SDE. SDE in its interrupt service routine receives 16 speech data bytes via programmable flags PF0 to PF7 of ADSP2185 connected to FPGA

2.1.3 SDE

The required tasks of SDE are to (i) receive the processed speech data bytes (EL# - Electrode Number, CH# - Electrode Charge) from ADSP2185 for sampled frame of speech signal, and store in the internal RAM, (ii) encode stored speech data bytes based on basic synchronous serial communication protocol format with single character sync byte and (iii) send serially each frame to the ASK modulated RF transmitter at high rate (>100Kbps) to meet the requirement of frame by frame reception of speech signal and frame by frame stimulation of electrodes without loss of data frame.

2.1.4 ASK Modulator

The output of SDE as serial data bits of processed speech signal is given as input to ASK modulator that generates ASK signal for wireless transmission to IRS. Wireless power and data from BWSP to IRS implant module are transcutaneously transmitted using inductively coupled RF link between the transmitter coil of BWSP and the receiver coil of IRS.

2.2 VHDL Model of Speech Data Encoder

VHDL model of the proposed SDE has been developed. Top level entity consists of 16 input ports as shown in Figure 2. Out of these eight inputs are served as data port pins for reading the processed speech data bits that contains the information about electrode number and its corresponding charge. As the selected DSP processor Analog Devices ADSP2185 solely cannot perform the purpose of the speech processor for auditory prostheses, the Xilinx FPGA is added to fulfill the requirement. Combination of a standard digital signal processor ADSP2185 and Xilinx FPGA Spartan–3 is used to implement Speech Processor. As a laboratory model ADSP2185 processor is used to realize the Continuous Interleaved Sampling (CIS) speech processing algorithm and FPGA is used to realize the speech data encoding. Interfacing between FPGA and ADSP2185 is shown in Figure 2. Eight data lines from ADSP2185 (D_0-D_7) is interfaced as 8 data lines to Spartan 3, four address lines (A_0-A_3) and two control lines (active low WR and active low IOMS). Data transfer from ADSP2185 to Spartan 3 is based on the interrupt (active low IOMS) initiated by ADSP2185 for every 1ms time interval with lowering the write signal after 8-bit data and 4-bit address is placed on the data and address lines.

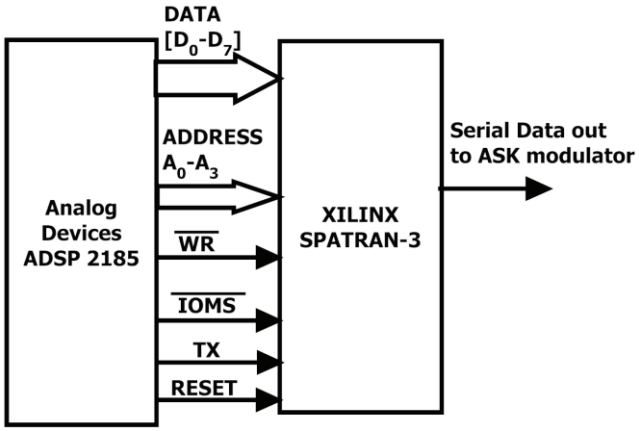


Figure 2: VHDL model of SDE interfaced with FPGA

3. Software Design Issues for Speech Processor

Software requirement for the operation of SPAP are given below

(i) CIS speech processing algorithm implementation
  (a) Digital FIR band pass filtering for speech data bytes received from CODEC by DSP
  (b) Envelope detection (Rectification and low pass filtering
  (c) Buffering 8 BPF outputs.

(ii) FPGA based SDE
  (a) Receiving and storing the processed speech data from ADSP2185
  (b) Speech data encoding as per synchronous serial communication protocol format

(iii) Serial data output for transmission of encoded speech data bytes for every 1mS speech sample.

These requirements are pictorially represented as a structured chart shown in Figure 3.

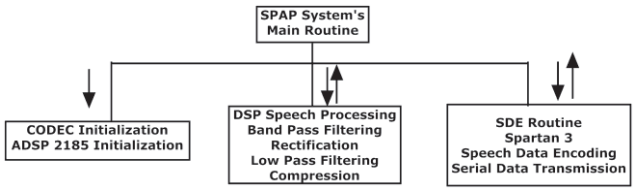


Figure 3: Structure chart for Speech Processor Software

3.1 CIS Speech processing algorithm implementation

The speech signal processing program for BWSP is developed as embedded program and coded in ADSP-2185 assembly language under EZ-KIT Lite/Visual DSP++3.5 IDE. CODEC is configured to sample the incoming speech signal at 11025 samples per second. There are 11025 samples /sec (i.e approximately 11samples for 1ms) as the speech input is sampled at 11.025 KHz. Since the selected electrode stimulation rate is fixed at 1000pps/channel, it is required to find sample value typically maximum in every 11 samples of every 1ms. Each received sample is processed in the following sequential stages: band-pass filtering, rectification, and low pass filtering. The temporal envelope in each channel is extracted with full-wave rectifier and low pass filters were designed to smoothen the amplitude variations with a cutoff frequency of 200Hz to allow maximal transmission of temporal envelope cues while avoiding aliasing when a relatively low carrier rates are used. After 11th sample is processed, ADSP2185 sends processed information of electrode numbers and charges of 8 channels to the SDE with 840 cycles of ADSP-2185 processor left as spare after CIS processing and communication with SDE leading to 1ms latency time from the input speech sample to an output bi-phasic pulse. A power-law transformation is used to map the relatively wide dynamic range of derived envelope signals onto the narrow dynamic range of electrically-evoked hearing. The acoustic envelope of amplitude '*x*' (1000 as minimum to exclude noise floor and 32565 as maximum of 15-bit value as per 1.15 format of ADSP2185 processor: 60dB to 90dB dynamic range) is mapped to the electrical amplitude '*y*' (1 to 255 of 8-bit unsigned value: 0dB to 48 dB) according to the power-law relation

$$y = Ax^p + B$$

The constants A and B are chosen such that the input acoustic range [x_{min} , x_{max}] is mapped to the electrical dynamic range [THL, MCL], where THL is the Threshold Hearing level and MCL is the Most Comfortable Level of hearing of the respective patient. Threshold is the highest stimulation at which no sound sensation occurs and ensure that the patient does not hear the THL level stimulation even in quite. Maximum Comfort Level is the highest stimulus level at which sound is loud but still comfortable. 50 for THL and 100 for MCL are chosen as default values for any patient. These stimulation levels ensure the safe and acceptable electrical stimulation levels to understand the speech/sound signals fitted by identification of THL and MCL with several iterations to adjust the speech processor for each individual patient by an experienced audiologist [16]. Actual values are programmed by the audiologist using clinical programming. For power law compression function, the constants A and B can be computed as follows:

$$A = \frac{MCL - THL}{x_{\max}^p - x_{\min}^p}$$

$$B = THL - Ax_{\min}^p \qquad \text{Where } p<=0.0001$$

After the power-law compression, every processed sample is compared with every previous processed sample and its maximum value is stored in the buffer until all samples are processed. The same process is carried out for all 8 channels that contain band-pass filtering, rectification, and low pass filtering with the same input sample.

The stored processed samples of all 8 channels are sent to the SDE via parallel interface. BWSP sends the processed information to SDE for every 1.00 milliseconds which in turn drive the electrodes (simulated electrode resistances) at 1000pps/electrode via transcutaneous RF link and IRS.

3.2 FPGA based implementation of SDE

The SDE has been implemented as speech data receiving and encoding on Xilinx Spartan 3. The register transfer level (RTL) description of the Speech Data encoder was implemented on Xilinx Spartan 3 using VHDL behavioral description. The SDE implementation consists of (a) Realization of a dual port RAM for storing the received processed speech data from ADSP2185 that contains the electrode number along with its corresponding excitation charge for 12 electrodes and (b) formats the processed speech data bits (Speech data encoding) as per synchronous serial communication protocol format and to implement parallel to serial converter to transmit the encoded data serially to the RF transmitter.

Table 1 - Device Utilization Summary (xc3s200-4pq208)

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	148	3,840	3%
Number of 4 input LUTs	286	3,840	7%
Logic Distribution			
Number of occupied Slices	225	1,920	11%
Number of Slices containing only related logic	225	225	100%
Number of Slices containing unrelated logic	0	225	0%
Total Number of 4 input LUTs	418	3,840	10%
Number used as logic	286		
Number used as a route-thru	132		
Number of bonded IOBs	12	141	8%
Number of BUFGMUXs	1	8	12%

The programs are realized to store the received data from the DSP unit into the Dual Port RAM, at the Xilinx- FPGA kit frequency and transmit the contents of RAM at 172kbps baudrate serially to the RF transmitter. Only after the data is read from all the 16 registers of the read port, it gets refreshed with the data present at that particular instant for the write port. All the data inputs are required for SDE are received from ADSP2185. The SDE performs two functions (a) store the data from D0-D7 of ADSP2185 and (b) transmits the data available on RAM serially to RF transmitter. The sequence of operations for storing the data is (i) IOMS signal should be low, which makes the enable of RAM high, as is passed through a NOT gate, (ii) Data available from D0-D7 are transferred to the register specified by the address line and (iii) write signal should be low during storing. After all the data bytes are stored in dual port RAM, the SP makes the transmit signal is high, the data stored on RAM is read and transmitted serially bit by bit. After all data stored in RAM are transmitted, the SDE is ready to receive the command from ADSP2185. A bit stream pattern was generated using VHDL model of the SDE and finally downloaded on Xilinx Spartan 3 FPGA chip using JTAG interface. Table-1 shows the resource utilization summary of Spartan 3 FPGA chip

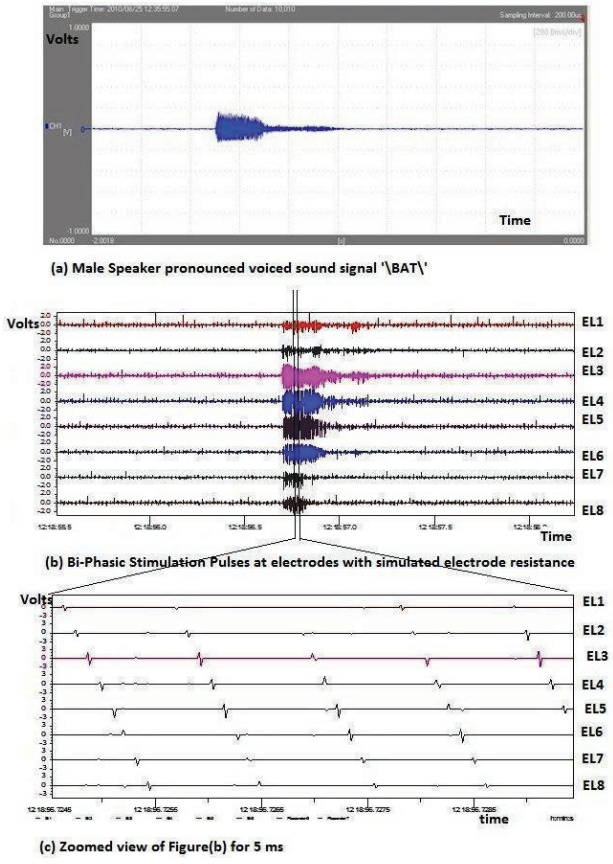


Figure 4. Measured waveforms at various points in the system. Top panel shows speech input token for vowel sound "BAT" measured at an output node of the analog preprocessor, and the middle panel show stimuli for the eight channels of stimulation with 5k resistors as loads. The bottom panel shows the interlacing of stimulus pulses using an expanded timescale. Waveforms in (c) are recorded as triangular instead of rectangular pulses due to the use of low speed data acquisition system (WaveBook 512A, IO Tech Inc, USA).

4. Experimental results

A Laboratory test station is arranged as a test setup to conduct testing experiments for a total CI system: BWSP, RF Link with variable inter-coil antenna distance from 5mm to 10mm, IRS, and load for electrode stimulation with i) 5KΩ resistive loads connected between eight active electrode nodes and reference electrode node in place of the twelve line electrode array and ii) an electrode line array of 12 platinum-iridium rings in ringer solution that exhibits impedance properties equivalent to electrode nerve interface. Several test experiments were conducted and the results recorded. The results were analyzed and found as expected. Figure 4 shows waveforms at various points in the CI system for the experiment of male speaker pronounced vowel sound "BAT". The top panel shows speech input ("BAT") measured in an output node of the analog front end and the middle panel shows stimuli of 8 channels with 5KΩ simulated electrode resistance loads. The bottom panel shows the most of the stimulation to 2, 3, 4, 5, and 6 electrodes and remaining electrodes 1, 7 and 8 electrodes with lower stimulation using an expanded time scale.

5. Discussions

The advantage of using FPGA's is that the embedded designer can create special purpose functional units that can perform limited intended tasks very effectively and efficiently. FPGAs can be reconfigured dynamically as well based on the requirement. FPGA based design is worth considering if the desired performance cannot be achieved using the DSP processor alone. The speech processor is designed as laboratory model for practical implementation of new speech processing algorithms in ADSP2185 DSP processor, protocols and encoding schemes in FPGA and their functional verification using SDE and RS. The BWSP is battery powered (3.7V, 3100mAH Li-Ion) with +5V generation using LM2621 (step-up DC-DC switching regulator) and overall current dissipation of the entire system is 150 mA that corresponds to 20 hours of continuous operation using 3100mAH lithium-ion rechargeable battery, allowing a patient to use the BWSP for a full day without recharging. The final product model of BWSP has dimensions as 110mm X 55mm X 30mm with the weight of 125grams, whereas Sprint speech processor of Cochlear Limited, Australia has dimensions of 103mm X 67mm X 23 mm with the weight of 146 grams [15] and Wearable Speech Processor of Nano Bioelectronics and Systems Research Center (NBS-ERC) of Seoul National University, Korea with dimensions of 82mm X 49mm X 19mm including the 1800 mAH rechargeable battery with approximately 17 hours of continuous operation[8].

6. Conclusion

The DSP and FPGA based speech processor for an auditory prosthesis is implemented. The speech processing algorithm CIS is implemented and processed information is encoded by FPGA based speech data encoder. FPGA based Dual-Port RAM for serial transmission is developed using VHDL. The parallel to serial transmission developed by using VHDL is verified by using high-speed data acquisition module WAVEBOOK and DasyLab software. FPGA based implementation is more attractive in the realization as the realization is more economical and easily reconfigurable. The cochlear implant product development (Indian Cochlear Implant System-ICIS) based on developed prototype model of CI System is ongoing by Defence R and D Organization - Naval Science and Technological Laboratory, Visakhapatnam with

the involvement of the following organizations: Teknosarus Embedded Systems Pvt. Ltd, Hyderabad for BWSP and Headset coil, Advanced Numerical Research and Analysis Group (ANURAG), Hyderabad for ASIC design and development and fabrication. Once the product is developed, plans are ongoing for animal testing and approval from Indian Health Authority – Central Drug Standards Control Organization (CDSCO). Implementation of the FPGA based complete SPAP is under progress with the aim of reducing the size and more features.

7. Acknowledgments

The findings reported in this paper form part of a joint project on development of cochlear implant device between Andhra University and Naval Science and Technological Laboratory, Visakhapatnam. The authors gratefully acknowledge the suggestions and insights provided by Prof Stephen J. Rebscher, University of California, San Fransisco, Prof B.S. Wilson, Adjunct Professor, PRATT School of Engineering, Duke University, Prof SJ Kim, School of Electrical Engineering and Computer Science and the Nano Bioelectronics Systems Research Center, Seoul National University and Prof. Philip Loizou, University of Texas, Dallas, during the development of prototype of CI System. The authors also acknowledge the invaluable and rich contribution of Mr. V.P. Rao, former Scientist-F and Mr. M. Mohan Krishna, Scientist-G of Naval Science and Technological Laboratory, Visakhapatnam during the development of CI System.

8. References

[1] F. Berthelot, F. Nouvel, and D. Houzet, "Design methodology for runtime reconfigurable FPGA: from high level specification down to implementation," in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS '05)*, vol. 2005, pp. 497–502, Athens, Greece, November 2005.

[2] K. Raja Kumar and P. Seetha Ramaiah, "Programmable Digital Speech Processor for Auditory Prostheses", Proceedings of IEEE International Region-10 Conference on Innovative Technologies for Societal Transformation – TENCON, 2008, November 18 - 21, 2008, Hyderabad, India.

[3] K. Raja Kumar and P. Seetha Ramaiah, "Microcontroller Based Receiver Stimulator for Auditory Prosthesis", Proceedings of IEEE International Region-10 Conference on Innovative Technologies for Social Transformation – TENCON, 2008, November 18 - 21, 2008, Hyderabad, INDIA

[4] K. Rajakumar and P. Seetha Ramaiah, "Personal Computer Based Clinical Programming Software for Auditory Prostheses", Journal of Computer Science, Vol 5. No. 8, pp. 589-595, 2009

[5] K. Raja Kumar and P. Seetha Ramaiah, "Development of Receiver Stimulator for Auditory Prosthesis", International Journal of Computer Science Issues, Vol. 7, Issue 3, No 2, May 2010.

[6] P. Loizou, "Mimicking the Human Ear: An overview of signal processing techniques used for cochlear implants," IEEE Signal Processing Magazine, 15(5), 101-130.

[7] Fan-Gang Zeng , "Trends in Cochlear Implants", Trends In Amplification Volume 8, Number 1, 2004, pp T1- T34.

[8] Soon Kwan An, Se-Ik Park, Sang Beom Jun, Choong Jae Lee, Kyung Min Byun, Jung Hyun Sung, Blake S. Wilson, "Design for a Simplified Cochlear Implant System", *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 6,pp . 973-982, June 2007

[9] B. S. Wilson, S. Rebscher, F. G. Zeng, R. V. Shannon, G. E. Loeb,D. T. Lawson, and M. Zerbi, "Design for an inexpensive but effective cochlear implant," Otolaryngoly.—Head Neck Surg., vol. 118, no. 2, pp. 235–241, Feb. 1998.

[10] Zerbi, M. Wilson, B.S. Finley, C.C. and Lawson, D.T, "A Flexible Speech Processor Cochlear Implant Research", Proceedings of the Digital Signal Processing workshop, 13-16 Sep 1992, pp 5.7.1 - 5.7.2.

[11] Fontaine,R.;Alary,A.;Mouine,J.;Duval,F.; "A programmable speech processor for profoundly deaf people", Proceeding of the IEEE 17th Annual Conference Engineering in Medicine and Biology Society, 20-23 Sep 1995, Vol 2, pp 1617 – 1618.

[12] Bogli, H., Dillier, N. "Digital Speech Processor For The Nucleus 22-channel Cochlear Implant", Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 31 Oct-3 Nov 1991, pp 1901 – 1902.

[13] Hussnain Ali, *Member, IEEE,* Talha J. Ahmad, Asim Ajaz, and Shoab A. Khan, "Laboratory Prototype of Cochlear Implant: Design and Techniques", Proceedings of the 31st Annual International Conference of the IEEE EMBS, September 2-6, 2009, pp 803-806.

[14] Chen wei-bing ; Zhou ling-hong ; Wang lin-jing ; Xiao zhong-ju, "The Development of Six-Channel Speech Processor for Cochlear Implant Based on DSP", Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008, 16-18 May 2008, pp 1613 – 1616.

[15] User Manual for the SPrint speech processor and accessories, Nucleus Cochlear Implant System, http://professionals.cochlearamericas.com/cochlear-products/nucleus-cochlear-implants/cochlear-implant-support-materials/user-manuals/user-man

[16] P.R.Deman, K. Daemers, M. Yperman, F. E., "Basic Fitting and Evaluation Parameters of a Newly Designed Cochlear Implant Electrode", Acta Otolaryngol 124, 2004, pp 281-285.

The case for Ethernet in Automotive Communications

Lucia Lo Bello
Dipartimento di Ingegneria Elettrica,
Elettronica ed Informatica
University of Catania, Italy
Viale A. Doria, 6- 95125 Catania,
Italy
+39-095-7382386
lucia.lobello@dieei.unict.it

ABSTRACT

Several factors seem to favor the introduction of Ethernet technology in automotive communications. The spreading of Ethernet as an in-vehicle network for today's cars or for those in the near future is being broadly announced by spokespersons for major carmakers and automotive electronics companies. Even in the scientific community there is a growing interest in the topic, shown by the increasing number of studies on the performance of Ethernet-based technologies such as Switched Ethernet or Time-Triggered Ethernet in automotive embedded systems.

This position paper provides an overview on facts and trends towards the introduction of Ethernet in automotive communications and discusses how and to what extent Ethernet technology is likely to step in and provide benefits to the different automotive functional domains. The paper will also discuss which Ethernet technologies would be the possible candidates for the automotive industry.

Categories and Subject Descriptors

C.3.[Special-Purpose and Application-Based Systems] (J.7) [Computers in Other Systems].

General Terms

Performance, Design, Experimentation, Standardization.

Keywords

Automotive communications, in-car networks, Switched Ethernet, Time-Triggered Ethernet, Audio Video Bridging.

1. INTRODUCTION

A growing interest towards Ethernet as an in-vehicle network for cars of today and those in the near future has been recently shown by the industry. Several spokespersons for major carmakers (e.g. BMW, Daimler) and automotive electronics companies (e.g., Bosch, Continental, Micrel, etc.) explicitly addressed the case for Ethernet for automotive applications [1-3],[6],[41], providing examples of the current use of Ethernet in some applications and outlining what's next according to their companies' view. Stimulated by the interest shown by industries and supported by ongoing projects such as the SEIS project [3-4], academic research, often in collaboration with carmakers, is investigating the performance of Ethernet/IP [6-11], Ethernet ABV [12],[31], or Time-Triggered Ethernet [13] in automotive embedded systems.

Several factors seem to favor the introduction of Ethernet technology in the automotive communication systems arena. Some of them are similar to those that, ten years ago, motivated the interest towards the introduction of Ethernet in automation as either a complement or replacement of traditional fieldbuses. The automotive domain is, however, quite different from automation environments. An in-car embedded system is typically divided into several functional domains that feature different requirements and specific constraints [14].

In this context, this position paper discusses how and to what extent Ethernet technology is likely to step in and provide benefits to the different automotive functional domains. The potential for making Ethernet a complement or even replacement to other network technologies in their respective functional domains is also addressed.

The paper considers current Ethernet technologies summarizing significant results from related works. The aim is to assess which Ethernet technology may be suitable for which automotive functional domain.

The paper is organized as follows. Sect.2 elaborates on the motivations for introducing Ethernet in automotive communications. Sect.3 discusses which Ethernet technologies would be the possible candidates for automotive communications, while Sect.4 identifies which automotive functional domains would benefit from using such technologies and provides comparative assessments between Ethernet and the automotive networks currently used in the addressed domains. Finally, Sect.5 concludes the paper giving conclusive remarks and some directions for further investigation into the adoption of Ethernet in cars.

2. MOTIVATIONS FOR ETHERNET IN AUTOMOTIVE COMMUNICATIONS

The aim of this section is to discuss the motivations for using Ethernet as an in-vehicle network. The baseline of this discussion is an overview of significant facts that lean towards the adoption of Ethernet in automotive communications. Such facts are detailed in the following.

Fact 1: Traffic requirements are steadily growing, so there is a need for more bandwidth.

Premium cars today count more than 70 ECUs that implement hundreds of distributed functions to provide, e.g., comfort, safety, infotainment, and which produce many communication exchanges. Several new applications are bandwidth-intensive. For instance, to increase the ease of driving and safety, many

applications, such as lane departure warning systems, signs/traffic lights recognition and collision avoidance systems, require enhanced picture image and sensor resolutions.

The demand for inter-ECU communication has also heavily increased and it is expected to continue growing in the future.

More bandwidth is also required by On-Board Diagnostics (OBD). The amount of software embedded in the cars of today is growing rapidly, due to the constant advancements of functionalities provided by in-car electronic systems. On-Board Diagnostic is needed by many vehicle functions such as, emissions monitoring, diagnosis of components and properties, service and maintenance with the possibility of downloading and updating software.

The time spent on the reprogramming (also called flash update or flashing) of automotive ECUs, which is necessary to upload new applications to the electronic control unit during the manufacturing of the car or at the repair shop, has already become a critical cost factor. Fast diagnostics and shorter update times are strongly required for efficiency and cost reduction.

The United Nations started an action towards the establishment of a legal global standard for the On-Board Diagnostics of cars and trucks and entrusted the International Organization for Standardization (ISO) with the creation of the World Wide Harmonized – OnBoard Diagnostics (WWH-OBD) standard. The aim of the global standard is to replace the regional standards for vehicle diagnosis for emission control [15]. The standard is also known as Diagnostics over Internet Protocol (DoIP) and uses Ethernet as the PHY. The DoIP standard, under specification as ISO 13400 [16], will therefore foster the use of Internet Protocol (IP) for diagnosis and of Ethernet as a replacement for CAN for the reprogramming and diagnostics of automotive Electronic Control Units. This replacement is necessary, as the CAN bus at 500 kbps has become a bottleneck today. Replacing CAN with 100 Mbps Ethernet significantly reduces the time needed to reprogram an ECU [15].

For example, BMW has already been using Ethernet technology to reprogram the calibration software for the engine control modules since early 2008 [1]. In [17] a few figures regarding Vehicle Flashing Times are given that are quite interesting. In the 4th-generation BMW 7 series, to upload 81 MB via CAN 10 hours were required. In the 5th-generation BMW 7 series, to upload 1 GB via Ethernet only 20 minutes were required. The potential savings through faster reprogramming exceeds the costs due to the introduction of Ethernet.

Among bandwidth-demanding applications there are those related to telematics and infotainment that also require support for IP/Web-based applications' need to become more open for non-automotive devices.

Similar considerations also hold for communication between vehicles and the external world, due to applications such as remote monitoring, fleet management, Internet-based automotive applications and Car-to-X communications.

Fact 2: A common network technology would reduce the communication complexity.

Multiple and heterogeneous networks support the different automotive functional domains [14][18]. Table 1 summarizes the different functional domains to be found in a car today and the kind of communication they generate.

Table 1. Functional domains and relevant communications

Functional domain	Communication
Powertrain	Data for the control of engine, transmission, gearbox, etc.
Chassis	Data for the control of car stability and dynamics, i.e., suspension, steering and braking
Body & Comfort	Driving unrelated data concerning the comfort of both driver and the passengers (climate control, windows lifts, seat control, mirrors, doors..)
Driver assistance	Data for driving support operating without user intervention (rear-view, side-view and top-view services, night vision service, speed limit information, lane departure warning, etc.)
Telematics/Infotainment/HMI	Interactive systems presenting data about car operation and driving conditions (navigation systems, route and traffic related information, dashboard, head-up display, etc.)
Entertainment	Driving unrelated data such as audio and video programs, rear seat entertainment, hand-free phones, personal connectivity, etc.

The communications are quite different from one functional domain to another. In some domains, such as powertrain, the main requirement is real-time communication, as the traffic consists of exchanges of small real-time packets. Other domains have more relaxed time constraints, but require more bandwidth. This difference is also reflected in the different network technologies adopted, that span from the low-bandwidth Local Interconnect Network (LIN) [19], used mainly for low-speed communications in the body and comfort domain, to the Controller Area Network (CAN) [20], used in various flavors over different domains (i.e., bandwidth ranging from 100 to 500 Kbps) and to FlexRay [21] that provides 10 Mbps. In the entertainment and infotainment domains as well as in camera-based driver assistance systems the situation is also quite heterogeneous, as here communication is supported by point-to-point Low-Voltage Differential Signaling (LVDS) wires or by analogue Color Video Blanking Signal (CVBS) cables or, more recently, by the Media Oriented Systems Transport (MOST) protocol [22] with data rates of 25 Mbps, 50 Mbps and 150 Mbps. Traffic shaping mechanisms for IP-based in-car switched Ethernet networks, implemented in the switching device [23] or in the video source [24], were investigated for camera-based driver assistance services.

The communications may be quite different even within the same functional domain when several functions are present that feature different data rates and constraints. As a result, several network types are also found within the same domain, with different characteristics in terms of bandwidth, real-time support, etc. Moreover, when a new network technology providing, for example, a higher bit rate is introduced, it is usually intended for supporting novel applications. As a result, it usually does not

replace, but complement the already existing networks. The reason for this is that in the automotive environment as long as a (sub)system works properly, there is no willingness to change it, for the sake of keeping the costs low.

This network heterogeneity complicates the communication exchanges, so the usage of a single network technology, where applicable, would be beneficial to avoid the need for gateways. Moreover, it has to be considered that many new applications also require communications between functions belonging to different domains. For instance, in new hybrid vehicles, intelligent battery management systems located in the powertrain domain will optimize charging and discharging strategies based on navigation data from the telematic domain.

In-car interdomain communications between multiple not directly compatible networking technologies requires the support of complex gateways. The gateway functionality may be either centralized in one ECU to which every bus is connected, or distributed over several ECUs, each one acting as "sub gateway". These are complex gateways that require application knowledge. For example, as explained in [4], if a device connected to MOST has to send a request to configure a parameter to a device connected to FlexRay, the communication requires multiple steps that involve a mediating gateway. To correctly perform these steps, such a gateway requires knowledge about the application. This means that every time a change in the application is made, the gateway has to be adapted. This is not a desirable property. A common networking technology for in-car control units would solve the problem and simplify the electronic architecture of the vehicle.

2.1 The Ethernet suitability for Automotive Communications

Ethernet is a promising candidate for in-car communications. The main motivation for this is the higher bandwidth provided by Ethernet (100 Mbps onwards) as compared to current in-car networks. Such an increased bandwidth paves the way for applications, like Advanced Driver Assistance Systems (ADASs), which make the volume of exchanged data in automotive communication continuously grow.

Another enabling factor for using Ethernet as a common networking technology for in-car communications is the assessed technology, which entails that there is a large knowledge already available that allows for better testing, maintenance and development. Thanks to the Ethernet's wide use, standardization and openness, a large availability of high-quality chips on the market and therefore low-cost product development and manufacturing can be expected/predicted. On the contrary, the main competitors of Ethernet, MOST and FlexRay, are only used in the automotive domain, and this entails a smaller market penetration and thus higher costs for products based on these technologies [25].

In addition to the above mentioned features, Ethernet technology is scalable, thus meeting the scalability requirement imposed by today's automotive systems, where the number of nodes to interconnect steadily increases.

Another strong point in favor of Ethernet is the support offered to the IP stack. With Internet connectivity, the IP protocol will be used in-car, opening the way to enhanced navigation functionalities, remote diagnostics and location-based services. Investigations into usage of the Internet Protocol (IP) and the Ethernet in automotives is in progress in academia, the car industry and companies producing automotive electronic devices (BMW [1], Bosch [2], Continental [3], Daimler [6] to mention just a few). Providing the basis for IP as a common networking technology for control units in the car to reduce the complexity of the car electronic architecture is the aim of the SEIS project [4][5], a German project coordinated by BMW Forschung und Technik GmbH in Munich. The SIES project investigates IP-based communication both inside the car and between the car and the environment. The target is not to replace all the technologies currently in use, but to keep using them on an IP-based network and to resort to alternative technologies, selected among those already well established in other industrial contexts, when needed. Among alternative technologies, Ethernet and its real-time variants are addressed and suitable physical layers and network topologies for the automotive domain are investigated. Attention is paid to the IEEE 802.1 AVB protocol and its extensions [26-29] to transfer multimedia data in real-time over IP.

In [6] Daimler's view on the usage for Ethernet in automation is summarized. Basically, the company sees Ethernet as a good solution to have higher bandwidth for future automotive applications at reasonable costs, through a well-proven and widely used technology that has been tested in other domains (such as industrial automation, avionics and telecommunications). Another advantage they mention is the support to IP, which is valuable, as IP is a natural candidate for applications like vehicle diagnostics, in-car internet access, smart charging in electric cars, etc.

Also BMW is really towards the use of Ethernet in cars. In [1] a roadmap is drawn, where unshielded Ethernet was introduced as a diagnostic interface in 2008, while shielded Ethernet was introduced for Rear Seat Entertainment in 2008 and for the Park Assist Camera for the new X5 (pilot expected in 2013).

2.1.1 Ethernet Electromagnetic Compatibility issues

Automotive networks typically operate under high temperature and high electromagnetic radiations, so the question is whether Ethernet can correctly operate in these conditions. Temperature is not an issue for Ethernet, which performs well under high temperatures, thanks to the low power consumption. As far as Electromagnetic Compatibility (EMC) issues are concerned, two aspects have to be considered. The first is the Electro-Static Discharge (ESD), which consists of the unwanted short-duration electric current that flows when two charged objects come into close proximity or even in contact and may cause damage to electronic equipment. Many new Ethernet devices have improved their Electro-Static Discharge performance thus meeting the limits. The second aspect refers to Electro-Magnetic Interference (EMI) i.e., the unwanted effects of unintentional generation, propagation and reception of electromagnetic energy. While Standard Ethernet 100 Base TX unshielded exceeds the typical limit for EMC emission, Standard Ethernet 100 Base TX shielded technology does not exceed the limit, but unfortunately requires expensive cable and connectors. Plastic Optical Fiber can be used for 100 Mbps Fast Ethernet with a reach of 100 m, but it is a costly solution and cost is an issue in the automotive field. The most effective solution, as reported in [1] and [30], is Ethernet Unshielded Twisted Single Pair (UTSP) at 100 Mbps,

that successfully passes the EMC immunity test. The UTSP is unshielded, provides for one pair of wires at 100 Mbps with full duplex operation and requires standard inexpensive cables and connectors. Automotive qualified Ethernet devices currently available on the market, e.g., from Micrel (transceiver, switches) and from Broadcom® (transceiver), are designed to meet automotive EMC specifications.

As the cable lengths in cars is limited and never reaches 100 m (this is the maximum length specified by the IEEE 802.3), assuming a lower maximum cable length, for instance 10 meters, the Ethernet transmitter drive strength can be reduced accordingly, thus also reducing the output signal amplitude and therefore emissions.

3. OVERVIEW OF AUTOMOTIVE-RELEVANT ETHERNET TECHNOLOGIES

Recent Ethernet technologies provide support for real-time behavior and QoS. Real-Time support is provided by Industrial Ethernet protocols in IEC 61784 [32][33], that improve real-time capabilities of Ethernet-based networks for industrial scenarios. However, IEC 61784 technologies are not as widely-used in the mass market as standard Ethernet and are too costly for the automotive domain. The main challenge for the automotive industry will instead be to transfer and extend standard IP and Ethernet into cars and still fulfill the automotive requirements. Moreover, no intention to use any Industrial Ethernet for the automotive case is in place. PROFINET, for instance, addresses trains (train profile for PROFINET IO), not cars.

QoS support is offered by the IEEE 802.1 Audio/Video Bridging (AVB) standard [26-29], that provides for highly reliable audio and video applications over IEEE 802 networks.

Another promising candidate is the TTEthernet technology (TTEthernet) [34], which is marketed by TTTech Computertechnik AG for use in avionics, in aerospace applications and in other real-time domains. TTEthernet enables deterministic time-triggered communications, rate-constrained and event-triggered communications over the same network interface [42]. The technology is compatible with legacy Ethernet (ARINC 664 Part 7 Standard, 802.3) [35]. Integration with AVB would also be possible [36].

In the following we will examine the features of both the IEEE AVB standard and TTEthernet and discuss their possible application domains for automotive usage.

3.1 The IEEE Audio Video Bridging standard

AVB is a common name for the set of technical standards defined by the IEEE 802.1 Audio/Video Bridging Task Group. The AVB standards provide the specifications for time-synchronized low latency streaming services through IEEE 802 networks.

AVB includes three specifications:

- IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications (gPTP) [29].
- IEEE 802.1Qat: Stream Reservation Protocol (SRP) [28].
- IEEE 802.1Qav: Forwarding and Queuing Enhancements for Time-Sensitive Streams (FQTSS) [27].

The IEEE 802.1as Time Synchronization provides precise time synchronization of the network nodes to a reference time. It synchronizes distributed local clocks with a reference that has an accuracy of better than 1 us. The IEEE 802.1Qat Stream Reservation allows for the reservation of resources within switches (buffers, queues) along the path between sender and receiver. The IEEE 802.1Qav Queuing and Forwarding for AV Bridges separates time-critical and non time-critical traffic into different traffic classes extending methods described in the IEEE 802.1Q standard and performs traffic shaping at the output ports of switches and end nodes to prevent traffic bursts.

For seven hops within the network, the AVB standard guarantees a fixed upper bound for latency. Two QoS classes are defined, i.e.

- Class A, that provides a maximum latency of 2ms
- Class B, that provides a maximum latency of 50ms.

With a careful planning of periodic execution and mapping to the high priority queues within switches, AVB is able to guarantee low jitter. As the resource reservation protocol is able to dynamically handle QoS, new devices can join the network at any time and QoS can be maintained through a combined design approach, where a QoS configuration made at the end of the production line is adapted on the field afterwards.

The work in [31] addresses the accuracy of the time synchronization mechanism of Ethernet AVB under varying temperature conditions. The measurement results provided are encouraging.

3.2 The Time-Triggered Ethernet

TTEthernet [34][36][42] combines the determinism, fault-tolerance properties and real-time behavior of the time-triggered technology with the flexibility, dynamics and legacy of "best effort" Ethernet. TTEthernet offers several advantages. First, it offers higher bandwidth compared to FlexRay or CAN (100 Mbps onwards). Second, it supports communication among applications with diverse real-time and safety requirements. Finally, TTEthernet provides three different traffic types: time-triggered (TT) traffic, rate-constrained (RC) traffic and best-effort (BE) traffic.

Time-triggered (TT) messages are transmitted at predefined times and have precedence over the other kinds of traffic. They are suitable for brake-by-wire, steer-by-wire systems (avionics). Rate-constrained (RC) messages do not follow a sync time base, so multiple transmissions may occur at the same time and messages may queue up in the network switches, leading to increased transmission jitter. RC messages are sent at a bounded transmission rate that is enforced in the network switches, so that for each application a max predefined bandwidth, together with delays and temporal deviations within given limits, are guaranteed. Rate constrained messages are suitable for multimedia or safety-critical automotive and aerospace applications that need highly reliable communication, but do not feature strict temporal constraints. Best-effort (BE) messages use the remaining bandwidth and have less priority than TT and RC messages. BE messages have no guarantee on whether and when they can be transmitted, i.e., on the delay and on the delivery at the destination. These messages are suitable for all legacy Ethernet traffic (e.g. Internet protocols) without any QoS requirement.

TTEthernet is used as the backbone system in the NASA Orion spacecraft, the successor of the Space Shuttle [37]. TTEthernet is also a SAE standard (AS6802).

More details on TTEthernet and its properties can be found in [34].

4. WHICH ETHERNET TECHNOLOGY FOR WHICH DOMAIN

The question we attempt to answer here is which Ethernet technology is suitable for the automotive environment. The choice really depends on the functional domain. In the following, we will consider possible application scenarios for Ethernet AVB and TTEthernet.

4.1 Application scenarios for AVB in cars

The IEEE AVB standard [26-29] has recently attracted attention as a potential in-vehicle network technology for multimedia, infotainment and driver assistance. There are multiple reasons for this interest in AVB, such as the enhanced QoS provided, the IEEE standardization, no need for license fees and, last but not least, prices and quality comparable to those of standard Ethernet.

In-car audio/video systems, in addition to common playback functions, also handle content that requires a timely delivery. For instance, turn-by-turn navigators continuously present directions to the users in graphics or speech and dynamically update the path to the destination taking into account traffic and road conditions. Moreover, these systems give the user step-by-step vocal advices about the street name, the distance to the turn and whether to turn left or right.

The Audio/Video content associated with infotainment systems is steadily increasing, but the usage of point-to-point dedicated connections for audio and video content, such as the currently adopted shielded LVDS cables, has to be discontinued for many reasons. Among them, wiring complexity, that also affects maintenance, reliability and weight, and costs, in terms of wires, connectors and fuel consumption.

In-vehicle infotainment networking is today dominated by MOST technology [22]. However, according to the Nov.2008 Hansen Report [40], MOST is not "open enough" compared with CAN, LIN and FlexRay. For this reason, in several years alternatives, including Ethernet with AVB extensions, might step in [39].

4.1.1 Comparative assessments between MOST and AVB

MOST was originally designed for automotive infotainment. It therefore exploits the available bandwidth optimally for all kinds of media streaming. Despite MOST outperforming Ethernet in payload efficiency (PE=Payload data/Sent data) [12], with MOST the total network bandwidth is shared among all connected devices, while Ethernet AVB is a switched network that multiplies the available bandwidth [12]. As AVB utilizes bandwidth only between source and destination node connections, there is a significant bandwidth saving that allows a higher throughput over an AVB network compared with a MOST network, even when they operate at equivalent bit rates [39].

Some works see a potential for AVB for time-triggered-like communications, thanks to the support provided in terms of time synchronization and synchronous data transport [25]. However further work has to be done in this direction to improve performance.

4.2 Application scenarios for Time-Triggered Ethernet in cars

Chassis and powertrain functions operate mainly as closed-loop control systems and their implementation is moving towards a time-triggered (TT) communication, as this approach is able to provide a deterministic communication service. TTEthernet would be a good candidate for these subsystems, especially for the chassis domain, whose behavior has a strong impact on the vehicle's stability, agility and dynamics, and so is very critical from a safety standpoint.

Deterministic communication overcomes the problem of interdependencies between components, which is a major issue and cost factor in today's automotive distributed systems. A deterministic communication system significantly reduces the integration and test effort, as it guarantees that the cross-influence is completely under the control of the application and is not introduced by the communication system. Moreover, determinism facilitates the system composability (i.e., ability to integrate individually developed components) and real-time behavior.

As stated in [36], the introduction of TTEthernet as in-car network would allow to reduce the number of end systems and to integrate several distributed functions on a small number of ECUs. This is because TTEthernet meets the requirements that this approach imposes, i.e. that bandwidth be apportioned exactly and deterministically without statistical fluctuation (jitter) of the network traffic, and that bit rates be guaranteed.

Other possible scenarios for TTEthernet in automotive applications are:

- Advanced Driver Assistance Systems (ADAS), thanks to the combination of high bandwidth and TT communication.

- Multimedia, thanks to the reliable communication and guaranteed data rates for audio and video. Moreover, by using TTEthernet for both ADAS and infotainment, driver assistance systems and infotainment could be integrated into the same network.

- X-By-Wire, thanks to its real-time, fault-tolerant and fail-operational behavior that meets the communication requirements typical of these systems.

4.2.1 Comparative assessments between FlexRay and TTEthernet

The paper [13] offers a competitive analysis of FlexRay and TTEthernet. Based on a mathematical model, the analysis provides comparative results for real-time relevant metrics like latency, jitter and bandwidth. The general eligibility of TTEthernet for in-vehicle applications is shown in a scenario where a fully utilised FlexRay system is replaced by a time-triggered Ethernet. The paper also discusses the utilization benefits offered by a switched system, like TTEthernet, when using group communication, while FlexRay is limited to broadcast communications. The analysis in [13] shows that FlexRay real-time traffic can be supported by TTEthernet. Jitter and latency are comparable. The sample configuration used is

shown in Table 2, while the results obtained are given in Table 3 (redrawn from [13]).

Table 2. Sample configuration (source [13])

	FlexRay	TTEthernet
Bus speed	10 Mbps	100 Mbps
Max payload size	254 bytes	1500 bytes
Min payload size	1 byte	46 bytes
Topology	Two active stars/switches	
Wire length	72 m	
Divergence of quartz	220 ppm	
Cycle time	16 ms	

Table 3. Jitter and latency results (source [13])

	FlexRay	TTEthernet
Latency min payload	12.2 us	24 us
Latency max payload	265.2 us	372 us
Jitter bounds	6.4 us	<10 us

FlexRay and TTEthernet share very good properties for automotive communications. Both of them provide support for fully deterministic data communication for time-critical applications and have built-in fault-tolerance and safety mechanisms. While FlexRay is qualified for automotive, TTEthernet does not have this property yet. FlexRay controllers have the ISO 26262 (ASIL-D) certification, not yet available for TTEthernet.

TTEthernet provides higher bandwidth than FlexRay (100 Mbit/s vs. 10 Mbit/s). Software stacks as well as development and configuration tools are available for both technologies. The two technologies currently differ as far as costs are concerned, being TTEthernet more expensive due to the lack of a mass production of TTEthernet products till now. This difference may be overcome with increasing market penetration.

Both FlexRay and TTEthernet support a number of network nodes and network topologies that are adequate for the needs of automotive applications. In particular, FlexRay supports bus, star and mixed topologies. TTEthernet supports star and star bus, while bus and ring are possible with special switch components. FlexRay is a de facto standard in automotive and TTEthernet is a SAE standard (SAE AS6802). While the EMC of FlexRay has been proven in automotive, TTEthernet EMC for automotive is currently under test.

4.3 Ethernet in automotive communications: evolution or revolution?

Ethernet was recently introduced in several car models as cost-efficient high-speed data access for diagnostics, software updates and multimedia for entertainment (Rear Seat Entertainment Systems). As reported in [1], the pilot applications for Ethernet according to the BMW view are Driving Assistance Functions. The BMW plan is to use Ethernet instead of shielded Low-Voltage Differential Signaling (LDVS) video transmission for the surround view system to provide good overview during maneuvering. The company target is to accomplish this by 2013 [1] and significant cost reduction is expected.

Other targeted applications are night vision with person recognition, speed limit information, entertainment video transmission (TV, DVD, etc.).

The introduction of Ethernet in cars looks more an evolution than a revolution: In the next few years, Ethernet will not replace existing automotive networks completely, but a "migration path" to facilitate the communication between the existing automotive networks and Ethernet is needed [25]. This is the motivation behind some effort to investigate how to realize gateways between AVB networks and the automotive networks currently in use, such that the QoS offered can be maintained across the network borders. For instance, the work [25] proposes both a MOST/AVB gateway and a FlexRay/AVB gateway to support synchronous data transport, and uses an evaluation system built within the SEIS project [3] activities to validate the proposed gateway concepts.

The paper [43] addresses a migration concept for transferring CAN traffic over Ethernet/IP. The idea is to use full duplex switched Ethernet connections for multiplexing data originating from CAN ECUs and streaming data coming from co-located high bandwidth sensors over one Ethernet connection, for the sake of reducing packaging, weight and system costs.

5. CONCLUSIONS

Ethernet usage in cars is expected to spread in several domains. The first one is diagnostics, where Ethernet is already in use and will replace the bottleneck CAN. Ethernet usage will further grow thanks to the DoIP standard that allows for seamlessly interfacing the car to a service centre network or remote laptop.

Another success story of Ethernet in cars is expected in the multimedia and infotainment domains. In today's cars Ethernet already connects the Rear Seat Entertainment system to the Head Unit, thus providing high speed access to the mass storage located in the head unit. The AVB standard will compete with MOST which, in turn, is expected to fast displace the LDVS data transfer technology.

Advanced Driver Assistance Systems involving cameras represent another use-case for Ethernet, as these applications require high bandwidth to support high speed data communication and FlexRay is not suitable for this. ADAS may therefore be the right use-case for Ethernet, especially for TTEthernet.

In terms of timeline, Ethernet is expected to enter the non-safety critical domain first. While Ethernet definitely has a bright future for video, audio and infotainment, its usage in automotive domains with hard real-time constraints depends on some critical factors. For real-time control, the car industry is (slowly) moving from CAN to FlexRay, so it could take time for Ethernet to step in. FlexRay is used for some time-critical and safety-critical applications and will likely continue to be used in the powertrain and vehicle dynamics management domain. An aspect to consider is that there are not so many Ethernet COTS components suited to cars, due to EMC issues. Currently, Broadcom is launching the Broadcom BroadR-Reach™ PHYs family of transceivers, while

Micrel offers automotive qualified Ethernet devices (PHY transceivers, integrated MAC/PHY controllers, switches) that reduce the drive strength via internal software registers or via a modification in hardware. Assuming that there are some proprietary technologies, the question is whether the automotive industry will rely on a product/standard that is proprietary.

An open question regards the topology to be used in cars. One could think about a single Ethernet backbone supporting all the kinds of traffic (safety-critical, multimedia), but the study in [8] showed that different traffic classes over a switched Ethernet in-car network influence each other, causing the violation of critical data constraints. The work in [8] showed that a Switched Ethernet with a star-based topology may support different traffic types while providing satisfying QoS only if the network is managed in such a way that overload never occurs and a prioritization mechanism is used.

One possible option foresees an infotainment architecture centered around an AVB Ethernet backbone conveying all the traffic, i.e. in-vehicle data and control traffic, together with audio/video streaming for passenger entertainment, driver assistance, mobile interconnect connectivity. However, for the sake of integrity, reliability and safety, it is probably better to keep safety-critical communications separate from infotainment ones. Some companies, like Micrel [41] envisage a single standard Ethernet for multimedia applications and non-critical data traffic and an AVB cloud (i.e., a kind of sub-network where all devices must support AVB capabilities) for time-critical traffic. As reported in [41], it was suggested that only the so-called Audio Video Bridging for Automotive (AVA) subset of the AVB specification, including the IEEE 1722 AVB packet and the PTPv2 Time Synchronization (IEEE 802.1as) would be required for the automotive needs.

The separation between different traffic types is definitely possible with TTEthernet, that provides a native support for deterministic communication while also allowing for rate-constrained and best-effort exchanges.

As far as new visions and new possibilities for Ethernet in the automotive field are concerned, according to the Bosch and BMW view, in a future architecture based on the deployment of Domain Control Units (DCUs) and a backbone connecting those domains, Ethernet will be the ultimate choice (expected start 2020).

Another possible scenario for Ethernet is relevant to electric cars. The next generation of electrical vehicles represents a unique opportunity for a significant rethinking of current automotive network architectures. A shift from proprietary solutions to a novel network architecture, based on an established standard technology, would allow for faster design and analysis of the network transmission schedule, better quality and performance assessments. The adoption of a common communication network architecture would simplify the task of ECU suppliers allowing for component reusability across different car manufacturers shortening the time to market of their products.

6. ACKNOWLEDGMENTS

This work has been partially funded by the European Commission in the framework of the ARTISTDesign NoE under the grant number 214373. The Author is grateful to Michael Short and Luis Almeida for their kind support and to Françoise Simonot-Lion, Nicolas Navet, Jörn Migge, Jean-Luc Scharbarg, Marco Di Natale, Daniel Herrscher and Andreas Kern for the interesting discussions.

7. REFERENCES

[1] Bruckmeier, R. 2010. Ethernet for Automotive Applications. Freescale Technology Forum (Orlando, FL, US, June 23, 2010).http://www.freescale.com/files/ftf_2010/Americas/WBNR_FTF10_AUT_F0558.pdf.

[2] Hammerschmidt, C. 2011. Ethernet to gain ground in automotive applications, Bosch predicts. *EETimes Europe*. (4 Feb.2011). http://electronics-eetimes.com/en/ethernet-to-gain-ground-in-automotive-applications-bosch-predicts.html?cmp_id=7&news_id=222905757

[3] Noebauer, J. 2011. *Is Ethernet the rising star for in-vehicle networks?* Invited Talk at the 16th IEEE Conference on Emerging Technologies in Factory Automation (Toulouse, France, Sept.7, 2011). ETFA 2011.

[4] SEIS - Security in Embedded IP-based Systems. www.strategiekreis-elektromobilitaet.de/projekte/seis/.

[5] Glass, M., Herrscher D., Meier H., Piastowski M., and Schoo P., 2010. SEIS – Security In Embedded IP-Based Systems. *ATZ Elektronik* 5, (01-2010), 36-40.

[6] Kern, A., Zhang, H. and Teich, J. 2011. Testing switched Ethernet networks in automotive embedded systems. In *Proceedings of the 6th IEEE International Symposium on Industrial Embedded Systems* (Vasteras, Sweden, 15-17 June, 2011) SIES 2011. IEEE Industrial Electronics Society, Piscataway, NJ, 150-155. DOI=10.1109/SIES.2011.5953657

[7] Lim, H.T., Volker L., Herrscher, D. 2011. Challenges in a future IP/Ethernet-based in-car network for real-time applications. In *Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference, 2011.* (San Diego, California, June 5-10, 2011). DAC 2011. IEEE, New York, NY, 7-12.

[8] Lim, H.-T., Weckemann, K., Herrscher, D. Performance Study of an In-Car Switched Ethernet Network without Prioritization. In: *Communication Technologies for Vehicles (Lecture Notes in Computer Science 2011)*, Springer, Heidelberg, 2011.

[9] Lim, H.T., Krebs, B., Völker, L., and Zahrer, P. 2011. Performance Evaluation of the Inter-Domain Communication in a Switched Ethernet Based In-Car Network. In *Proceedings of the 36th IEEE Conference on Local Computer Networks* (Bonn, Germany, Oct. 04-07, 2011). LCN'11. IEEE, Piscataway, NJ.

[10] Rahmani, M., Steffen, R., Tappayuthpijarn, K., Giordano, G., Bogenberger, R., and E. Steinbach.2008. Performance Analysis of Different Network Topologies for In-Vehicle Audio and Video Communication. *In Proceedings of the 4th International Telecommunication Networking WorkShop on QoS in Multiservice IP Networks*. (Venice, Italy, Feb.13-15, 2008). IT-NEWS 2008. IEEE; Piscataway, NJ, 179-184. DOI=10.1109/ITNEWS.2008.4488150.

[11] Hintermaier W., Steinbach, E. 2010. A System Architecture for IP-camera based Driver Assistance Applications. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), (San Diego, CA, June 21-24, 2010). IEEE, Piscataway, NJ, 540-547. DOI= 10.1109/IVS.2010.5548103.

[12] Noebauer J,. Zinner H. 2011. Payload efficiency and network considerations of MOST and Ethernet. *Elektronik automotive. Special issue MOST*. March 2011, 14-17. http://www.elektroniknet.de/automotive/most-spezial/technik-know-how/article/77135/2/Friend_or_Foe/.

[13] Steinbach, T., Korf, F., and Schmidt, T. C. 2010. Comparing Time-Triggered Ethernet with FlexRay: An Evaluation of Competing Approaches to Real-time for In-Vehicle Networks. In *Proceedings of the 8th IEEE Intern. Workshop on Factory Communication Systems.* (Nancy, France, May 18-21, 2010). WFCS'10. IEEE Industrial Electronics Society, Piscataway, NJ.

[14] Navet, N., Simonot-Lion, F. 2009. Trends in Automotive Communication Systems. *Embedded Systems Handbook, 2nd Edition, Part three. Dr. R. Zurawski, Ed.* CRC Press, Taylor and Francis Group, Boca Raton, FL. 13-1, 13-24.

[15] Lang, J. 2010. ISO 13400 – Diagnostics over Internet Protocol (DoIP) at EB, *EB Automotive Software Newsletter,* 2010. http://www.elektrobit.com/

[16] ISO 13400-1:2011 - Road vehicles -Diagnostic communication over Internet Protocol (DoIP) -Part 1: General information and use case definition. Published on Oct.18, 2011.

[17] Thomsen, T., Drenkhahn, G. 2008. Ethernet for AUTOSAR. *1st AUTOSAR Open Conference & 8th Premium Member Conference* (Detroit, MI, Oct. 23, 2008). http://www.autosar.org/download/conferencedocs/07_Elektrobit_Ethernet_for_Autosar.pdf

[18] Nolte, T., Hansson, H., Lo Bello, L. 2005. Automotive Communications - Past, Current and Future. In *Proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation* (Catania, Italy, Sept. 19-22, 2005). ETFA '05. IEEE Industrial Electronics Society, Piscataway, NJ, 1, 985-992. DOI=10.1109/ETFA.2005.1612631.

[19] LIN Consortium, *LIN Specification Package Revision 2.0*, Motorola GmbH, Schatzbogen 7, 81829 Munchen, Germany, Sept. 2003.

[20] *CAN Specification Version 2.0*, Robert Bosch GmbH, Postfach 50, 70049 Stuttgart, Germany, Sept. 1991.

[21] FlexRay Consortium, *FlexRay Communications System Protocol Specification Version 2.1 Revision A*, Altran GmbH & Co. KG, Schillerstr.20, 60313 Frankfurt am Main, Germany, Dec. 2005.

[22] MOST Media Oriented System Transport, Multimedia and Control Networking Technology, Rev. 2.4, OASIS Silicon Systems, 2005. http://www.mostcooperation.com/publications/Specifications_Organizational_Procedures/index.html?dir=291

[23] Rahmani, M. Tappayuthpijarn, K., Krebs, B., Steinbach, E., and Bogenberger, R. 2009. Traffic Shaping for Resource-Efficient In-Vehicle Communication. *IEEE Transactions on Industrial Informatics*, 5, 4, 414-428, 2009.

[24] Hintermaier W., Steinbach, E. 2010. A System Architecture for IP-camera based Driver Assistance Applications. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, (San Diego, CA, June 21-24, 2010). IEEE, Piscataway, NJ, 540-547. DOI= 10.1109/IVS.2010.5548103.

[25] Zinner, H., Noebauer, J., Gallner, T., Seitz, J. and Waas, T., 2011. Application and realization of gateways between conventional automotive and IP/Ethernet based networks. In *Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference* (San Diego, CA, Jun 05-09, 2011). DAC'11. ACM, New York, NY, 1-6.

[26] IEEE, Inc. Audio Video Bridging Task Group. IEEE Standard 802.1 edition. October 2010.

[27] IEEE, Inc. Forwarding and Queuing Enhancements for Time-Sensitive Streams. IEEE802.1Qav/D7.0 edition. October 2009.

[28] IEEE, Inc. Stream Reservation Protocol. IEEE 802.1Qat/D6.1 edition. June 2010.

[29] IEEE, Inc. Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. IEEEP802.1AS/D7.5 edition. October 2010.

[30] Jones, M. 2009. *EMI Challenge to Ethernet in the Car. White paper.* Micrel Inc., San Jose, CA. Feb. 2009, p.1-6. http://www.micrel.com/applications/auto/automotive_ethernet_emi.pdf

[31] Kern, A., Zinner, H., Streichert, Noebauer, J. and Teich, J. 2011. Accuracy of ethernet AVB time synchronization under varying temperature conditions for automotive networks. In *Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference.* (San Diego, California, June 5-10, 2011). DAC 2011. IEEE, New York, NY, 597-602. DOI=10.1145/2024724.2024862

[32] International Electrotechnical Commission, IEC 61784-1, Digital data communications for measurement and control - Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems, 2008. http://www.iec.ch.

[33] International Electrotechnical Commission, IEC 61784-2, Industrial communication networks- Profiles, Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3, 2010, http://www.iec.ch.

[34] TTA-Group: TTEthernet-Spezifikation. 2008. http://www.ttagroup.org/ttethernet/specification.htm

[35] ARINC: Specification 664P7-1. 664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network. 2009. https://www.arinc.com/cf/store/catalog_detail.cfm?item_id=1269

[36] TTTech Computertechnik AG. 2011. TTEthernet in Motion. TTTech White paper.2011. http://www.automationworld.com/whitepapers/TTTech-TTEthernet_in_Motion.pdf

[37] Walko, J. TTTech, NASA Team on TTEthernet for Space Apps. *EETimes,* Apr.14, 2009. http://www.eetimes.com/electronicsnews/4195051/TTTech-NASA-team-on-TTEthernet-for-space-apps.

[38] SAE: AS6802 - SAE Standards. 2010. http://www.sae.org/servlets/works/documentHome.do?comtID=TEAAS2D&docID=AS6802&inputPage=wIpSdOcDeTaIlS

[39] Kreifeldt, R.. 2009. AVB for automotive use. *AVnu Alliance White paper*. http://www.avnu.org.

[40] The Hansen Report on Automotive Electronics. Nov. 2010.

[41] Jones, M. *Automotive Ethernet Diagnostics and Beyond!. White paper*, Micrel Inc., San Jose, CA. Dec. 2010, p.1-10.

[42] Kopetz, H., Ademaj, A., Grillinger, P. and Steinhammer. 2005. The Time-Triggered Ethernet (TTE) Design. In *Proc. of the 8th International Symposium on Object-Oriented Real-Time Distributed Computing* (Seattle, WA, May 18-20, 2005). ISORC'05. IEEE, Piscataway, NJ, 22–33. DOI=http://doi.ieeecomputersociety.org/10.1109/ISORC.2005.56

[43] Kern, A., Streichert, T., Teich, J. 2011. An automated data structure migration concept - From CAN to Ethernet/IP in automotive embedded systems (CANoverIP). In *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition* (Grenoble, France, March 14-18, 2011). DATE 2011. IEEE, Piscataway, NJ. 1-6. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5763027&isnumber=576299

On the Use of Code Mobility Mechanisms in Real-Time Systems

Luis Lino Ferreira
CISTER Research Centre / School of Engineering of the Polytechnic Institute of Porto
Porto, Portugal
llf@isep.ipp.pt

Luís Nogueira
CISTER Research Centre / School of Engineering of the Polytechnic Institute of Porto
Porto, Portugal
lmn@isep.ipp.pt

ABSTRACT

Applications with soft real-time requirements can benefit from code mobility mechanisms, as long as those mechanisms support the timing and Quality of Service requirements of applications. In this paper, a generic model for code mobility mechanisms is presented. The proposed model gives system designers the necessary tools to perform a statistical timing analysis on the execution of the mobility mechanisms that can be used to determine the impact of code mobility in distributed real-time applications.

Keywords

Real-time systems, distributed embedded systems, mobile systems, code mobility, quality of service.

1. INTRODUCTION

Real-time systems are increasingly shifting from a set of small, local applications to powerful, resource-hungry, open distributed applications [4]. By the very nature of open real-time systems, the availability of resources is unknown beforehand and service provisioning can only be determined dynamically as new applications arrive to the system. Consequently, there is an increasing demand for supporting distributed applications with the flexibility to offload parts of their computations to neighbour or "in the cloud" nodes due to local resource scarcity, while ensuring the real-time behaviour of these applications, both during execution and during reconfiguration, after mobility of code has occurred.

Therefore, open real-time distributed systems must provide applications the support to: i) use services provided by remote components; ii) move part(s) of the application's code to remote nodes; and ii) guarantee real-time behaviour. The first requirement can be supported by a service-based infrastructure [4], to easily and transparently interconnect local and remote parts of an application. The second requirement can be supported by code mobility frameworks, allowing the installation and execution of parts of an application in remote nodes [9]. Finally, a real-time resource manager can support the third requirement. A well-established solution is to use capacity reserves. This has been proved to be successful in improving the response times of soft real-time tasks while preserving all hard real-time constraints, both for CPU [3] and network [2] scheduling.

1.1 Related work

Although not widely studied, a few solutions have already been proposed to analyse the impact of code mobility on the real-time requirements of applications.

In [11], the authors propose and experimentally characterise the behaviour of a hard real-time framework that supports the migration of tasks between nodes. However, the work does not propose a mathematical model that enables system designers to account for the impact of the mobility protocol on the overall timing behaviour of applications.

A strategy for minimising the impact of code mobility in a hierarchical preemptive fixed priority scheduling system for Real-Time Java is proposed in [10]. The authors mainly determine the points in time at which the migration process should be started, which guarantees that the deadlines of tasks are met and that the migration process is executed between consecutive evocations of a migratable task.

Statefull services require the transfer of state, whose duration depends on the length of the data being transferred. However, during this period of time no transactions can be executed on that service (known as blackout time). However, such determination is only possible in systems with a well-known and controlled timing behaviour. Therefore, in [12], the authors tackled the problem of minimising the blackout time by proposing a partial blocking and a non-blocking approach for state transfer, which are capable of providing real-time guarantees.

Nevertheless, none of these works focus on the mobility mechanism itself. Such mobility mechanisms should be supported by a mobility framework that enables the runtime relocation of services in response to reconfiguration/update events (e.g., the system might reconfigure itself due to the disappearance of a node involved in a computation).

As an example, consider running a video game on a mobile device that has offloaded parts of its computations to neighbour devices. Reconfiguration in such a distributed cooperative execution might be required if one of the nodes, currently running one of game's components, is no longer capable of outputting the required QoS. In such case, the component can be migrated to another node able to supply the required QoS. Ideally, such change should be executed seamlessly, i.e. the game delays should (preferably) be unnoticeable.

Examples of works that tackle the specific problem of selecting the new distributed configuration are [4] and [1]. The former allows the determination of a distributed configuration that maximises the satisfaction of the user's QoS preferences among a set of allowed QoS levels. The latter tries to fulfil the same goals, but each service is only allowed to specify a single QoS level. However, a mechanism to determine the impact of code mobility in distributed real-time applications is still missing.

1.2 Contributions and paper structure

Service mobility in a distributed execution environment is a complex operation that evolves through several phases, including

sending the code and state to the destination node and rebinding connections between components. Additionally, resources must be explicitly reserved on the destination node, prior to the start of the mobility process. Due to its complexity, we propose that a Mobility Management framework (represented in Figure 1 by Mx) should control the mobility of code between nodes of a distributed system. This paper focuses on the model and timing analysis for a generic code mobility mechanism for distributed soft real-time applications. The proposed model is generic enough, helping the system designer to define the most appropriate parameters for the mobility management modules and to determine the feasibility of the timing constrains imposed on applications, including mobility and reconfiguration events.

The remainder of the paper is organised as follows. Section II defines the generic model for the distributed applications and for the mobility mechanism. Section III discusses and analyses the code mobility phases and their timings. The main consequence of the mobility mechanism is the introduction of a bounded inaccessibility period during which one of the application's services is not available. The proposed analysis allows computing the adequate resources required by the mobility framework to guarantee the timeliness of the application. Finally, Section IV discusses the model provided in the paper and presents some conclusions

2. System Model

2.1 Module components

This work applies to soft real-time applications composed by a set of interconnected components, each supplying some service, either in the same local node, but particularly when components are distributed among several nodes. The model considers the system to be composed of a set of N nodes $\{H_1, ..., H_N\}$ and a set of M services $\{S_1, \ldots, S_M\}$. Services are interconnected through links. $l_{x,y}$ characterises a connection between services S_x and S_y, (Figure 1). Each service and each link has a set of real-time requirements that are out of the scope of this paper (a detailed discussion can be found in [4]).

Each node runs a *Mobility Management* module M_x, where x is the index of the node. Each module M_x can be connected to other Mobility Management modules M_y through a network connection, $l_{mx,my}$.

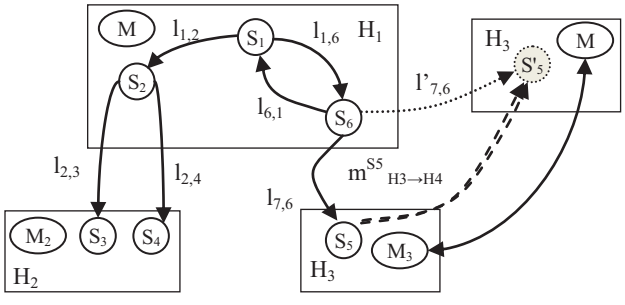


Figure 1. System Model

As depicted in Figure 1, an operation $m^{S5}_{H3 \rightarrow H4}$ represents the mobility of service S_5 between nodes H_3 and H_4. In such case, H_3 is denoted as the *source* node and H_4 is denoted as the *destination* node. Link $l'_{7,6}$ represents the connection that has to be established after the mobility operation is completed (rebinding). Consequently, connection $l_{7,6}$ will have to be safely deleted prior to $l'_{7,6}$ becomes operational. By safely, we mean that no messages should be lost or delivered to wrong nodes. This operation implies offloading the code of S_5, its data state, and rebinding its connections, all within timing constraints.

2.2 Resource management

It is assumed that access to the system's resources can be modelled as a set of isolated servers, either related to CPU [3] or network [2] scheduling. Each of these servers is characterised by its maximum reserved capacity (Q_i) that can be used during a period (T_i); at the end of this period the capacity is replenished. Other CPU schedulers can also be used, like the Capacity Sharing and Stealing scheduler (CSS) proposed in [13]. For the network scheduling, any scheduling algorithm with similar characteristics can also be used, like the ones based on the Flexible Time-Triggered approach [14].

Based on this guarantees, it is possible to determine services' average response time using the formulations proposed in [3]:

$$R_i^{avg} = C_i^{avg} + (T_i - Q_i) \sum_{k=0}^{+\infty} \left(1 - F_C(k \times Q_i)\right) \qquad (1)$$

where C^{avg}_i represents the average execution time of task Ti and $F_C(x)$ is the cumulative distribution function (c.d.f.) of the task's execution time. In the remainder of the paper, we will use the notation $R(Q_i, T_i, F_C^i())$ to represent Equation (1).

2.3 Mobility Management Framework

We assume the existence of a modular Mobility Management framework in each node, similar to the one proposed in [9].

These mobility management modules have CPU and networking servers assigned to them, guaranteeing the timing requirements of its operations. Servers associated with the CPU offer a capacity of C_F over a period T_F. Network resources are split between two channels, one for bulky data transfer and another for the exchange of short control messages. The first has a capacity of B_{data} and a period of T_{data} while the second has a capacity of B_{ctrl} and a period of T_{ctrl}. The main advantage of using these two channels is that we can guarantee small response time for control messages, but for larger data transfer we are able to make the transfer with small overhead.

2.4 Service's internal state

In the proposed model, services are able to split their internal state into different *State Items*, representing different variables, different objects or combinations of both. It is up to the service to define how state items are configured. The state of a service is thus a set of state items defined exclusively by the service, where a State Item (SI^{Si}_p) is only associated to a service S_i, is defined as a tuple:

$$SI_p^{Si} = \{ID_p^{Si}, B_p^{Si}\}$$

ID^{Si}_p univocally identifies this State Item and B^{Si}_p is the bandwidth required for the transfer of this state item. Some state items are created during the service initialisation and are not changed subsequently, while others are updated regularly when service calls are executed. Therefore, state items are divided in two groups: one

that can be migrated during the normal operation of the service (S*tatic Sate Items*) and another that can only be migrated if there are no ongoing service calls (*Dynamic State Items*).

Based on the model exposed in this section, Section III shows how it is possible to devise a timing model for a generic mobility mechanism.

3. Code Mobility Timing model

Service mobility can be split in two main phases: *Preparatory* and *Blackout* phases. During the *Preparatory* phase, the migrating service continues operational in the source node. This phase is further divided into three subphases: *mobility decision*, *code shipping*, and *initial state transfer*. During the *Blackout* phase, the service is totally inaccessible to others. It includes the subphases: *state transfer*, *connections rebinding*, and *service restart*. Some of the subphases are executed serially while others can be executed in parallel. Figure 2 depicts a timeline containing an example of a mobility procedure. A detailed description and analysis of each step is given in the following subsections. In this analysis, for the sake of simplicity, we assume that no other service mobility operation occurs during the complete procedure and that a higher-level resource control framework assures such control.

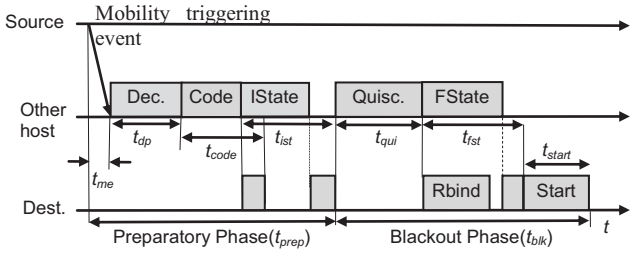


Figure 2 – Mobility-related timings

3.1 Preparatory Phase

3.1.1 Decision process

The start of service mobility (*mobility triggering event*) results from a decision by the application (currently using the service) or by request from an external entity (the user, another application or specific framework). As an example, in Figure 2, the triggering event is received from another node.

State changes can also trigger service mobility whenever a user requests the execution of an application that can only be admitted into the system if the system is reconfigured by migrating some services of previously admitted applications to neighbour nodes. As an example, consider that a user decides to play an mp3 file in its mobile device, having to migrate part of a local application to a neighbour notebook.

Note that migration can only be allowed if there is a feasible system configuration that allows the service to continue operating within its required QoS levels. Algorithms such as those provided by the Prism [1] or CooperatES [4] frameworks take a high-level approach, finding a solution for the distribution of the application services between nodes in a way that maximises a global utility function and, simultaneously, guarantees enough resources (CPU, network, memory, etc) for every admitted service. While the former assumes just one possible QoS level to the application, the latter assumes that each service can work with multiple QoS levels, each one with a different utility value to the overall system. Additionally, the algorithms proposed in [4] are capable of generating a system configuration in a bounded amount of time. These algorithms are able to use a global view of the system state or can simply use a partial view of the system, e.g. if the node computing a decision only has access to a limited number of nodes.

We should point out that the algorithms proposed in [1] and [2] do not take into account the cost introduced by systems reconfiguration and particularly code mobility.

3.1.2 Code shipping

After finding a new distributed solution, the source node informs the destination node of the QoS requirements for the service being migrated. The destination node can then make all necessary local confirmations on the feasibility of receiving the service.

Then, the service code is coded (*e.g.* for data serialisation) for transmission on the source node, shipped through the network, and decoded on the destination node (e.g. by using a deserialisation method).

The bandwidth required to transfer the code is equal to β_{code}, a constant, since the code size is not expected to vary during transit. Therefore, the average time required for the transmission of code (t_{code}) can be calculated by:

$$t_{code} = R\left(C_F^S, T_F^S, F_C^{code,S}()\right) + R(B_{data}, T_{data}, \beta_{code}) + R\left(C_F^D, T_F^D, F_C^{code,D}()\right) \quad (2)$$

$F_C^{code,S}()$ and $F_C^{code,D}()$ are the c.d.f. of the execution time required by the framework, on the source and destination nodes, respectively.

3.1.3 Initial state transfer

We assume that a set of *Static State Items* (*e.g.* configuration data) can be transferred prior to the quiescence of the service on the source node. After the transfer of the state items, the destination node acknowledges its reception. Consequently, the delay associated with the initial state transfer is given by:

$$t_{ist} = R\left(C_F^S, T_F^S, F_C^{ist,S}()\right) + R\left(B_{data}, T_{data}, F_C^{ist,N}()\right) + R\left(C_F^D, T_F^D, F_C^{ist,D}()\right) \quad (3)$$

where $F_C^{ist,N}()$ is the c.p.f. for the required bandwidth, $F_C^{ist,S}()$ and $F_C^{ist,D}()$ are the c.p.f. of the CPU processing time, on the source and destination node, respectively.

3.1.4 Total delay of the Preparatory phase

The time required for the *Preparatory* phase is given by:

$$t_{prep} = t_{me} + t_{dp} + t_{code} + t_{ist} - R\left(C_F^D, T_F^D, F_C^{code,D}()\right) \quad (4)$$

where t_{me} is the time that elapses from the event that triggered the mobility of a service until being received by the node responsible to determine a new system configuration. It is assumed that the new system configuration is computed in a bounded time t_{dp} [4].

It is important to note that, depending on the scenario, some of these timings can be equal to zero. As an example, assume the case where it is the user that decides to migrate its application from its mobile device to its TV.

Most importantly, during this phase the service continues totally operational, but the characterisation of this delay is required in order determine the dynamic of the mobility procedure.

3.2 Blackout Phase

3.2.1 Quiescence achieving

Usually, in reconfiguration operations, the service to be updated has to be in a safe state called quiescence [7]. In this state, the service being migrated:

i) is not currently engaged in a transaction;
ii) will not initiate a new transaction;
iii) is not servicing a transaction; and
iv) no transaction has or will be initiated by other services that require service from this service. At the same time, all services connected with the migrating service must go into a passive state, which requires the fulfilling of condition i) and ii).

One initial solution to achieve quiescence has been proposed in [7], while a less demanding solution, called tranquillity was later proposed in [8]. Achieving quiescence requires the completion of pending requests by a service and the knowledge of all other services that might issue new requests. These other services must evolve into a passive state in which they cannot evoke the service being migrated, although they can evoke other services available in the system. The time needed to achieve quiescence can be determined through a timing analysis of the mechanisms proposed in [7] or [8]. This calculation, out of the scope of this paper, is assumed to be known and equal to t_q.

We argue that achieving quiescence is not a necessary condition for the mobility of services in a distributed system, as shown by the implementation described in [9], if the service calls are stored by the mobility management and delivered to the destination node only after the completion of the mobility procedure.

3.2.2 Final state transfer

Several different approaches can be considered for state transfer:

i) transfer all state in a single bundle [10];
ii) propagate only the operations done on state items [5];
iii) separate the state space into several groups of items, each transferred with its own periodicity [6] or;

retransmit the state whenever it changes [12]. The mobility model here considered adapts to these approaches.

The final state transfer is the subphase that most influences the latencies of a service migration, due to its duration and due to the service being in a quiescent state (it involves the transfer of *Dynamic State Items* which can only maintain consistency if the service is not operational).

The set of state items that can only be transferred after achieving quiescence require a bandwidth of $F_C^{fst,N}()$ and CPU processing requirements of $F_C^{fst,S}()$ and $F_C^{fst,D}()$, respectively on the source and destination nodes.

CPU processing time is required for the preparation of the data to be sent and the required processing time to decode the data on the destination node. Therefore, the final state transfer duration (t_{fst}) can be calculated, similarly to the case of t_{ist}, as follows:

$$t_{fst} = R\left(C_F^S, T_F^S, F_C^{fst,S}()\right) + R\left(B_{data}, T_{data}, F_C^{fst,N}()\right) + R\left(C_F^D, T_F^D, F_C^{fst,D}()\right) \quad (5)$$

3.2.3 Connections rebinding

In the migration process, connections between services need to be changed according to the new location of the migrating service.

This procedure can be performed in parallel with the *final state transfer* and it involves the exchange of messages between 2 or more nodes: the source, destination and, if any, other nodes whose services connect to the service being migrated. It mainly requires the exchange of messages containing the location of the new end points, which requires a bandwidth of β^{Si}_{reb}. Therefore, assuming that service S_i has $ncon^{Si}$ connections with other nodes, the total bandwidth required to rebind all connections (β_{reb}) is $ncon^{Si} \times \beta^{Si}_{reb}$. The time required internally by each service to change the connection end point addresses is considered negligible.

The exchanged messages can also be used to withdraw all connected services from the passive state. Therefore, the rebinding time (t_{rbind}) is given by:

$$t_{rbind} = R\left(B_{ctrl}, T_{ctrl}, \beta_{reb}\right) \quad (6)$$

Since the number of exchanged message can be high, but with a small payload, its transmission is performed by the communication server assigned for control messages.

3.2.4 Service restart

The final subphase, which starts at the end of both the connection rebinding and final state transfer subphases, is responsible for the restart of the service on the destination node. All code and state must already be on the destination node and all necessary operations for the installation of the service (if required) have been completed. After being started the service re-establishes its internal state using the state items previously transferred and enters full operation. This operation is performed by the service using its scheduling budget (C^D_{Si}, T^D_{Si}), and therefore the time required for service restart is given by:

$$t_{rstart} = R\left(C_{Si}^D, T_{si}^D, F_C^{rstart,D}()\right) \quad (7)$$

where $F_C^{fst,S}()$ is the p.d.f. of the CPU requirements for service restart.

3.2.5 Total delay of the Blackout phase

During this phase, all transactions involving the migrating service are stopped, thus leading to a blackout period (t_{blk}). On a real-time system this time is particularly important since it influences the timeliness of the distributed application. Therefore, the total duration of the *Blackout* phase is given by:

$$t_{blk} = t_q + max\{t_{fst}, t_{rbind}\} + t_{rstart} \quad (8)$$

Since the final state transfer and the rebinding of connections can be executed in parallel, then we use the function $\max\{t_{fst}, t_{rbind}\}$ to determine the maximum of both subphases.

As discussed previously, the Quiescence Achieving subphase might be eliminated if the system is supported by adequate mobility management facilities. The rebinding process is based on a simple exchange of messages and on the reconfiguration of transmission and receptions ports. The service restart is an operation with a small overhead. But, the final transfer subphase delay varies with the size of the data being transferred. Particularly, when the state size is high, strategies like the ones proposed in [12] can be used in order to reduce t_{fst}. Such strategies enable the implementation of partial blocking and non-blocking approach on service calls for a migrating service.

4. Mobility framework Architecture

A Mobility Framework, which enables the mobility of services on the Android Operating system, has been developed [9]. The framework will used to demonstrate the use of the proposed model on real scenarios.

The framework is implemented as an Android service, which takes care of service migration to and from another node, at the same time it interacts with the operating system Resource Manager in order to determine if the QoS requirements of the service can be supported.

The Android operating system is used both due to its open source nature to its innovative architecture. Although its use to support real-time applications is still debatable [15] it nevertheless provides a suitable architecture for quality of service-aware applications in ubiquitous, embedded systems [16].

The core services provided by the framework are the: *Discovery Manager*, *Package Manager*, *State Manager* and *Execution Manager*. Additionally, the framework also relies on a *QoS Manager* module that is responsible for assuring that QoS requirements of each service can be meet. Figure 3 depicts the main modules of the framework.

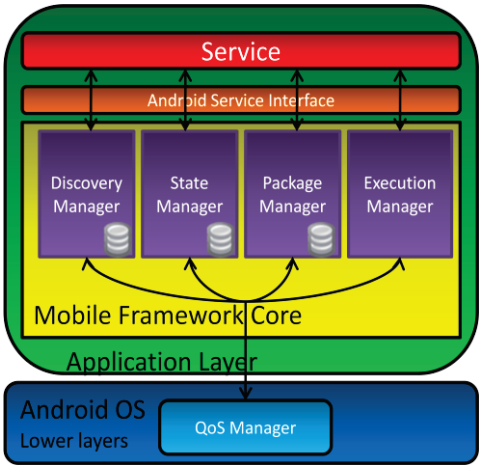


Figure 3 – Mobility framework

The *Discovery Manager* module is designed to discover neighbour devices on a local network and advertise the host device capabilities. Advertise messages contain information about the applications and services installed, their associated Intents interfaces and QoS requirements. Originally, Android intents provide the means for the reutilization of functionalities implemented by other application installed in the same device. Therefore, the *Discovery Manager* provides a standard mechanism, for each node, to obtain information about installed services and about the availability of resources in neighbour devices. It also keeps track of node and service disconnections from the network.

The *Package Manager* is used to install, uninstall and transfer the code of Android services, which are contained in *APKs* files. This module is also responsible for the interaction with the *QoS Manager* in order to request specific QoS levels for the service being handled. Therefore, it is the responsibility of the *QoS Manager* to accept or reject service installations, particularly if the QoS required level cannot be guaranteed.

The *State Manager* handles both the initial and final state transfer operations in a flexible way, based on the state items paradigm.

The *Execution Manager* allows launching services on a host device or on a remote node through the exchange of Android Intents that allow the programming of transparent applications (in relation to the distribution). In this implementation an Intent resolution procedure, based on the data collected by the Discovery Manager, determines if the Intent can be run locally or if it must be redirected to the node, where the service is running.

The *QoS Manager* administers the system resources, either locally, on a node, or in a distributed environment. It also encapsulates the functionalities of high level QoS control frameworks, like the one defined in [4]. Consequently, this module can interact with our framework conveying orders for the deployment of services in the distributed system.

5. Conclusions and Future Work

In recent years, many real-time systems have become open to unpredictable operating environments where both system workload and platform may vary significantly at run time. As such, the set of applications to be executed and their aggregate resource and timing requirements are unknown until runtime but, still, a timely answer to events must be provided in order to guarantee a desired level of performance.

In this context, a distributed execution of resource intensive applications among neighbour nodes seems a promising solution to address the increasingly complex demands on resources and desirable performance.

This paper proposed a generic model for code mobility in soft real-time systems, where applications are constituted by interconnected distributed services.

The main consequence of mobility to the running application is that it might result on a temporary degradation on the provided quality of service, due to the consequent blackout period. We state that it is up to the application programmer to determine the amount of degradation that can be supported by the application.

As such, this work gives the system designer the necessary tools to perform a statistical timing analysis on the execution of the mobility mechanisms and to determine the most appropriate parameters of the mobility framework components, either in relation to the local (CPU) or to network resources.

The proposed model divides the mobility mechanism in two phases, thus allowing a reduction on the time during which a service is inaccessible (the *Preparatory* phase is not considered).

This work can leverage future research in the field of code mobility and service update in distributed real-time systems. The proposed analysis can support the development and evaluation of suitable mobility mechanisms. Future work will focus on the use of the state items paradigm to propose new state transfer algorithms.

REFERENCES

[1] S. Malek, G. Edwards, Y. Brun, H. Tajalli, J. Garcia, I. Krka, N. Medvidovic, M. Mikic-Rakic, G. Sukhatme, "An Architecture-Driven Software Mobility Framework," Journal of Systems and Software, Vol. 83 Issue 6, June, 2010, pp 972-989.

[2] T. Nolte and K. Lin, "Distributed Real-time System Design using CBS-based End-to-end Scheduling," in Proc. of the 9th International conference on Parallel and Distributed Systems, pp. 355 – 360, 2002.

[3] L. Abeni, G. Buttazzo, "Integrating multimedia applications in hard realtime systems", in Proceedings of the 19th IEEE Real-Time Systems Symposium, Madrid, Spain, 1998, p. 4.

[4] L. Nogueira and L. Pinho, "Time-bounded Distributed QoS-Aware Service Configuration in Heterogeneous Cooperative Environments", in Journal of Parallel and Distributed Computing, Vol. 69, Issue 6, June 2009, pp. 491-507.

[5] D. Bourges-Waldegg, Y. Duponchel, M. Graf and M. Moser, "The fluid computing middleware: bringing application fluidity to the mobile Internet", in Proc. of the 2005 Symposium on Applications and the Internet, pp. 54- 63, 2005.

[6] D. Preuveneers and Y. Berbers, "Context-driven migration and diffusion of pervasive services on the OSGi framework", in International Journal of Autonomous and Adaptive Communications Systems, Vol. 3, No. 1, pp. 33-22, 2010.

[7] J. Kramer and J. Magee, "The Evolving Philosophers Problem: Dynamic Change Management", in IEEE Trans. on Software Engineering, Vol. 16, Issue 11 (Nov. 1990), pp. 1293-1306.

[8] Y. Vandewoude, P. Ebraert, Y. Berbers and T. D'Hondt, "An alternative to Quiescence: Tranquility", in Proc. of the 22nd IEEE Int. Conf. on Software Maintenance, Washington, DC, (Sep. , 2006), pp. 73-82.

[9] J. Gonçalves, L. Ferreira, L. Pinho and G. Silva, "Handling Mobility on a QoS-Aware Service-based Framework for Mobile Systems", in Proc. of the 8th IEEE International Conference on Embedded and Ubiquitous Computing (EUC 2010), Hong Kong, December 2010, to be published.

[10] M. ALRahmawy, A. Wellings, "A model for real time mobility based on the RTSJ," in Proc. of the 5th international Workshop on Java Technologies For Real-Time and Embedded Systems (Vienna, Austria, Sep. 2007), vol. 231. ACM, New York, NY, pp. 155-164.

[11] B. K. Choi, S. Rho, R. Bettati, "Fast software component migration for applications survivability in distributed real-time systems," in Proc. of the 7th Object-Oriented Real-Time Distributed Computing, Vienna, Austria, May 2004, pp.269-276.

[12] E. Schneider, "A Middleware Approach for Dynamic Real-Time Software Reconfiguration on Distributed Embedded Systems", PhD Thesis, Université Louis Pasteur – Strasbourg, 2004.

[13] Nogueira, L., Pinho, L., "A Capacity Sharing and Stealing Strategy for Open Real-time Systems", Published in Journal of Systems Architecture, Volume 56, Issues 4-6, April-June 2010, pp. 163-179.

[14] P. Pedreiras, P. Gai, L. Almeida, G. Buttazzo, "FTT-ethernet: A flexible real-time communication protocol that supports dynamic QoS management on ethernet-based systems", IEEE Transactions on Industrial Informatics, vol. 1, no. 3, p. 162-172, August 2005.

[15] Maia, C., Nogueira, L., Pinho, L., "Evaluating Android OS for Embedded Real-Time Systems", Proceedings of the 6th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT 2010), Brussels, Belgium, 2010, pp. 63-70.

[16] Maia, C., Nogueira, L, Pinho, L., "Cooperative embedded application in Android Environments", Submitted for publication on the 8th International Workshop on Java Technologies for Real-time and Embedded Systems - JTRES 2010.

Evaluating the Benefits and Feasibility of Coordinated Medium Access in MANETS

Marcelo M. Sobral
Federal University of Santa Catarina
PGEEL - CTC - Campus Universitario - Trindade
Florianopolis, SC, Brazil
sobral@das.ufsc.br

Leandro B. Becker
Federal University of Santa Catarina
DAS - CTC - Campus Universitario - Trindade
Florianopolis, SC, Brazil
lbecker@das.ufsc.br

ABSTRACT

Mobile ad hoc wireless network (MANETs) are characterized by a highly-dynamic topology where neither the duration of links between nodes nor their densities within the network can be foreseing. To better understand the effects of such issues in the medium access we provide a performance evaluation of two distinct MAC protocols. The first is our previously proposed HCT (Hybrid Contention/TDMA) Real-Time MAC protocol, which continuously adapts to topology modifications in order to provide a kind of coordinated medium access. Its performance is compared with a contention-based, non-coordinated CSMA protocol, which is the typical MAC protocol used in MANETs. We analyze both protocols with respect to their ability to deliver messages in a timely manner. More specifically, we compared the ratio of messages delivered within their deadlines and the medium utilization provided by these protocols. Such aspects where analyzed considering mobile networks with different spatial densities and speeds of nodes. This study also addresses the protocols overhead, especially for HCT-MAC. Obtained results show that HCT-MAC appears as a good solution for applications like search-and-rescue, autonomous highway driving (platooning), and multimedia, which require some kind of QoS guarantee in respect to the timely delivery of messages.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organization**]: Computer-Communication Networks—*network protocols*; C.4 [**Computer Systems Organization**]: Performance of Systems

General Terms

Performance,Experimentation

Keywords

Wireless Networks, MAC, MANETs, Real-Time Communication

1. INTRODUCTION

Analyzing some new-generation embedded applications one can see that they rely on mobile-connectivity. For instance, in the CarTel project [4] data is collected from sensors located on automobiles that move around the city. Modern Intelligent Transportation Systems use vehicle-to-vehicle (V2V) systems like platooning, which helps to reduce traffic congestions and provide safe driving [18]. The space community is developing distributed satellite systems (DSS) [2], where multiple mini-satellites in varying configurations are used to achieve a mission's goals collaboratively. Most of these applications require some kind of QoS guarantee in respect to the timely delivery of messages.

It happens that mobility causes topology changes and temporary link disruptions, affecting communications predictability. So the challenge in this context is how to rely on wireless links to achieve timing guarantees. This issue presents a kind of contradiction in the real-time domain, as it conflicts with the need for temporal determinism. This allows us to conclude that a suitable MAC for such scenario would be the one that can prompt react to topology changes, so that the effects of mobility are minimized.

Several MAC protocols where designed to be used in ad hoc networks, but most of them did not take into account mobility issues, as contantion-based medium access has been the typical solution to deal with mobility. In [9], Kumar et al presented a survey about MAC protocols used in ad hoc wireless networks. The well known Z-MAC [13], for instance, is a dynamic protocol that adapts itself to the network conditions, using CSMA during normal workload and TDMA in high workload. Its drawback comes from the high overhead for reconfigurations (about 30s according to authors), which makes it not suitable for mobile applications. Another example is the AdHoc-MAC [1], which was conceived for inter-vehicles communication using a distributed TDMA slot allocation mechanism named RR-ALOHA. Its drawback comes from the need of configuring the application offline, making it not applicable for mobile applications.

Despite the limitations of such protocols, it is possible to observe that hybrid approaches to medium access control are the key to achieve timely behavior in mobile networks. Inspired on that we proposed the so-called Hybrid Contention/TDMA-based (HCT) MAC [14], which aims to provide a time bounded medium access control for mobile nodes that communicate through an ad hoc wireless network. The key issue in this protocol is to self-organize the network in groups of adjacent nodes called *clusters*, as a mean to solve the problem of timely transmission of messages. It assumes a periodic message model and a transmission cycle divided in time-slots, where each cluster reserves a predefined number of time-slots that can be assigned to its member nodes.

The current paper presents the recent advances in HCT-MAC design and discusses results obtained from an intensive simulation study related to its application in MANETs ap-

plications that rely on timely delivery of messages. Its goal is to emphasize the benefits of having coordinated medium access to achieve the timing requirements when compared to the traditionaly used contention-based approach (CSMA).

In the simulations we compare the ratio of messages delivered within their deadlines and the medium utilization presented by HCT-MAC and CSMA protocols, considering networks with different spatial densities and speeds of nodes. We also present some hints on the existing performance bounds of the protocols. To conclude, our study analyzes the feasibility of using an adaptive protocol like HCT-MAC in respect to its overhead.

The remainder of the paper is structured as follows. Section 2 discusses the main related works. Section 3 provides an overview of our HCT-MAC protocol. Section 4 details the performance metrics to be analyzed in our evaluation. Section 5 presents the simulation experiments and discusses the obtained results. Finally, section 6 concludes the paper.

2. RELATED WORKS

The great majority of research works concerning MAC protocols for MANETs tackle the use of contention-based protocols [9]. Indeed, these protocols easily adapt to topology changes, since no agreement between nodes is required prior to transmissions. That is the case of CSMA/CA protocols, like the variations implemented in the standards IEEE 802.11 [7] and IEEE 802.15.4 [6].

However, despite their adaptability to topology changes, they are not predictable regarding medium access delay, so cannot guaranty a timely delivery of messages. This happens mostly due to collisions and inherent random backoffs. For this reason we investigated other MAC protocols that address the timely delivery of messages using ad hoc networks. But, on the other hand, few of them were designed taking mobility issues into consideration.

Some contention-based MAC protocols include prioritization mechanisms, which could be used to implement a scheduling policy. For instance, the Black Burst is a MAC protocol which employs a preamble with variable length to prioritize messages, but it does not adapt to frequent changes in topology [16]. The WiDom is another contention-based MAC protocol, which adapts the dominance protocol used in the CAN bus to a wireless channel, implementing static-priority scheduling with a large number of priority levels [11], but it was designed to networks with static topologies. In the case of the IEEE 802.11 standard, the different IFS (Inter Frame Space) and initial contention windows defined in the access categories of EDCA, a statistically prioritized version of its CSMA/CA MAC protocol, can only provide a coarse-grained prioritization with few priority levels and do not avoid collisions [5].

Another set of MAC protocols called hybrid combine contention and resource-reservation to perform medium access. The Z-MAC [13] defines a TDMA transmission cycle where each node can allocate one time-slot and use it to transmit messages in a contention-free manner. It also allows nodes to use other nodes time-slots, but using CSMA with lower priority. Z-MAC was designed for wireless sensor networks with static topologies, so its main drawback in mobile networks resides in the delay it can suffer when nodes need to allocate time-slots, since it employs a distributed consensus protocol.

The IEEE 802.15.4 standard also includes a hybrid operation mode, using contention-based medium access with CSMA/CA and contention-free medium access with GTS (Guaranteed Time Slots). In that mode, groups of nodes are formed to share a superframe, which is a cyclic interval of time when member nodes can receive and transmit messages. Superframes are started by a control frame called beacon, that must be transmitted by the group coordinator. The standard does not define how the group coordinators must schedule their superframes (i.e., how to schedule beacon transmissions), such that the superframes of different groups of nodes do not overlap. In [8] it was proposed a beacon scheduling algorithm for networks with cluster-tree topologies. Proposals for mesh topologies where presented in [3] and [10], however they considered only networks with static topologies. Thereby we conclude that existing hybrid MAC protocols do not cope with dynamic topologies, and thus are not suitable for mobile networks.

3. THE HCT-MAC PROTOCOL

The Hybrid Contention/TDMA-based (HCT) MAC was designed to provide time bounded medium access control for nodes that communicate through a mobile ad hoc wireless network (MANET). While the first ideas around this protocol where presented in [14], the protocol has been improved over the last years. This section details the core concepts and the most recent aspects related with HCT protocol.

3.1 Adopted Network Model

It is assumed a network model where mobile nodes communicate through a MANET. Nodes move continuously according to some mobility model, leading to the creation and extinction of data links among them. Transmissions are performed in broadcast and there are no acknowledgements in the MAC layer. Nodes exchange both data and control messages in a periodic manner. We assume a pessimistic scenario, which means that nodes can transmit data messages at the same time.

Control message periods (or transmission cycles) are static and defined offline. This way, within the neighborhood of each node, there is a limit in the amount of messages that can be transmitted without collisions within a transmission cycle. This suggests that designer should take a closer look into the application ir order to properly parameterize the protocol.

Since in MANETs the topology changes frequently, it is not possible to guarantee that a node can always transmit one message at each cycle. This can be related to the density of nodes, defined by the number of nodes whose transmissions can collide at a given location in the network. If that density exceeds the limit of messages per cycle, some nodes will not be able to transmit their messages. Besides that, in the adopted network model nodes perform an opportunistic resource-reservation, such that they can obtain a temporary throughput guarantee. Since network topology is dynamic, such resource-reservation must adapt to solve conflicts that can arise when neighborhoods of nodes change.

3.2 Protocol Overview

A key issue in HCT design is how to self-organize the network nodes to coordinate transmissions. The adopted solution consists in creating dynamic groups of adjacent nodes called *clusters*, as discussed in [14]. It assumes that wireless links within groups of nodes can last enough to allow the

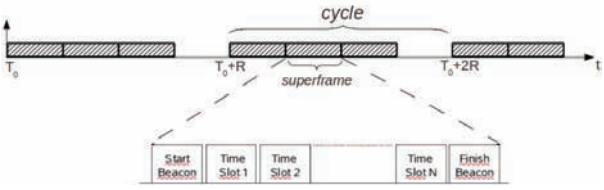


Figure 1: Timing in HCT: cycles of length R divided in superframes

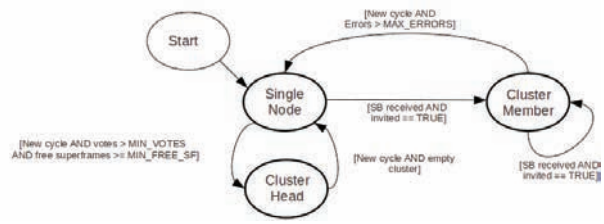


Figure 2: Clustering in HCT-MAC

use of a resource-reservation approach among clusters.

The HCT-MAC is a hybrid protocol because it has both contention-based and resource-reservation characteristics. It accesses the medium with contention is performed in a CSMA-like manner and resource-reservation is implemented similarly to a TDMA. Initially all nodes operate in contention-based mode and, as they succeed to form clusters, they might operate in resource-reservation mode.

In HCT time is organized using a periodic and hierarchical structure, as illustrated in figure 1. A *cycle* is the basic period for transmissions, thus it works like a time unit for the protocol (it is an interval of time that is common to all clusters). Cycles are divided in *superframes* that are allocated by clusters. Finally, superframes are subdivided in *time-slots*, which can be used by nodes for message transmissions. Superframes are delimited by two control frames called *start beacon* and *finish beacon*. Transmission cycle length is a key parameter in HCT-MAC. It limits the number of superframes per cycle and, consequently, the number of available clusters.

The TDMA component of the HCT depends on the clustering of the nodes, which must be performed in a self-organized manner. Self-organization is a requirement because the HCT protocol was designed to be used in mobile ad hoc networks, where nodes are not previously aware of the topology, neither of their neighborhoods. The protocol assumes that each node performs initially a contention-based medium access and, as they become a cluster member, they switch to reservation mode. In other words, as nodes self-organize in clusters they can reserve bandwidth and transmit messages in a timely manner. More information about this procedure can be obtained in [15].

In HCT-MAC clusters represent sets of nodes that agree to share a superframe, which represents a portion of the network bandwidth. It must be noted that a node can send messages to any other node within its range, since clustering has the single purpose of helping nodes to allocate time-slots within a superframe. A key element in the cluster topology is the cluster-head, a special node responsible to start cluster transmissions with a start beacon, to account for idle and used time-slots, and to report successful transmissions within a finish beacon sent in the end of a superframe. Ideally, the cluster-head should be the node with the best link qualities to adjacent nodes within the region to be covered by the cluster, in order to minimize the probability of transmissions errors in the scope of the cluster.

Table 1: Evaluation Metrics

Metric	Description
Rate of received frames	Ratio between received frames and number of time-slots
Rate of delivered frames	Ratio between successful delivered frames and transmitted frames
Clusterized rate	Ratio between clusterized cycles and total cycles
Disconnected time	The interval of time a node is expected to wait to enter a new cluster

The clustering procedure is driven by two main guidelines: i) single nodes elect the best nodes to become cluster-heads and ii) cluster-heads choose and invite the best nodes to become cluster members. Figure 2 shows how nodes change their roles between cluster-head, member node and single node. According to that, each node starts as single node, and can change to member node if an invitation is included in the start beacon received from a cluster-head. It becomes again single node if it received a start beacon which does not include an invitation, or if it does not receive a start beacon within one transmission cycle. To become cluster-head, a single node must receive enough votes from its neighbours. Finally, a cluster-head becomes single node if its cluster is empty. Clustering is performed continuously, such that HCT can adapt to topology changes which affect links qualities between nodes.

4. EVALUATION METRICS

This section presents and discusses the set of metrics used in our experiments, summarized in table 1. The main goal of the experiments were to analyze the benefits of using a hybrid protocol like HCT to provide timely delivery of messages and also to achieve better medium utilization. Another target of the experiments was to evaluate the feasibility of using HCT, i.e., to estimate the overhead of its coordination mechanisms.

Timely delivery of messages is implicit in HCT, i.e., if messages arrive the destination they are within the deadline. This is due to the TDMA component of HCT.

Medium utilization is a prominent result of a MAC protocol, because it informs how much of the channel capacity can be effectively used. A MAC which presents a given probability of transmission errors due to collisions cannot fully utilize the channel capacity. Moreover, in this case some messages are expected to be lost due to collisions. In fact, contention-based MACs, like the well known CSMA/CA [6, 5] and its variations, perform a probabilistic medium access and commonly use random delays before transmitting messages to reduce the probability of collisions. However, a MAC that employs some kind of coordination among transmissions of different nodes, like our proposed HCT-MAC, can improve the medium utilization. In this case, channel can be bet-

ter utilized if collisions are very unlikely and random delays become unneeded.

The performance of HCT with respect to medium utilization was investigated by two metrics called i) *rate of received frames* and ii) *rate of delivered frames*. The first metric gives the ratio between the number of received frames and the maximum allowed according to TDMA. The second metric refines the former metric by accounting for the successful delivered frames, which were received by nodes for whom they were addressed. When combined, these metrics can estimate how close of TDMA performance HCT was. scenario defined by spatial density of the network and mobility pattern.

HCT performance in respect to medium utilization is expected to lie between a pure contention-based MAC like CSMA and a pure TDMA MAC. In case of TDMA and assuming a frame fills one time-slot entirely, if a transmission cycle T has N time-slots, a node can receive at most $N-1$ frames per cycle. In a CSMA MAC, nodes contend for the medium and thus their transmissions are subject to collisions. If nodes transmit frames with period T using CSMA, the number of frames each node receives is expected to be smaller than $N-1$ due to collisions. Since HCT combines both medium access modes, the amount of frames each node receives should be upper bounded by TDMA and lower bounded by CSMA.

The expected enhancement in medium utilization and timely delivery of messages that can be provided by HCT depend on the feasibility of its resource-reservation mechanisms. The resource-reservation access mode of HCT depends on the network self-organization in clusters, which occurs continually. Nodes form a cluster when some node is elected as cluster-head and invites other nodes to use the time-slots which are available to the cluster, as explained in section 3. *Both election of cluster-head and invitation of cluster members are driven by the measured link quality estimation performed by each node, in such way a cluster can be composed by nodes with good relative links qualities. In a mobile network, links qualities change over time due to nodes movements, which implies clusters being modified or dissolved, and new clusters being formed. This way, each node can alternate intervals of time when it is member of cluster and when it waits to enter a new cluster.*

In order to evaluate such feasibility, two metrics where defined: *clusterized rate* and emphdisconnected time. The metric *clusterized rate* represents the average ratio of the number of clusterized cycles experienced by each node and the total number of transmission cycles. Since there is a limit in the number of possible clusters within 2 hops, the *clusterized rate* is expected to depend both on the network size and spatial density. Moreover, mobility can make clusterized intervals shorter.

The so-called *disconnected time* represents the interval of time a node is expected to wait to enter a new cluster. This comes from the fact that mobility implies changes in clusters memberships, what means that a clusterized node can leave its cluster (or even its cluster can be dissolved) and wait some time until entering a new cluster. Once outside a cluster a node cannot benefit from the contention-free medium access provided by HCT. The *disconnected time* is calculated as a cumulated probability density function which gives the probability that a node suffers a given delay to enter a new cluster.

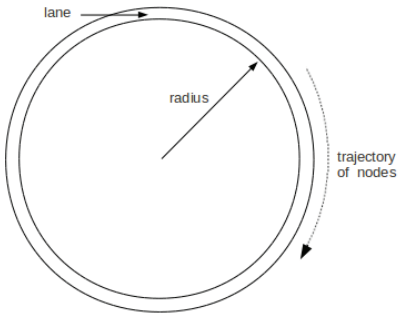


Figure 3: Mobility model: nodes move following a circular trajectory

5. SIMULATION EXPERIMENTS

This section presents a simulation study developed in order to provide a clear understanding on the ability of HCT-MAC protocol to fully utilize the medium and deliver frames within their deadlines in networks with mobile nodes, compared to a traditional CSMA protocol. It also investigated to which extent HCT was able to clusterize nodes in the simulated scenarios, which relates to the feasibility of its resource-reservation mechanisms.

Simulations were performed using the Omnet++ framework [17]. HCT model used as physical layer the radio and wireless channel models from project Castalia, maintained by the National ICT at the University of Australia [12]. They implement the signal model proposed in [19] and simulated a IEEE 802.15.4 compatible radio. These models were modified by us to support mobility.

In our simulations we used a sort of circular mobility model. Nodes moved along the 10 m width circular track (see figure 3) with variable radius. Groups of 40 to 60 nodes were disposed randomly along the circular track, moving in the same direction with speeds between 0 and 40 m/s, with a 2 m/s step. Once started a simulation, the speeds of nodes did not change. The track radius ranged from 10 up to 150 m, in the case of networks with 40 nodes, and 30 up to 300 m in networks with 60 nodes, both using steps of 10 m.

The spatial densities of the networks were calculated by the ratio between that enclosed area and the number of nodes, and was expressed as the average distance between nodes. This way it was possible to vary the spatial density of the networks and their degree of mobility. The physical layer parameters were chosen to simulate an indoor environment with no obstacles between nodes, and typical transmission range of 150 m. Table 2 summarizes the simulation parameters.

In respect to the network load, each node in the simulation periodically sent a message addressed to the neighbour which presented the best link quality in the previous transmission cycle. The resulting workload was balanced such that all nodes sent and received (in average) the same amount of messages.

HCT specific parameters are summarized in table 3. Each transmission cycle in HCT had 6 superframes with 8 time-slots each, with time-slots length of 1 ms. It allowed clusters with at most 7 nodes, as 2 time-slots are reserved for Start and Finish Beacons, but cluster-heads used Start Beacons to encapsulate their data messages. One superframe was

Table 2: General Simulation Parameters

Simulation Parameter	Value
Period of messages	48 ms
Deadline	96 ms
Maximum hops	1
Message length	16 bytes
Simulation Time	120 seconds
Mobility Model	Race (circle)
Circle Radius	from 10 upto 300 meters
Speed	from 0 upto 40 m/s
Number of Nodes	40 and 60
Sensitivity	-95 dBm
Default Transmission Power	-5 dBm
Thermal Noise	-100 dBm
Path loss exponent	2.4
Path loss at d0	55 dBm
d0	10 m

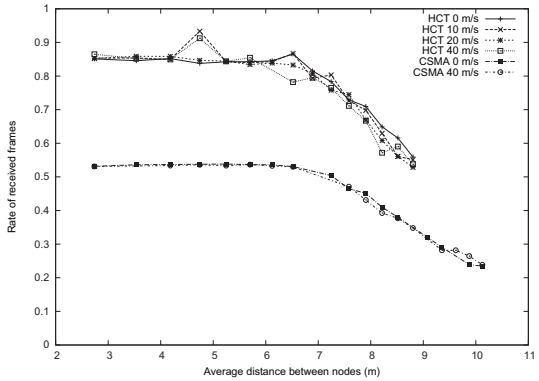
Table 3: HCT Simulation Parameters

Simulation Parameter	Value
Cycle length	48 ms
Time-slot	1 ms
Superframe size	8 time-slots
Number of superframes	6

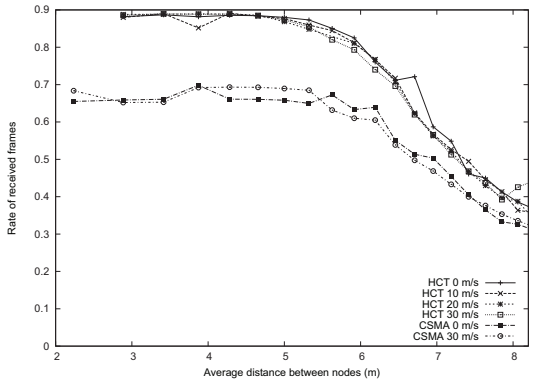
reserved for contention-based access, to be used by single nodes. In that case, such nodes contended for the medium only within unallocated superframes.

5.1 Analysis on Performance and Medium Utilization

The *rate of received frames* obtained for HCT with particular maximum speeds of nodes had a low variability, considering networks with 40 and 60 nodes. In both cases, the *rate of received frames* presented similar results with different speeds (from $0m/s$ to $30m/s$) as shown in figures 4(a) and 4(b). It must be noted that in these experiments the transmission cycle of HCT would allow at most 35 nodes in resource-reservation mode in every location of the network (i.e. 5 clusters with at most 7 nodes each, since one superframe was reserved to contention-based access). Therefore, in networks with 40 nodes almost every node could clusterize and operate in resource-reservation mode, even in higher spatial densities. However, in networks with 60 nodes this was not true unless the spatial density corresponded to neighborhood sizes around 35 nodes. It can be clearly seen that HCT outperformed CSMA as spatial density increased (i.e. as average distance between nodes decreased). In this case, as nodes got closer their neighborhood sizes increased, leading to a higher probability of collisions in CSMA. Since HCT organizes as many nodes as possible in clusters to perform a short-range resource reservation, these clusterized nodes could transmit without incurring in collisions. The curves also show that CSMA performance approached HCT as spatial density decreased. This can be related to the smaller resulting neighborhood sizes, which resulted in lower probability of collisions if CSMA was used. Finally, since CSMA does not perform any resource-reservation nor self-



(a) Network with 40 nodes



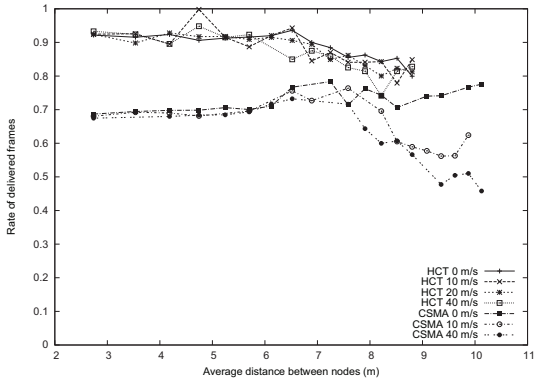
(b) Network with 60 nodes

Figure 4: Rate of received frames

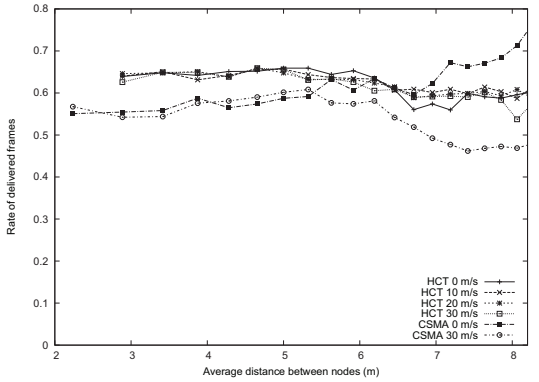
organization, it must be little affected by speeds of nodes as confirmed in the plots. A similar result was obtained for the *rate of delivered frames*, which accounts for the received frames which were actually addressed to receiving nodes.

Rate of delivered frames relates the amount of delivered frames with the number of data frames generated over the network. As shown in figure 5(a), HCT presented a significantly higher *rate of delivered frames* than CSMA in networks with 40 nodes, when higher spatial densities were considered. The variability of this *rate of delivered frames* did not present a significant dependence to speeds of nodes in the case of HCT, but with CSMA the results for lower spatial densities were better with lower speeds. In networks with 60 nodes, shown in figure 5(b), HCT still outperformed CSMA in higher spatial densities but with a smaller difference.

Obtained results regarding *rate of received frames* and *rate of delivered frames* showed that HCT presented a better medium utilization than CSMA. It also showed that in some scenarios HCT approached the performance that a TDMA MAC would provide with respect to medium utilization. The better performance of HCT can be related to its resource-reservation access mode, which allows node to



(a) Network with 40 nodes



(b) Network with 60 nodes

Figure 5: Delivered messages

obtain exclusive access to the medium in a contention-free manner.

5.2 Analysis on Network Self-organization

Since the self-organization capability of HCT is the key to its resource-reservation mode, in this section it is investigated the self-organization of the simulated networks. In other words, it was evaluated the suituability of HCT to deal with mobility issues. Therefore it was measured the number of cycles nodes where able to stay clusterized in the different scenarios, and also how long nodes had to wait in order to enter a new cluster.

The *clusterized rate* was calculated for each simulation run of networks with 40 and 60 nodes, as shown in figure 6. It can be seen that in networks with 40 nodes the *clusterized rate* was quite steady and did not vary significantly with speed. In this case, most of nodes could clusterize since at most 5 clusters within 2 hops, with 7 nodes each, can be formed. This way, even in scenarios with high spatial densities almost all nodes were clusterized. But in networks with 60 nodes a proportionally smaller number of nodes could clusterize in high spatial densities. The *clusterized rate* in those networks remained steady in denser scenarios, and increased as the spatial density decreased enough to allow more clusters to be formed (but restricted to the defined limit within 2 hops around each cluster).

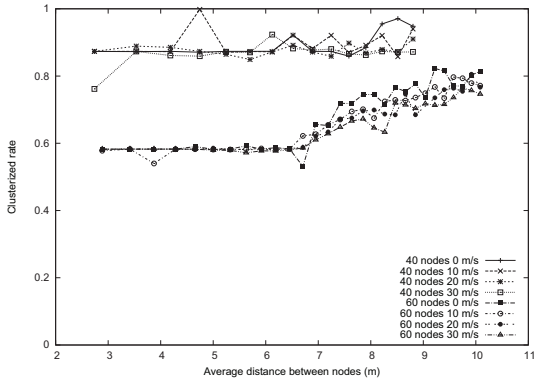


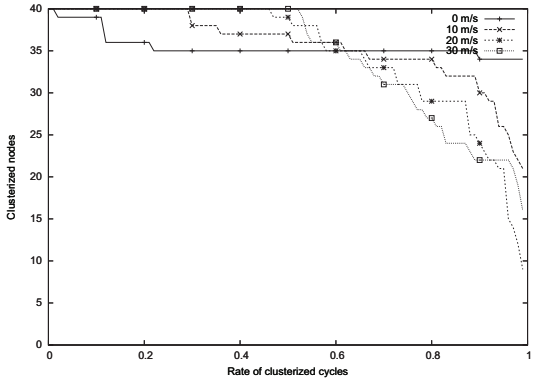
Figure 6: Clusterized rate

The *clusterized nodes* gives the number of nodes which presented at least a given number of clusterized cycles. It was calculated considering networks where the spatial density allowed a high *clusterized rate*. In networks with 40 nodes and average distance of $7.8m$ between nodes, shown in figure 7(a), the *clusterized nodes* had a clear dependence with speed. The figure shows that there was a threshold in the *clusterized rate* above which *clusterized nodes* suddenly and steadily decreased. Despite that, many nodes could stay clusterized almost all the time. When networks with 60 and average distance of 10 m between nodes were considered, the threshold in the *clusterized rate* appeared earlier and the steepness of *clusterized nodes* decay was more intense. In fact, in networks with 60 nodes few nodes were able to stay clusterized all the time. If *clusterized rate* and *clusterized nodes* reflects how many of the total transmission cycles correspond in average to clusterized cycles, it lacks the information about how long a node is expected to wait before becoming a cluster member.

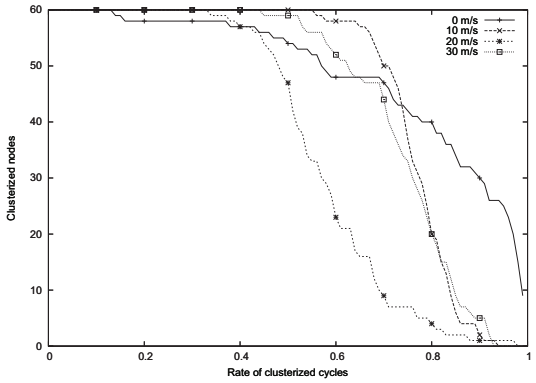
In the case of networks with 40 nodes, it can be expected a short *disconnected time*, since almost everyl node was clusterized during the simulations, as shown in figure 8(a). It corresponds to a scenario where nodes were distant each other $4.7m$ in average, and the network had a high *clusterized rate*. It can be seen that once a mobile node left a cluster, it was very likely that it entered a new cluster within $200ms$ (which corresponded to about 4 transmission cycles in the experiments). In networks with 60 nodes, with average distance of $10m$ between nodes which resulted in a reasonable *clusterized rate*, it can be expected a longer *disconnected time* as shown in figure 8(b).

6. CONCLUSIONS

This paper concerned the medium access issues in MANETs applications that rely on timely delivery of messages. It presented and discussed results obtained from an intensive simulation study that investigated the use of our previously developed coordinated MAC protocol named HCT-MAC.

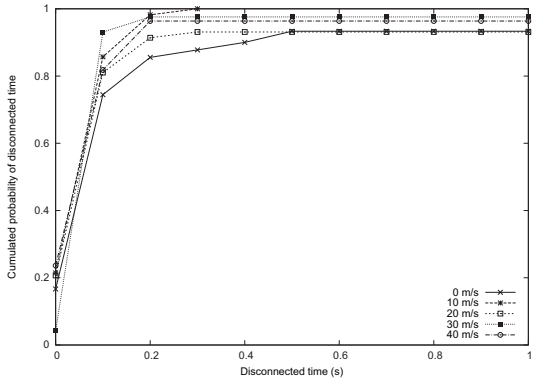


(a) Network with 40 nodes

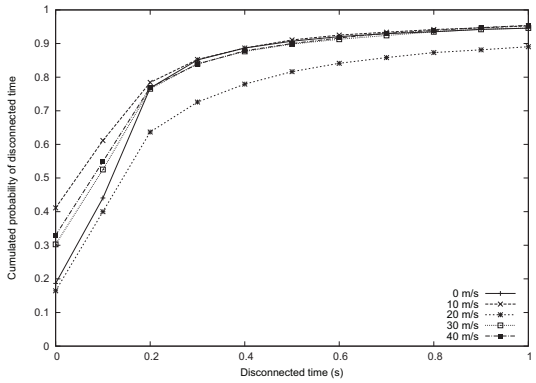


(b) Network with 60 nodes

Figure 7: Nodes with at least a given clusterized rate



(a) Network with 40 nodes



(b) Network with 60 nodes

Figure 8: Disconnected time

More specifically, the study investigated at which extent the HCT MAC protocol would improve the medium utilization and timely delivery of messages in scenarios with highly mobility of nodes. Performance comparisons against the tradiciontaly used CSMA protocol were provided, as this protocol is a *de facto* solution for medium access in MANETs.

Simulation results showed that HCT outperformed CSMA in scenarios with different network sizes, spatial densities, and speeds of nodes. Despite its overhead due to network self-organization and control frames, HCT still presented a higher medium utilization and rate of successfully delivered messages compared to CSMA. Moreover, in scenarios where HCT was able to keep almost the whole network self-organized, it approached the performance it would be expected from a TDMA-like MAC protocol. However, in networks with large spatial densities its performance was similar to CSMA.

The better performance on medium utilization and timely delivery of messages of HCT can be related to its resource-reservation access mode. That was analysed on the experiments on network self-organization, which gave the ratio of nodes which were able to clusterize and thus to operate in resource-reservation mode. These results showed that the ratio of clusterized cycles each node experienced during the experiments was related to the spatial distribution of nodes in the experiemnts, but there was no clear dependence on speeds of nodes. However, speeds influenced the time that nodes were expected to wait to become cluster members.

There still exist a number of questions regarding the performance of the HCT MAC protocol regarding the chosen metrics. It must be further clarified the the dependence of its performance on spatial distribution of nodes and mobility pattern. Therefore, a desired result is to predict its performance according to such characteristics of the network.

7. ACKNOWLEDGMENTS

Authors would like to thank the Brazilian funding agencies CAPES (grant 0616-11-7) and CNPq (grant 486250/2007-5) for their valuable support in the development of this work.

8. REFERENCES

[1] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta.

Adhoc mac: new mac architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services. *Wirel. Netw.*, 10(4):359–366, 2004.

[2] C. P. Bridges and T. Vladimirova. Agent computing applications in distributed satellite systems. In *9th International Symposium on Autonomous Decentralized Systems (ISADS 2009)*, Athens, Greece, 2009.

[3] R. Burda and C. Wietfeld. A Distributed and Autonomous Beacon Scheduling Algorithm for IEEE 802.15.4/ZigBee Mesh-Networks. In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS 2007)*, Pisa, Italy, October 2007. IEEE.

[4] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: A distributed mobile sensor computing system. In *4th ACM SenSys*, Boulder, CO, November 2006.

[5] IEEE. *802.11e: Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY); Ammendment 8: Medium Access Control (MAC) Quality of Service*. IEEE Computer Society, 3 Park Avenue, New York, NY, USA, 2005 edition, Outubro 2005.

[6] IEEE. *802.15.4: Wireless Medium Access Control (MAC) and Physical Layer(PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE Computer Society, 3 Park Avenue, New York, NY, USA, 2006 edition, October 2006.

[7] IEEE. *802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) Specifications*. IEEE Computer Society, 3 Park Avenue, New York, NY, USA, 2007 edition, Março 2007.

[8] A. Koubaa, A. Cunha, and M. Alves. A time division beacon scheduling mechanism for ieee 802.15.4/zigbee cluster-tree wireless sensor networks. In *Proceedings of the 19th Euromicro Conference on Real-Time Systems*, pages 125–135, Washington, DC, USA, 2007. IEEE Computer Society.

[9] S. Kumar, V. S. Raghavan, and J. Deng. Medium access control protocols for ad hoc wireless networks: A survey. *Ad Hoc Networks*, 4(3):326–358, 2006.

[10] P. Muthukumaran, R. de Paz Alberola, R. Spinar, and D. Pesch. Meshmac: Enabling mesh networking over ieee 802.15.4 through distributed beacon scheduling. In J. Zheng, S. Mao, S. F. Midkiff, and H. Zhu, editors, *ADHOCNETS*, volume 28 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 561–575. Springer, 2009.

[11] N. Pereira, B. Andersson, and E. Tovar. Widom: A dominance protocol for wireless medium access. *IEEE Transactions on Industrial Informatics*, 3:120–130, 2007.

[12] H. N. Pham, D. Pediaditakis, and A. Boulis. From simulation to real deployments in wsn and back. In *IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007*, pages 1–6, June 2007.

[13] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-mac: a hybrid mac for wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 90–101, New York, NY, USA, 2005. ACM Press.

[14] M. M. Sobral and L. B. Becker. A wireless hybrid contention/tdma-based mac for real-time mobile applications. In *ACM Symposium on Applied Computing 2008, Real-Time Systems Track*, Fortaleza, Brazil, March 2008.

[15] M. M. Sobral and L. B. Becker. Towards a clustering approach to support real-time communication in ad-hoc wireless networks. In *Brazilian Workshop on Real-Time Systems 2009 (WTR 2009)*, Recife, Brazil, May 2009.

[16] J. L. Sobrinho and A. S. Krishnakumar. Quality-of-service in ad hoc carrier sense multiple access wireless networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1353–1368, 1999.

[17] A. Varga. The omnet++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference*, pages 319–324, Prague, Czech Republic, June 2001. SCS – European Publishing House.

[18] J. Voelcker. Cars get street smart. *IEEE Spectrum*, 44(10):16–18, Oct. 2007.

[19] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 517–526, 2004.

The Possibility of Wireless Sensor Networks for Commercial Vehicle Load Monitoring

Jieun Jung
RFID/USN Convergence Center
KETI
Gyeonggi-do, Republic of Korea
jejung@keti.re.kr

Byunghun Song
RFID/USN Convergence Center
KETI
Gyeonggi-do, Republic of Korea
bhsong@keti.re.kr

Sooyeol Park
IT Convergence Research Center
INNOSENSING Corporation
Seoul, Republic of Korea
cto@innosensing.co.kr

ABSTRACT
It is of great importance to be able to monitor and enforce vehicle weight limits for road authorities involved in almost all aspects of transportation and pavement engineering. For the active control of additional weight carried by overloaded vehicles, essential IT technologies such as sensors, measurement, and data processing have been applied. The integration of vehicle load monitoring systems with Wireless Sensor Networks (WSN) technology has a possibility of reducing installation efforts and costs, and enabling the quick and easy configuration of data acquisition and control systems. In this paper, we try to verify that WSN technology can replace the wiring sensor applications for vehicle monitoring issues. The WSN-based system includes: inclinometer sensors which measure variation of inclination values with load changes;, an Access Point (AP) that logs the data collected from all these sensors;, and a weight estimation algorithm. To reach the goal, we performed an experiment with real deployment and estimated the weight of trucks with an error of less than 3%. The result shows that it is possible to adopt WSN for commercial vehicle load monitoring.

Categories and Subject Descriptors
C.3.3 **[Special Purpose and Application Based Systems]**: Real-time and embedded systems

General Terms
Algorithms, Measurement, Performance, Experimentation

Keywords
Commercial Vehicle, Inclinometer Sensor, Weight Estimation, Wireless Sensor Networks

1. INTRODUCTION
Recently telematic technologies used for vehicle information have provided a number of practical services, such as vehicle/traffic tracking, vehicle/road safety, and entertainment services. [1] To start with, vehicle/traffic tracking services are already deployed on a commercial scale, and drivers on the move can be informed of their location, movement, and status in real-time by using mobile systems such as navigation systems, cell phones and so on. Furthermore, electromechanical sensors embedded in vehicles, such as pressure, acceleration, and temperature sensors help drivers to maintain the safety of surrounding vehicles.

However, vehicle accidents on the road have been on the increase, especially accidents caused by huge commercial vehicles including trucks and, trailers, and these are becoming a big issue nowadays. According to an analysis of traffic accidents in the Republic of Korea in 2008, 4.7 % of vehicle accidents were caused by huge commercial vehicles, but they caused 12.5% of death from traffic accidents. This reveals that accidents involving huge commercial vehicles are more likely to cause severe results. [2]

With the development of the distribution industry, more active control of overloading vehicles also becomes necessary for the management of highways and bridges. Overloading is one of the biggest elements which have a great influence on the deterioration of this SOC (Social Overhead Capital). When vehicles containing dangerous chemicals or fire risk materials overturn, this can even bring unexpected aftereffects. Hence, it is of great importance for the maintenance and operation of the road and bridge infrastructure to monitor and prevent vehicle overloading. Vehicle information, which is the basis of the successful creation of safety traffic environments, should be both collected and analyzed.

Various studies on monitoring over-weight vehicles have shown the benefits of monitoring them. At present, all road authorities use either stationary vehicle scales along the route or Weight-In-Motion (WIM) systems at toll plazas for vehicle load enforcement, although it involves expensive installation and calibration procedures. Firstly, static weight stations are inefficient in that they force vehicles to enter the stations and wait for the process in a queue. [3] To resolve this problem, mobile weighing systems have also been introduced and installed everywhere with no limitations. However, there still require vehicles on the move to stop.

Lastly, WIM systems are designed to capture and record vehicle axle weights and gross vehicle weights when they drive over in-pavement sensors such as road cells. Unlike the above static weight stations, they do not require the subject vehicle travelling in traffic lanes to stop. However, it is still challenging to calculate weight accurately without any vehicle information such as fuel type, year, model and so on. Furthermore, WIM systems are highly sensitive to electromagnetic disturbances caused mostly by lightning strikes in the vicinity of the equipment. [4, 5]

In a different approach, weighing systems embedded in the vehicles have also been designed. [6] However, the weighing systems implemented using wired communication methods have difficulties in wiring and configuration constraints. Vehicles such as cargo trucks also have trouble disassembling part of their system because they fold their back trailers in case they are empty. Therefore, we surveyed wireless technology and found that WSN technology could be an advance in commercial vehicle overload monitoring. WSN based on inclination measurement was

designed, calibrated, and tested to examine the possibility of using WSN for commercial vehicle load monitoring.

To fulfill the requirements, we first designed a WSN-based inclinometer sensor that measured inclination values with variations of load changes. The sensor has many unique challenges: the sensor needs to be insensitive to the other sensors and nearby electrical equipment and should have a long lifetime; the sensor has to be resistant to water and temperature; and a package of sensor nodes should be designed to simply be attached and removed from the vehicles' suspensions.

Given inclination measurements from the wireless inclinometer sensors, we still need to construct a load estimation algorithm that has good performance. There is an important challenge in estimating the weight of vehicles: the values of inclinometer sensors are greatly affected when they are used in lanes on slopes or rocky roads. In this paper, we introduce an approach that handles the challenge by utilizing both reference sensors and sensors to measure changes of tandem axles. To evaluate the performance, experiments which targeted 34 to 43 ton trucks were performed in a real environment. Moreover, we estimated the weight of the trucks with an error of less than 3%.

This paper is structured as follows: In Section 2, we present the specific design and rationale of the proposed a vehicle load monitoring system based on WSN. Section 3 examines the possibility of using WSN for commercial vehicle load monitoring with real implementation. Section 4 concludes the paper.

2. WSN-BASED VEHICLE LOAD MONITORING SYSTEM

In this section, we propose a WSN-based commercial vehicle load monitoring system to solve the problem and we detail the main challenges that need to be addressed. Our main approach to monitoring the load of vehicles is to use WSN-based inclinometer sensors which estimate vehicle load changes with variations of inclination values. The proposed system shown in Fig 1 consists of three main components: inclinometer sensors attachable to suspensions, and an Access Point (AP), and a weight estimation algorithm.

The following section presents how we developed and implemented the sensor design of the wireless inclinometer sensors, including the choice of inclinometer, casing and noise filters. We then describe the transmission protocol developed for this sensor node and the weight estimation algorithm in detail.

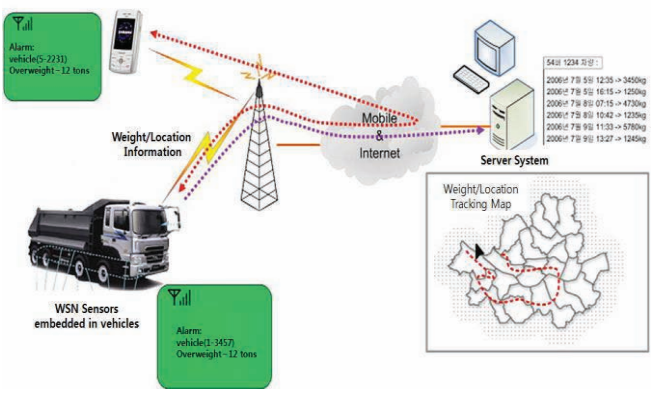


Figure 1. Semantic Diagram of WSN-based Vehicle Load Monitoring System Layout

2.1 Sensor Node Design

Table 1. Inclinometer data (SCA-61T-FAHH1G)

Parameter (units)	Performance characteristic
Sensitivity (V / g)	4
Noise Density (° / Hz)	0.0008
Current Consumption(mA)	2.4 ~ 4
Min. Operating Voltage(V)	4.75

We selected a MEMS (Micro Electro Mechanical Systems)-based single axis inclinometer, SCA-61T-FAHH1G, made by VTI Technologies, due to its low temperature dependency, high resolution, and low noise. [7] The SCA-61T-FAHH1G measures ranges of ±30° and with a resolution of 0.0025° (10 Hz BW, analog output) and other data about the inclinometer are shown in detail in Table 1. We then built an inclinometer sensor node using a small IC chip in which detection, signaling processing algorithms, and data transmission modules coexist as built-in-units. The sensor node is loaded with a low-power inclination measuring device, a microprocessor, and an RF transmitter and call signals. Data acquisition and all processes are performed by the sensor node itself. The microprocessor controls all the functions, including signal measurements, and executes the analysis algorithm. The measured inclination signal outputs and analyzed results are transmitted to a remote server through an RF transmission module, Zigbee PHY(CC2420). The device measures inclination in the given frequency ranges from inclinometer sensors in real time.

TI MSP430 is used as the main processor, and the RF transceiver consists of a four-wire SPI interface. Monitoring of the data and movement of each platform is done via RS-232 communication. The interface is made for JTAG (Joint Test Action Group) and SPI (Serial Peripheral Interface), are used for programming the board. An MFC interface is used with a two-pin connector, which is designed to act as an A/D transformation port and execute GPIO (General Purpose Input/output) through an ATmega128 setting so that the MFC interface can be used as an all-purpose interface. The Silicon Serial Number IC used for the ID of each board reads data only through a 1-wire interface. [8] Figure 2

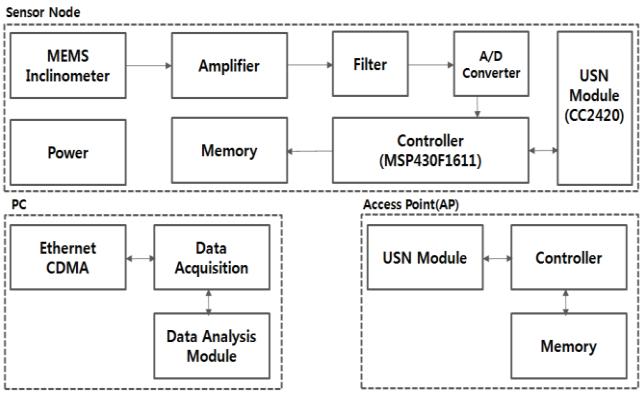


Figure 2. Block Diagram of the Inclinometer Sensor Node/AP/PC

shows a block diagram of the electronic circuit of the WSN-based inclinometer sensor node.

2.2 Transmission Protocol Design

For wireless transmission, we adapted the Zigbee and the architecture of the protocol consists of wireless sensor nodes, bridges and the AP. At the physical layer, the AP and all the nodes use an IEEE 802.15.4 compatible radio transceiver which uses the 2.4 GHz ISM band at a data transmission rate of 250 Kbits/s. The AP has both a wired and a wireless connection. The wireless connection is used to communicate with the sensor nodes while the wired connection of the AP is used to communicate with the PCs.

The MAC (Medium Access Control) layer is based on the TDMA for its reliable data transmission. The inclination data is transmitted in a series of rounds, and each round has a fixed number of packets to transfer, called the window size. For example if we have 100 packets and a window size of 5, the 100 packets are divided into 20 rounds. Only one acknowledgment is transmitted back to the sender, and lost packets in each round are retransmitted by looking at the lost packet information in the acknowledgment. Once the receiver has acquired every packet in the current round, the sender and receiver can move on to the next round. This prevents any packet collisions in the networks.

Moreover, we used a specific parameter, RF Group ID, to avoid radio frequency interference from surrounding wireless devices or systems. When data packet is transmitted, the AP will check whether it contain the same group id or not .

The transmission protocol is designed to meet both the reliable communication and power efficiency requirements. All sensor nodes can reach AP or repeater (if necessary) in one hop, and the AP with unlimited power supply can communicate with the sensor nodes in one hop. The simple architecture of this protocol is shown in Figure 3. We agreed that the exclusion of multi-hop transmission between sensor nodes can limit the network coverage by the maximum transmission range between the AP and repeater nodes. However, this network coverage is sufficient for our implementation of the vehicle load monitoring system.

We here adapted the low power listening algorithm which repeats sleep and wake period to save power consumption of sensor nodes. It enables them to stay awake for the minimum amount of time and prevent packet collisions. The procedure of our transmission protocol can be described as follows:

i. All the sensor nodes are configured with pre-assigned radio channel and a transmission time slot before installation.

ii. The AP sends a periodic synchronization message to the sensor nodes.

iii. The radio of all the synchronized sensor nodes will wake up and check if there is any transmission during its time slot, and go back to sleep if there is none.

iv. After completing data acquisition and processing, the AP transmits vehicle load information to its main server or navigation according to its data type.

2.3 Weight Estimation Algorithm

To accurately estimate the weight of vehicles, we designed a weight estimation algorithm based on the inclination measurements. We also improved the algorithm by using reference sensor nodes to correct the error of the slopes. The main principle of this algorithm is that the load of the vehicle is directly affected by the suspension which is a system of springs, shock absorbers, and linkages connecting a vehicle to its wheels. In other words, there is a correlation between changes of inclination and the vehicle-weight before and after loading. [9] Firstly, we dealt with the spring type suspensions generally installed in commercial vehicles.

Given the ADC value of sensor output SADC, weight of the empty W_e and full vehicle W_f, there exists a linear relationship between the ADC value and weight. Figure 4 shows an example of an estimating weight graph and Factor, the gradient of this graph, is $(W_f - W_e) / (ADC_f - ADC_e)$. The expected weight of the vehicle W_{ex} can be calculated by following equation (1).

$$W_{ex} = \text{Variation of } S_{ADC} \times \text{Factor} + W_e \quad (1)$$

However, this algorithm may produce incorrect results when vehicles are used in lanes on the slopes or on rocky roads. To develop an adjusted estimation algorithm, a reference sensor S1st measures the inclination of the road and a sensor S_{2nd} to measure changes of a tandem axle is additionally adopted. The expected weight of the vehicle W_{ex} for independent wheel suspension and tandem suspension can be calculated using the following equation (2).

$$W_{ex} = (\text{Variation of } S_{ADC} \pm (S_{1st\_ADC} \text{ or } S_{2nd\_ADC})) \times \text{Factor} + W_e \quad (2)$$

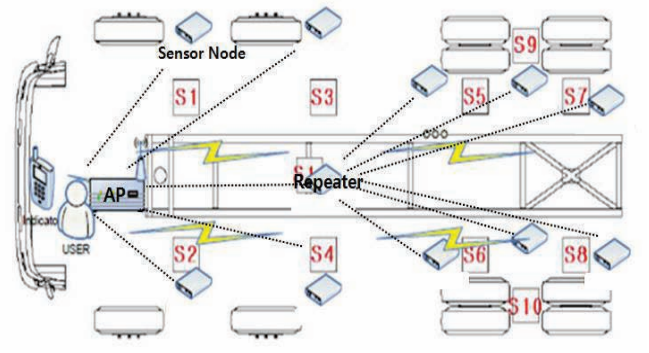


Figure 3. Communication architecture of protocol in vehicles

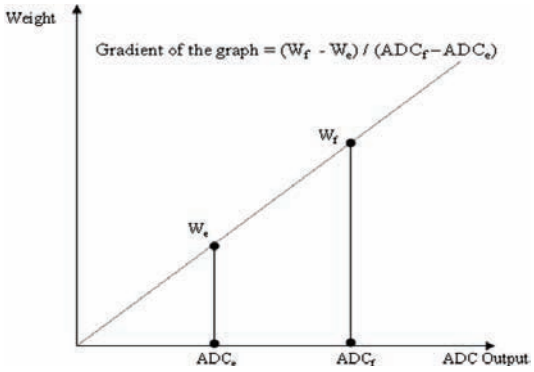


Figure 4. Example of Estimating Weight Graph

Based on an analysis of the accuracy in the experiment, we also found that duplication sensors Sd attached at both ends of an independent wheel suspension are necessary to estimate more accurate results. It often happens that the left and right sides of the suspension show different variations when they move on slopes. In this case, Factor, the gradient of this graph, is (W_f - W_e) / (Average Variation of S_{ADC} ± variation of $S_{1st\ ADC}$). The expected weight of the vehicle W_{ex} can be calculated using the following equation (3).

$$W_{ex} = (\text{Average Variation of } S_{ADC} \pm S_{1st\_ADC}) \text{ X Factor} + W_e \quad (3)$$

3. EVALUATION

In this section, we evaluated the performance of the proposed WSN monitoring system and Weight Estimation Algorithm at the test bed site. For the experiment, we installed inclinometer sensor nodes and APs in a real deployment, and chose three geographical features such as flat and rocky roads and slopes. Section A explains how we reduced the noise of the inclinometer sensor, Section B compares the performance of basic Weight Estimation Algorithm and the adjusted Weight Estimation Algorithm with additional reference and duplication sensors.

3.1 Inclinometer Sensor Performance

We measured the noise of the installed inclinometer sensor 1000 times at 100ms interval with no environmental interference (repeated 3 times). At first, the ripple measured was almost 4 ~ 5°, as shown in Table 2, since the power is not uniformly supplied. To reduce the supply noise, we applied a Regulator, LC filter, and LDO (Low Drop Output) and it shows the best result with 0.105° when both the LC filter and the LDO are adopted as shown in Table 3. However, the software should be supported for further improvement.

Table 2. Sensor output without LDO, LC Filter

	Min	Max	Ripple	Avg	Median	Mode	Var	St. Dev
1	-2.624	1.364	3.988	-0.655	-0.655	-0.909	0.394	0.628
2	-2.729	1.574	4.303	-0.632	-0.665	-0.944	0.413	0.643
3	-2.834	2.414	5.248	-0.590	-0.595	-0.665	0.533	0.730

Table 3. Sensor output after applying LDO, LC Filter

	Min	Max	Ripple	Avg	Median	Mode	Var	St. Dev
1	1.084	1.189	0.105	1.128	1.119	1.,.119	0.000	0.017
2	1.084	1.189	0.105	1.131	1.119	1.119	0.000	0.020
3	1.119	1.224	0.105	1.154	1.154	1.154	0.000	0.021

3.2 Estimation Algorithm Performance

3.2.1 Experimental Setups

We targeted 4 different trucks from 37 to 47 tons, as shown in Figure 5, and deployed inclinometer sensor nodes (ch 1 ~ ch15) and APs as presented in Figure 5. In Figure 5, ch 15 is an overall reference sensor, and ch 13 and ch 14 are tandem reference sensors. The experiment was repeated 80 times for each geographical feature; flat, rocky roads, and slopes, and the average value of the weight was used. The error is defined to be

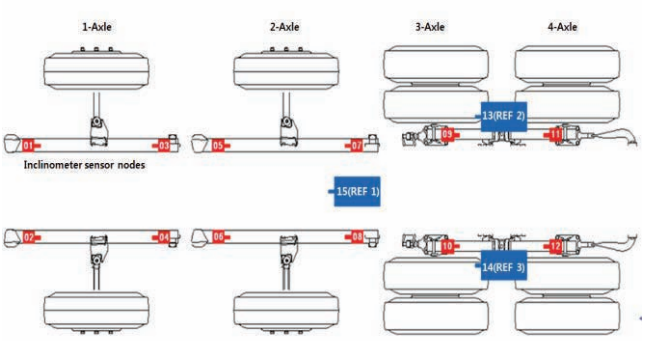


Figure 5. Experimental Setups

the difference between the weight measured by scales and the weight estimated by the basic and adjusted algorithms.

3.2.2 Result Data Analysis

We applied the basic Weight Estimation Algorithm and the adjusted algorithm using reference and duplication sensors. Table 4 summarizes the performance of the two algorithms for each geographical feature. In the case of the flat road, the two algorithms show almost the same error rate, from +1.5% to -1.6%. In the case of the slopes, we conducted the experiment on both uphill and downhill roads. The error rates of the uphill and downhill road were similar, ranging from +3.5% to -1.4 %, and the error rate was decreased by approximately 0.3% when the adjusted algorithm with the reference sensors was applied.

Finally, in the case of the rocky roads, the error rate was from +2.5% to -2.0%, and the error rate was decreased by approximately 0.3% when the adjusted algorithm with the reference and duplication sensors were applied. Through this experiment, we found that the adjusted algorithm improves the performance of the weight estimation by 0.3%, but it is still challenging to find right location and to install the additional sensors.

Table 4. Performance of weight estimation algorithm (%)

	Flat road		Slope				Rocky road	
			uphill		downhill			
	min	max	min	max	min	max	min	max
WEA	-1.6	+1.5	-2.4	+2.5	-0.8	+5.6	-2.0	+2.5
Adjusted WEA	-1.6	+1.6	-0.5	+4.2	-1.3	+4.5	-1.3	+2.9

3.2.3 Communication Performance

An experiment was carried out in order to evaluate the impact of the driving vehicle speed and vibration on the sensor node's measurements. We measured inclinometer data while the vehicle is on the move at normal speed (70 to 100km/h). The Figure 6 and Figure 7 represent the result when the vehicle moves and stops representatively. Based on the result, we confirm that all the sensor data is transmitted to the AP without loss. Only variation of magnitude is relatively high when the vehicle is moving.

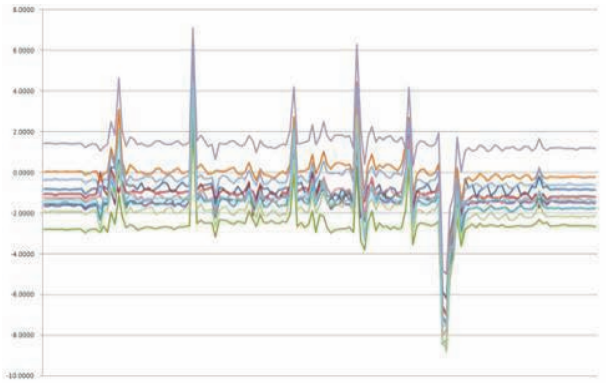


Figure 6. Inclinometer data when the vehicle stops

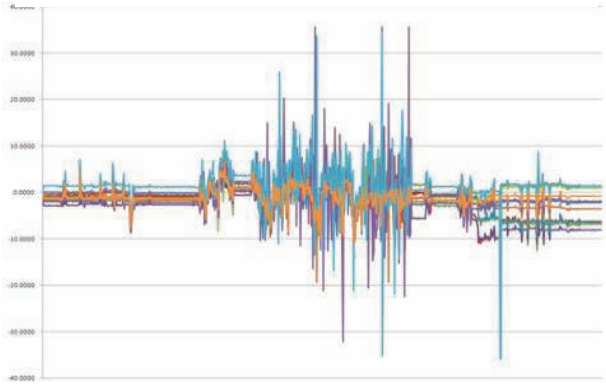


Figure 7. Inclinometer data when the vehicle is on the move

4. CONCLUSION

In this paper, we verified that WSN technology can be applied to existing wired sensor applications in heavy vehicle load monito ring. To reach the goal, we proposed a vehicle load monitoring system based on WSN, and presented experimental results obtaine d in real environments. Our main approach to monitoring the load of vehicles is to use WSN-based inclinometer sensors which estimate vehicle load changes with variation of inclination values. From the experiment, we also confirmed that the weight of the reference vehicles was estimated with an error of less than 3%. This result shows that it is possible to adopt WSN for vehicle monitoring.

However, we also found that simultaneous transmission over the various channels still remains a challenge because the amount of data was larger than the maximum channel capacity. And the number of sensors deployed in the vehicle should be minimized. To figure out this issue, additional research and experiments in real environments should be conducted to guarantee a future reliable and reasonable monitoring system based on WSN.

5. ACKNOWLEDGMENTS

This work is supported by the IT R&D program of MKE/KEIT (No. 2010-10038668) and by a grant (10CCTI-A052535-03-000000) from the Ministry of Land, Transport and Maritime of Korean government through the Core Research Institute at Seoul National University for Core Engineering Technology Development for Super Long Span Bridge R&D Center.

6. REFERENCES

[1] Roy C. Hsu., and L. R. Chen, An Integrated Embedded System Architecture for In-Vehicle Telematics and Information System, *Proc. IEEE ISIE 2005*, pp. 1409–1414, June 2005

[2] Simon Jeong, An Analytical Study of Heavy Weight Trucks' Traffic Accidents and Countermeasures:: on the Gyeong-bu Expressway, *Master thesis*, 2007

[3] Kwon Soonmin, Park Heuigu, Kim Jiwon, Kang Kyoungkoo, and Lee Dongjun, 17th ITS World Congress Busan, October 2010

[4] Qiu sheng Ru, Yang Yang, Zong qi Ning, Tao Song, The Application of Nerve Net Algorithm to reduce Vehicle Weight in Motion System Error, *International Conference on Optoelectronics and Image Processing*, vol. 2, pp. 511-514, November 2010

[5] Teerachai Deesomsuk, Tospol Pinkaew, Evaluation of effectiveness of vehicle weight estimations using bridge weight-in-motion, *The IES Journal Part A: Civil & Structural Engineering*, vol. 3, pp. 96 – 110, 2010

[6] Xiaohau Jian, Shhram H. Vaziri, Carl Haas, Leo Rothenburg, Gerhard Kennepohl, Ralph Hass, Improvements in Piezoelectric Sensors and WIM Data Collection Technology, *Auunal Conference & Exhibition of Transport Association of Canada*, 2009

[7] http://www.vti.fi/midcom-serveattachmentguid-2b9fadebe6fa4a0c24f144cd55fda22d/SCA61T_inclinometer_datasheet_8261900A.pdf

[8] Sukwon choi, Jieun Jung, Byunghun Song, and Sooyel Park, Possibility of Wireless Sensor Networks for Outside Exposed Gas Pipeline Monitoring, *International Conference on Information Processing in Sensor Networks(IPSN) RealWIN Workshop*, 2011

[1] Nicola Amati, Andrea Festini, Luigi Pelizza, Andrea Tonoli, Dynamic modeling and experimental validation of three wheeled titling vehicles, *International Journal of Vehicle Mechanics and Mobility Special Issu*e, vol. 49, 2011

Real-time routing and retry strategies for low-latency 802.15.4 control networks

Koen Holtman
Philips Research
High Tech Campus 34
5656AE Eindhoven, the Netherlands
+31 40 27 91461
Koen.Holtman@philips.com

Peter van der Stok
Philips Research
High Tech Campus 34
5656AE Eindhoven, the Netherlands
+31 40 27 49657
Peter.van.der.Stok@philips.com

ABSTRACT

In many applications of wireless control networks, the latency of message delivery is an important consideration. In a lighting control network where a light switch sends a wireless message to a lamp, a worst case end-to-end latency of 200 ms or better is desired, so that the working of the switch feels 'immediate' to the end user. This paper studies the probability that latency deadlines of a few hundred ms are exceeded. We use a 802.15.4 test network, located in a real-life office environment, to evaluate and compare the effects of several re-try and re-routing strategies and different MAC parameter settings. Testing under realistic conditions, in an office environment when people are present, is important to accurately determine worst case latency performance as experienced by end users. At night, without any people in the building, performance is much better than during the day. In order to accurately observe the effect of different strategies, test runs lasting at least a week are needed. We find that retrying message delivery via a single delivery route is sub-optimal. Keeping a set of two or more candidate routes for subsequent re-tries greatly improves worst-case latency. We show that the use of time slotting and energy saving strategies is not necessarily incompatible with the goal of optimizing for human-observable latency.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design]:** Wireless communication, C.2.2 **[Network Protocols]**: Routing protocols

General Terms

Measurement, Performance, Reliability, Experimentation, Human Factors.

Keywords

Control networks, wireless sensor networks, 802.15.4, latency, retry strategies, routing strategies, energy saving.

1. INTRODUCTION

A wireless control network differs from the more commonly considered wireless sensor network (WSN) because it incorporates actuators in addition to sensors. Whereas in most WSN designs, the object is to get all sensing data forwarded to a central (logging) location for later analysis, in a wireless control network the object is to control actuators based on data from close-by sensors. Home and office control systems are important applications of wireless control networks. The low cost of wireless sensor nodes, and their ability to run on batteries, (or even on energy harvested from the environment), makes it feasible to equip rooms with many sensors, which can be used to increase both comfort and energy efficiency.

The reliability and latency of a wireless building control system must be just as good as that of the wired system that it replaces. Consider the case where a lamp in a room is wirelessly connected to a switch on the wall. In such a setup, it is not the average end-to-end latency that will be important for the quality perception of the user, but the worst-case latency. If the latency is too high too often, the system will break the 'immediacy' mental model that the user has for light switches, which negatively impacts the user experience and ultimately the user acceptance. Our design goal is to minimize the probability that 200 ms latency is exceeded. As a rule of thumb in user interface design, 200 ms latency is still short enough that the user can accept the relation between stimulus and response as 'immediate'.

Contrary to most publications we are less concerned with the maximum achievable throughput, average-case latency, or scalability of routing algorithms. We expect that most communications in building control will not need more than 2 hops, with 4 hops being exceptional. We therefore focus on quality improvement of the 1-hop and 2-hop cases, which represent the bulk of the wireless control communication in the building.

We concentrate on the effects of multipath fading which lead to unexpected link failures of very good links during the day time. Effects coming from interference caused by the high density of nodes are not investigated. Therefore, the measurements concentrate on a relatively low number of nodes reflecting the node and message density we expect for building control

2. MEASUREMENTS IN LITERATURE

This paper is motivated by measured communication behaviour in buildings with occupants. In the literature measurements are described related to: Point to point communication, Sharing the medium between multiple senders, and Routing over a large dynamic network. This section summarizes the main published results as introduction to our routing suggestions to improve the probability that packets arrive in time (within their deadline).

A naïve model of transmission by an omni-directional antenna in empty space, states that the strength of the signal decreases with the square of the distance [9]. Unfortunately, space is not empty but is populated with reflectors and absorbers of the RF signal. Reflections from surfaces can meet the original signal with a slightly different phase thus reducing or completely removing the signal. Report [9] shows that most simple propagation models used in simulations do not correctly represent the transmission as measured in situ. Measurements of the communication quality between a single sender and a receiver indicate that there are usually three regions: (1) a clear region with little or no reception losses, (2) a transitional region where packet loss ranges from a few percent to complete loss unrelated to the transmission distance, and (3) a region where almost no packets are received.

Over time the link quality fluctuates as well. These phenomena are reported in [1][2]. The clear region for IEEE 802.15.4 [3] is reported to be on average between 3 and 15 meters in [5][6][8], where in some cases the signal had to penetrate office walls. In [7] it is observed that the height of the sender has a significant impact on packet yield. Papers [1], [4], [5] and [7] show that RSSI is not a good indicator for success rate. Link quality Indicator (LQI) is seen to perform better in [7]. From the papers we learn that the transmission range depends on the direction, that channels are not necessarily symmetric, and that channel strengths change over time, and with the height of node. Assuming that the point to point transmission is in the clear region, dependent on the load and the number of load generating senders, the medium throughput around a given node still suffers from multi-sender interference. In [4], confirmed by [6] it is shown that the total throughput obtained with one sender is halved when four senders try to occupy the channel. The lowered throughput can come from the many retries, the larger back-off times associated with retries, or the loss of packets.

The behaviour of the links over time is discussed in [10]. Conclusions are that the number of packets needed for successful transmission is a better quality indication than Reception Rate (RR). Another conclusion is that at a given time a stable link is the best link to use and that failure over a stable link is accompanied by failures over the unreliable links as well. In [2] and [12] it is shown that link quality and not hop count should be at the basis of path selection. In [12] also the unpredictable stability over the day is shown. In contrast to earlier papers, the measurements in [13] suggest that RSSI is a good indicator of link quality. The authors conclude that the new chip technology of the CC2420 improved the usability of the RSSI value. In [14] it is shown that irregularity of the radio signal has a high impact on the routing efficiency. In [11] and [13] an overview of wireless communication link-failure over time is presented.

The notion of real-time deadline and delay is not frequently cited in the context of routing. The authors of [15] show that no guarantees on end-to-end delays can be given but that we are confronted with an end-to-end delay probability distribution. This notion has led to the development of the SPEED protocol where the probability of meeting a deadline was recalculated during the progress of the packet over the multi-hop path [16]. The SWR protocol [17] uses multiple paths routing while removing packets which will not meet their deadline. Another probabilistic real-time routing protocol is presented in [18], where the forwarding time is estimated for each hop during transmission.

In this paper we investigate link performance over at least a week and try different routing and packet retry schemes to improve the transmission success rate within the deadline. This paper extends existing work by:

- Observations over periods of at least a week,
- Testing links that are in the clear region,
- Observation in an office building during working hours,
- Concentrating on one-hop and two-hop routes.

3. TEST NETWORK

Building on our earlier work [19][21], we constructed a test network with 8 network nodes in an office building, and did extensive measurements on latency. Walls are made of plasterboard, but the offices contain many large metal filing cabinets, creating a strong multi-path environment. Node locations for the tests are shown in figure 1. Each node 1-7 is a sensor node, sending messages with (arbitrary) sensing data to node 0, at random times with an average rate of 2.7 messages per second per sensor node. This high message rate is unrealistic for a control network in real-life situations, but it does ensure that we can gather sufficient statistics in a test run lasting about a week. As shown in figure 1, the nodes 0-4 are all in the same office. We located them behind obstacles in this office in such a way that there is no direct line of sight between them – this means that multipath cancellation (Rician or Rayleigh fading) can cause link failure even inside this office.

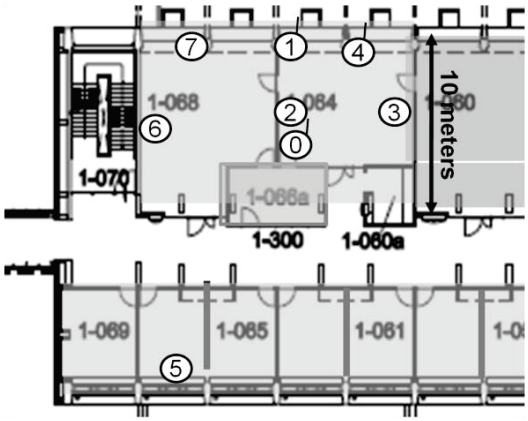


Figure 1. Sensor nodes 1-7 send messages to node 0.

The nodes use the 802.15.4 protocol in the 2.4 Ghz band, and are implemented using Jennic JN5139-Z01-M00/M001 wireless modules [20] with custom software implementing the delivery algorithm in figure 2. Nodes 0,1, and 4 have whip antennas, the other nodes have small ceramic antennas. All nodes can communicate directly with node 0, although the link quality between node 5 and node 0 proved to be lower than between the other nodes and node 0.

The messages are fixed-length and very small. Including protocol overheads, an 802.15.4 packet carrying a message has a length of 27 bytes. We use the 802.15.4 MAC uni-cast mechanism with CSMA/CA and acknowledgments, with each node doing up to SMRT ('single MAC invocation retries') packet sending re-tries whenever it finds a clear channel. The parameter SMRT is set to 4 (the 802.15.4 MAC default) in most of our measurements. In an outer loop, the MAC is invoked every WT ('wait time') milliseconds until an end-to-end acknowledge message (sent by node 0 via the reverse route) is received. Reception time of the

end-to-end ack determines the latency of the message. The parameter WT is set to 40 ms in most of the measurements.

Routes can contain multiple hops. If a node receives a message that it should forward, according to the routing instructions in the message, the node will invoke IEEE_802.15.4_MAC_unicast_with_ack() once to do the forwarding. If this fails, the routing node will discard the message. The original node will do a retry, possibly via another router, when its timer t2 runs out.

```
deliver_message(m) {
    start timer t1 counting up from 0 ms;
 do {
    pick a delivery route r;
    format message m with route r into packet p;
    start timer t2 counting down from WT ms;
    IEEE_802.15.4_MAC_unicast_with_ack(p);
    wait_until( (t2 < 0) ||
        (end-to-end acknowledgement received from MAC) );
 } while(no end-to-end acknowledgement received);
   measured_message_latency = t1;}

IEEE_802.15.4_MAC_unicast_with_ack(p) {
  Try to detect a clear channel, with exponential backoff;
    // we use MaxCSMABackoffs=4 and minBE=1 so this
    // detection phase takes between 1 and 19 ms.
  if(no clear channel found) return;
  for(i=0; i<SMRT; i++) {
    use radio to send packet p; //takes ~2 ms
    use radio to listen for MAC acknowledgement
            packet; // takes ~1 ms
    if(valid MAC acknowledgement packet heard) break;
 } }
```

Figure 2. Message delivery algorithm used by nodes.

Several variants of the algorithm in figure 2 are possible. For example, if the MAC fails to deliver a message with acknowledgment, the outer loop might retry immediately, rather than waiting for t2 to run out. We have not (yet) tested this variant – it creates a larger risk of packet storms in the system, but the speedup might be worth-while.

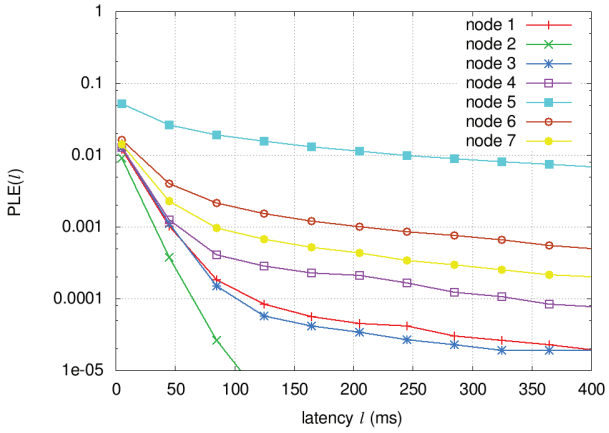


Figure 3. PLE(*l*) for all nodes with 1-hop route to node 0.

4. SINGLE HOP LATENCY

We consider the latency of routing along a single hop: we do a test where all sensor nodes always choose the direct route to node 0 to deliver their message with WT=40 ms and SMRT=4. In order to get sufficient statistics about the infrequent high latencies, we run the test network for many days. This creates test logs containing millions of message latency measurements per node. To interpret the test run data, with a focus on the question of how the end user will experience the latency, we found it instructive to define the metric PLE(*l*): the probability that a latency *l* is exceeded by a message. We compute it as follows from a test run:

PLE(*l*) for node *n* =

(number of messages sent during office hours by node *n* with measured_message_latency >=*l*) / (total number of messages sent during office hours by node *n*).

Because of our interest in human-observable latency, the PLE calculation does not consider messages sent outside of office hours – this is discussed in more detail in [19]. Figure 3 plots PLE(*l*) for all sensor nodes, showing large differences for the nodes. Node 2, closest to node 0, has the best (lowest) curve. Node 5, furthest from node 0, has the worst. For the other nodes, there is no strong correlation between curve position and the distance of the node *n* to node 0.

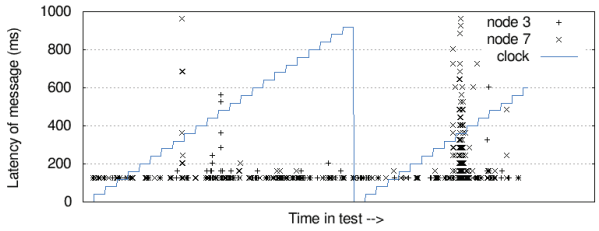


Figure 4. Messages with latency higher than 125 ms.

Figure 4 shows how high latencies occur over time, for nodes 3 and 7. The X axis is time over a 38 hour section of the run taken for figure 3, the saw-tooth line plots a 24-hour clock. The high latencies occur almost exclusively during office hours. Node 7 experienced latencies higher than 1000 ms off the Y axis scale. We see that a node can have 'good days' where a given latency deadline is never, or almost never, exceeded, and 'bad days' where it is exceeded often within a few hours. Test runs have to be long because every node needs to have a chance to experience its 'bad days' at the rate that they happen for that node. The unpredictability of the number of 'bad days' in a week, and the possibility of long term changes in the environment, makes us cautious in comparing curves from different test runs. To measure the differences between alternative message delivery strategies, we use a single long test run in which the nodes switch to a random different strategy every 10 minutes. Our testing approach extends earlier work, where conclusions were drawn on the basis of shorter run durations under cleaner conditions.

5. MAC INVOCATION INTERVAL WT

With WT=40 ms, a failing link makes the node invoke the MAC, every 40 ms, sending at most SMRT=4 packets. So if a link is failing, WT=40 ms generally leads to a PHY packet sending rate of 100 packets per second, until success. This is an excessively

high rate which may lead to packet storms. In [21], we used a 5-node test-bed (with SMRT=4) to investigate the effect of varying WT. Some indicative results from [21] are shown in figure 5. For node 4, far away from node 0, we see an improvement when lowering WT. This is conform expectation, because the time between retries is smaller and consequently the probability of meeting the deadline for a given number of retries is higher.

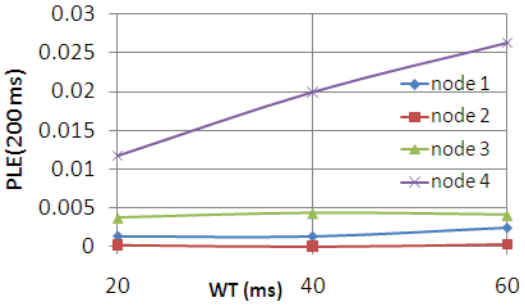


Figure 5. Effect of WT on PLE(200ms) for 4 nodes on different test bed, with 1-hop routing to node 0.

However, for the other nodes, closer to node 0, we see no difference that is statistically significant for a one-week test run. Apparently, with values of 66, 100, and 200 PHY packets per second, we are well past the point of increasing returns for these short hops. This can be explained as follows. First, if a link suffers from significant path fading for a duration much larger than the packet size, sending packets faster will not help. Second, the positive effect of sending more packets might be counterbalanced by the negative effects of longer medium occupation. Third, while the MAC is waiting for a clear channel, its 802.15.4 radio will not be able to receive an end-to-end acknowledge message directed to it. Instead, the MAC will interpret the attempt to deliver the message as a busy channel, causing it to wait. Finding out which of these effects, if any, is dominant needs more study.

In a similar test run on the 8-node test-bed of figure 1, we again saw no significant difference, for PLE(*l*) with l>100 ms, between WT=20 and WT=40. For all test results in sections 6-9 we therefore use WT=40 ms and SMRT=4.

6. IMPROVEMENT BY ADDING MORE CANDIDATE ROUTES

Except for nodes 5 and 6, the curves in figure 3 are satisfactory, from the standpoint of human-observable latency. Nevertheless, an optimization that makes the curves go down more steeply, increases the quality of the user experience. As we discussed more extensively in [19], a lot of the high latencies in figure 3 can be attributed to multipath fading of the single link. Performance can be improved if the sensor node maintains a list of multiple candidate routes, and re-tries along other routes if attempts along the preferred route fail. Figure 6 shows the performance of the test network with an improved routing scheme. Each node first tries the direct route to node 0 twice. If that does not work, routing via node 1 is tried, and if that does not work routing via node 2. If that still fails, the direct route is tried twice again, and so on. Values of PLE(200 ms) are greatly improved (reduced) except for node 2, which goes from extremely good to merely good.

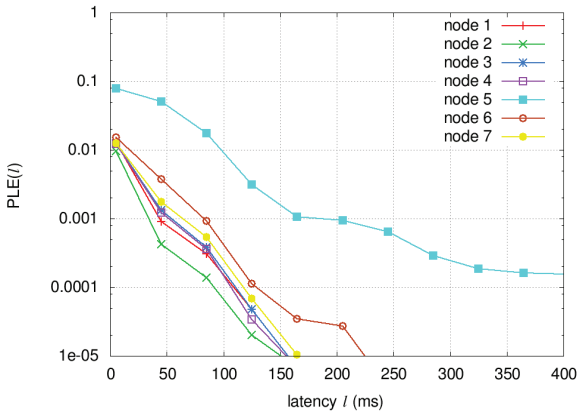


Figure 6. PLE(*l*) for all nodes using 3 candidate routes.

7. TWO-HOP LATENCY

We now turn to the study of two-hop routes. We measure the PLE for message delivery along each of the all possible 1-hop and 2-hop routes to node 0. In this test run we decrease the message sending rate in the system to 1.4 messages per node per second, to avoid a situation where system self-interference becomes a dominant factor in determining PLE. Each route is tested individually: the node keeps trying the route under test until the message is delivered. Over the course of 2 working days, each route was tested by sending 12500 messages over it on average. This number was sufficient to determine PLE(*l*) up to *l*=165 ms with reasonable statistical accuracy. A longer test run might show a different picture with PLE scores that are sometimes lower, because nodes have a bigger chance of experiencing a 'bad day'.

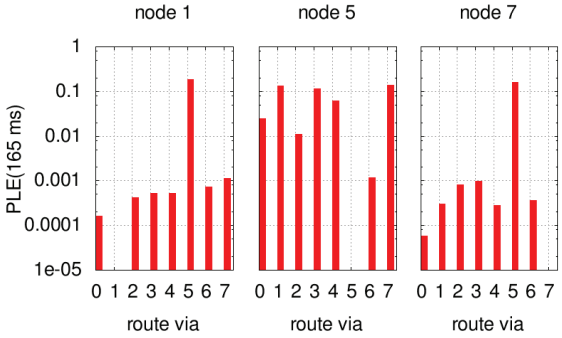


Figure 7. PLE(165 ms) for all 1-hop and 2-hop routes starting at nodes 1, 5 and 7.

Figure 7 shows some representative measurement results. As in figure 3, we see a spread in route quality. For all nodes but node 5, the 1-hop route directly to node 0 (the bar labelled 0 in the graphs) out-performs all 2-hop routes by a significant margin.

8. RSSI USAGE

As it takes a long time to measure PLE curves accurately, an obvious question is whether PLE could be predicted based on RSSI given the contradictory results from the literature presented in section 2. In the same test run we also measured the average RSSI value of each link. Also we devised a measure to combine the RSSI values of the two links involved in a two hop path

(called the route RSSI). In figure 8 we plot the relation between PLE(165 ms) and the route RSSI. The route RSSI for a 2-hop route is computed by multiplying the RSSI values of the two hops. We also tried addition and taking the minimum, but found that multiplication gave the best result in terms of PLE correlation. Looking at the 2-hop routes only, we see that a very low RSSI, RSSI <20, is a good predictor for a bad route. For RSSI>20 however, there is no longer any visible correlation between RSSI and route quality. Apparently, in this region, the link budget along the route is so high that, with the number of retries we do, it stops being the dominant mechanism in determining PLE. Therefore, if we are to find the best 2-hop routes with a high probability, we have to measure their actual PLE, though an RSSI cut-off at 20 can be used to reduce the number of routes to be measured.

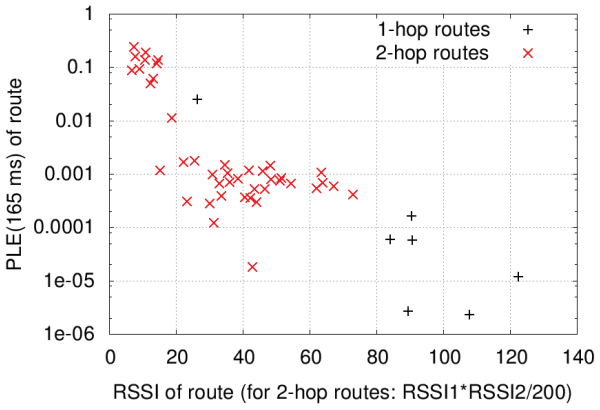


Figure 8. PLE(165 ms*)* versus route RSSI in test network.

Another technique to save resources in selecting routes is to base selection on PLE(*l*) measurements with *l*<165 ms, which can be measured more quickly. We have observed that the PLE curves hardly cross in the region from 85 to 165. Therefore using PLE(85 ms) instead of PLE(165 ms) could lead to nearly as good route selection results, while measurement times are shorter.

9. NUMBER OF CANDIDATE ROUTES

We now turn to the question of how many candidate routes a node should maintain in order to achieve the best possible PLE. As more candidate routes are added, especially routes with an individual PLE much worse than the PLE of the best candidate route, we can expect diminishing returns, or even a worsening of the PLE.

First, we run a one-week test comparing the performance of having *c* candidate routes in a node, with *c* in the range 1-4. The *c* candidate routes used by a node are always the *c* best (lowest PLE) routes identified in the test of section 7. A node first tries the best candidate route twice, and then tries the other candidate routes in from best-to-worst order. If the message is still not delivered, it tries the best route twice again, etc. The result of this test is that having multiple candidate routes outperforms having only one candidate route. However, after sending 100.000 test messages for each *c* for each node, we find no statistically significant difference in the PLE(200 ms) for 2, 3, and 4 candidate routes. In practical terms, having 2 candidate routes is as good as having 4 in this test.

In a second one-week test, we studied the performance of multiple candidate routes if all candidate routes are 'long' 2-hop routes. We eliminated the direct route to node 0, and the route via node 2 which has one very short hop, as candidates, and ran the test again with the remaining best routes. Ignoring inter-node interference, the results are therefore somewhat indicative of the PLE that can be expected in a larger test network, for nodes that are too far away from node 0 to reach it with a single hop. Figure 9 shows the test results. For node 5, we see a clear improvement in PLE(200 ms) when more candidate routes are added. For all other nodes, these is an improvement when going from 1 candidate route to multiple routes, but again no statistically significant differences for 2, 3, and 4 candidate routes. Figure 10 shows averages for the test results in figure 9.

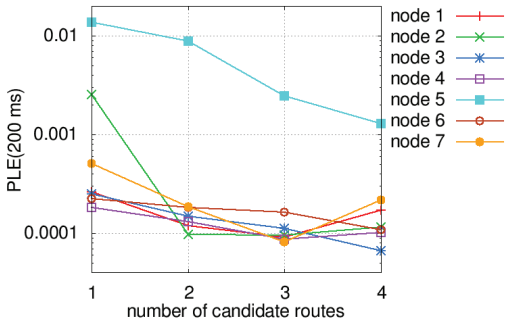


Figure 9. PLE(200 ms) for 'long' 2-hop routes only.

We conclude that in a dense network, for example the network of figure 1 but with node 5 excluded, keeping 2 candidate routes will be sufficient, and this can keep routing tables small. A more general networking solution should however have the ability to maintain more than 2 candidate routes, to optimize the PLE for relatively isolated nodes like node 5.

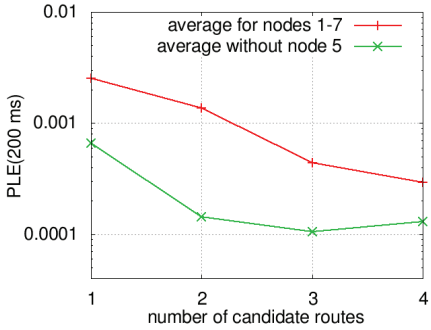


Figure 10. Average PLE(200 ms) for 'long' 2-hop routes.

For routing via long 2-hop routes only, figure 9 shows a best-case PLE(200 ms) of about 0.0001, which is high compared to the PLE(200 ms) values found in the test of figure 3, where direct routing to node 0 is allowed. Apparently, another factor than multipath fading is dominant in determining the value of the best case PLE(200 ms). It might be possible to lower the influence of this factor by re-tuning the system, for example by changing WT, decreasing the message sending rate per node (which lowers system self-interference), or changing the MAC parameters. This is a topic for further study.

10. PLE IN TIME SLOTTED NETWORKS

The use of a time-slotted 802.15.4 MAC, instead of CSMA, can be beneficial to latency in networks that need to handle high

traffic loads [22], because it avoids collisions and hidden node problems. In time slotted networks, a sensor node is constrained in the opportunities it has to use the channel for retries: so the expected PLE curves are different from the ones shown above, where nodes will use the channel with a very high duty cycle when retrying. Furthermore, time slotted networks get more efficient at carrying high traffic loads if slots are kept small, with less reserved time in a slot for MAC retries [22], so it is preferable in time slotted networks to run with a low setting for the MAC retry parameter, a low SMRT. It is interesting to study what happens to PLE if a test network with the algorithm of figure 2 is run with a lower SMRT, approximating a time slotted network.

Experiments were done on the test network shown in figure 11. This test network used the same node hardware as in figure 1, but was built in a different location in the same building. (The location of figure 1 had been refurbished, removing most metal closets, preventing us from reproducing the setup of figure 1.)

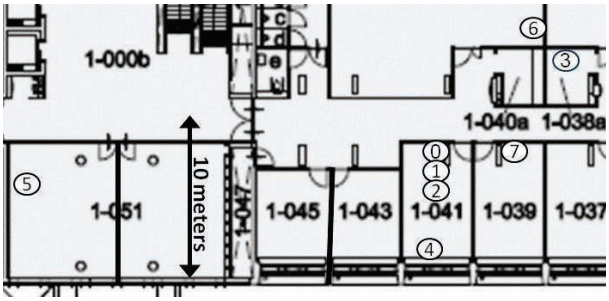


Figure 11. New test network. Sensor nodes 1-7 send messages to node 0.

Figure 12 shows the effect of different SMRT values on the PLE, with WT=40 ms, during a 3-week test. Each node does all delivery attempts over a 1-hop route to node 0, and node 0 always uses SMRT=4 when sending back its end-to-end acknowledge packet.

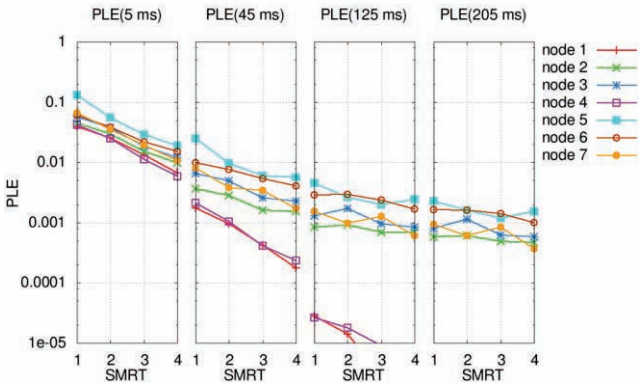


Figure 12. PLE values when routing directly to node 0, for different SMRT values (=number of MAC retries done in each delivery attempt).

Leftmost in figure 12, the PLE(5 ms) value can also be interpreted as the probability that the message is not delivered after a single MAC invocation. As expected, with a larger SMRT the probability of non-delivery is lower. Less intuitively, the graphs for PLE(125 ms) and PLE(205 ms) do not show any statistically significant effect of SMRT on the probability of delivering the message within the deadline. So, under the low traffic loads in our test network, when we go from SMRT=4 down to SMRT=1, operating on the channel like a time-slotted protocol with a 40 ms cycle time, PLE(205 ms) is hardly affected: the lower node channel occupancy associated with time slotting does not have a negative effect. For a higher network workload, lowering SMRT in a time slotted network is expected to improve worst-case latency [22]: a lower SMRT means shorter slots, so each node actually has more frequent opportunities to retry and empty its re-send queues. Overall, these results indicate that moving from CSMA to time-slotting with a fast cycle time is not incompatible with optimising for human-observable PLE.

Looking overall at the performance of the network nodes, we see that PLE(205 ms) for node 1 and 4 are so low that they are off the charts. All other nodes have a PLE(205 ms) around 0.0001, with much less spread than in the measurements done on the first test network (figure 3). The bad performance of node 2 in this test is somewhat surprising, given its location. In later tests with this network, node 2 performed better, so apparently it was just very 'unlucky' in the 3 weeks of this test.

11. ENERGY SCAVENGING SENSOR NODES

Given the positive result of the previous section, showing that a lower channel access rate is not incompatible with optimising PLE, the topic of energy scavenging nodes comes in scope. An energy scavenging sensor node is not powered by a battery or mains power connection, but extracts the energy it needs from its environment [23]. Consider an energy scavenging sensor node that uses a small solar cell to (re)charge a capacitor. Whenever a message needs to be sent, the radio will have to be used intelligently and sparingly, to minimise the probability of non delivery before the capacitor runs out of charge. It is not always possible or economical to put a very large capacitor in a low duty cycle energy scavenging sensor, to store a large reserve of energy just in case. Larger capacitors have larger leak currents: they require larger energy scavenging mechanisms just to keep them charged fully. Thus, when considering if an energy scavenging node using a capacitor of a certain size can be an acceptable control network product to an end user, first of all, we have to consider the probability that the node will not deliver the control message at all.

We used the testbed of figure 11 to study this probability. We ran a 3-week test, with all sensor nodes trying to send directly to node 0 (using no alternate routes), using different waiting times WT between sends, and computing from the test logs the probability of non-delivery after N MAC invocations. We use SMRT=1, to keep the energy used per MAC invocation as low as possible. This also allows us to refer to one MAC invocation as being 'one send' in the graphs and discussion below. Following the algorithm in figure 2, our sensor nodes use carrier sense to try to detect a clear channel before sending a message, whereas a typical energy scavenging node will omit the carrier sense as it uses a lot of energy. The test results, in figure 13, are therefore most representative of an energy scavenging network with a very low channel duty cycle.

Figure 13 clearly shows that reliability can be improved not just by sending more often, but also by waiting longer between sends. The nodes 1, 2, and 4, all very close to node 0, performed very well in this test run, with a failure probability so low that it is off the charts as N grows larger. The effect of increasing WT on reliability is large: if latency is not a concern (e.g. for a type of sensor node where the user does not notice or care about a long

latency) the best strategy for the node is to use a WT of 500 ms or even larger. However, for many types of energy scavenging nodes we would like to optimise both the probability of delivery and the PLE(200 ms).

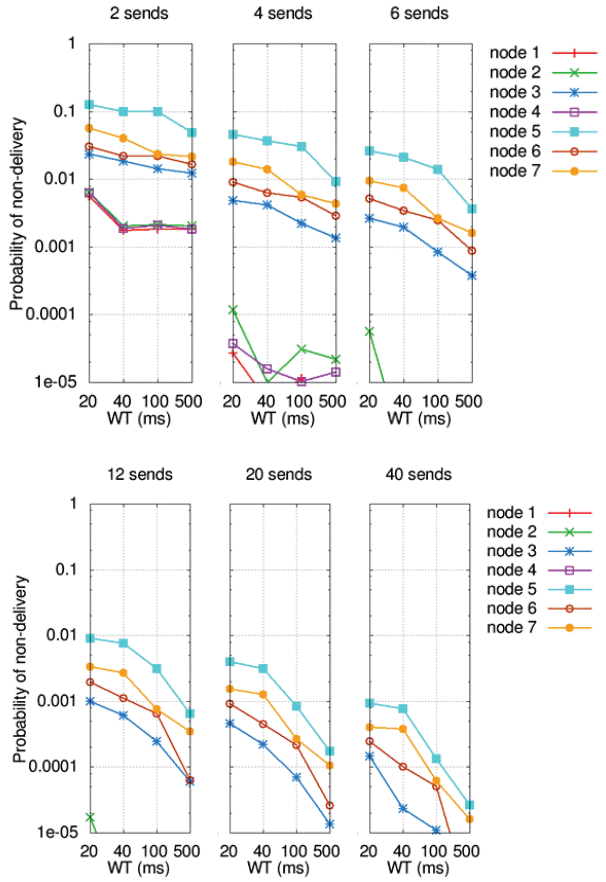


Figure 13. Probability of message non-delivery within N sends by an energy scavenging node, with different waiting times WT between sends.

12. ENERGY SCAVENGING NODES AND MULTIPLE CANDIDATE ROUTES

The results in sections 6 and 9 predict that an energy scavenging node might benefit from using multiple candidate routes when trying to deliver its message. We tested this prediction, with up to 3 candidate routes, as follows. We moved node 5 in figure 11 to office 1-041, locating it to the right of node 4, and then programmed node 1 and 5 to act as routers, with SMRT=4. All other nodes are set up to behave as energy scavenging nodes: we set SMRT=1, and configured them to use up to 3 routes: the direct route to node 0, the route via 1 to 0, and the route via 5 to 0. The nodes use WT=120 ms when trying via 1 route, WT=60 ms when trying via 2 routes, and WT=40 ms when trying via 3 routes – so each route is always tried once every 120 ms. Test results are shown in figure 14.

Leftmost in figure 14, when 3 sends are done, the effects of having multiple candidate routes are somewhat mixed. More significantly however, with 6 or more sends the use of multiple candidate routes significantly improves the overall network performance. For 6 sends, we have a satisfying result in that the probability of non-delivery is lower than 0.001 for all nodes, something that was not achieved after 6 sends in figure 13, even not with WT=500 ms. The use of multiple candidate routes also optimises PLE. In this test, when using 3 candidate routes with WT=40 ms, 6 sends are done within 200 ms, so with 3 candidate routes we obtained a PLE(205 ms)<0.001 for all nodes. It is clear that using multiple candidate routes is a very useful technique to increase the reliability of energy scavenging sensor nodes, especially if low latency is desired too.

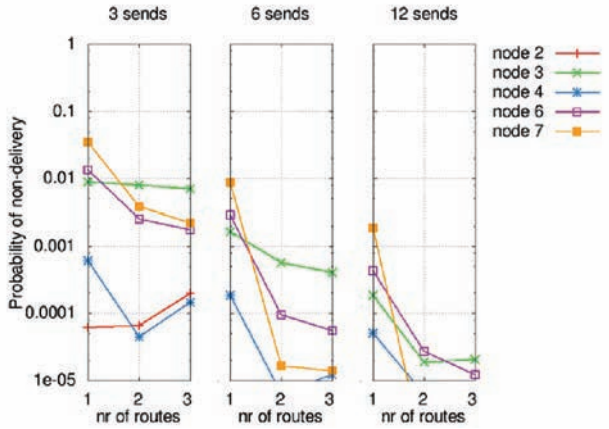


Figure 14. Probability of message non-delivery with different numbers of candidate routes and SMRT=1.

13. CONCLUSIONS

To make 802.15.4 control networks acceptable as a replacement for wired control, we must look into the way that end users experience the latency characteristics of the network. Based on this consideration, we have introduced a latency measure PLE, and optimized PLE(200 ms) based on measurements done during working hours in an office environment. Test runs lasting at least a week were done with most links in the clear region. The test results indicate that each node should maintain a list of multiple candidate routes that it can use to deliver a message. We have shown in [21] that candidate route information can be created and stored in advance. Only infrequent updates are necessary to adapt to changes that occur over timescales of weeks.

Compared to most WSN test networks in literature, our test network has a higher node density, as we expect multiple wireless sensors and actuators per room. This leads to a system where, if the retry strategy in the protocol is designed well, low link quality is no longer a dominant cause of latency deadlines being exceeded. Instead, for PLE(200 ms), fluctuating multi-path cancellation becomes a dominant cause [21], which can be eliminated by using multiple candidate routes. When this cause is eliminated, the presence or absence of a clear-region 1-hop link to the destination becomes a major determinant. We have some tentative evidence that system self-interference will become a significant determinant at some point above an average rate of 40 packets per second in the network, not counting MAC acknowledge packets. This high average packet rate is unrealistic in a small building control network, but might be approached in a large control network that needs to use multi-hop routes often to deliver messages. It is likely that the introduction of an

exponential back-off in the invoking the MAC will improve worst-case latency under higher network loads.

While our test network uses CSMA, our measurements predict that time slotted networks with a short cycle time are not necessarily at a disadvantage when it comes to achieving a good PLE(200 ms).

Finally we have looked at the problems faced by energy scavenging nodes based on capacitors, that can only do a few retries before their energy runs out. We have measured for such nodes, across the parameter space of capacitor size and retry speed, the expected reliability of message delivery over a single hop fixed route. We then show that the technique of using multiple candidate routes can be very useful to optimize the reliability of energy scavenging nodes, while simultaneously achieving a good PLE(200 ms).

There are still many unanswered questions related to the PLE(200 ms) quality metric. In particular, the problem of predicting or optimizing PLE under high network loads has not yet been addressed.

14. REFERENCES

[1] J. Zhao, and R. Govindan, *Understanding Packet Delivery Performance in Dense Wireless Sensor Networks*, SenSys 2003.

[2] A. Woo, T Tong, D. Culler, *Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks*, SenSys, 2003.

[3] IEEE Computer society, part 15.4, *Wireless medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless personal Area networks (LRWPAN),* Institute of Electrical and Electronics Engineers, Inc, 2003.

[4] J-S. Lee, *An Experiment on Performance Study of IEEE 802.15.4 Wireless Networks, ETFA 20*05.

[5] O. Hyncica, P. Kacz, P. Fiedler, Z. Bradac, P. Kucera, and R. Vrba, *The Zigbee Experience,* ISCCSP 2006.

[6] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, Performance Study of IEEE 802.15.4 Using Measurements and Simulations, WCNC 2006.

[7] M.M. Holland, R.G Aures, W.B. Heinzelman, Experimental Investigation of Radio Performance in Wireless Sensor Networks,2nd IEEE workshop on wireless mesh networks, 2006.

[8] M. Zuniga, and B. KrishnamaChari, *Analyzing the Transitional Region in Low-Power Wireless Links*, SECON, 2004.

[9] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan. and C. Elliott, *Experimental evaluation of wireless simulation assumptions*, Dartmouth Computer Science Technical report, TR2004-507, June 2004.

[10] A. Cerpa, J.L. Wong, M. Potkonjak, D. Estrin, *Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing*, MobiHoc, 2005.

[11] A. Meier, *Safety-Critical Wireless Sensor Networks*, PhD thesis 18451, ETHZ, 2009.

[12] D.S.J. de Couto, D. Aguayo, B.A. Chambers, R. Morris, *Performance of Multihop Wireless Networks: Shortest Path is Not Enough,* ACM SigComm communications review, Vol 33, No 1, 2003.

[13] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, *Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks*, Technical Report SING-06-00, 2006.

[14] G. Zhou, T. He, S. Krishnamurthy, and J.A. Stankovic, *Impact of Radio Irregularity on Wireless Sensor Networks,* MobiSys '04, June 2004.

[15] Y. Wang, M.C. Vuran, S. Goddard, *Cross-layer analysis of the End-to-end Delay Distribution in Wireless Sensor Networks*, 30th IEEE RTSS, 2009.

[16] T. He, J.A. Stankovic, C. Lu, T. Abdelzaher, *"SPEED: a stateless protocol for real-time communication in sensor networks,* ICDCS-23, 2003.

[17] M. Soyturk, D.T. Altilar, *Reliable Real-Time Data Acquisition for Rapidly Deployable Mission-Critical Wireless Sensor Networks*, IEEE INFOCOM, 2008

[18] R. S. Oliver, G Fohler, *Probabilistic Routing for Wireless Sensor Networks*, 29th IEEE Real-RTSS WiP 2008.

[19] K. Holtman. *Long-duration Reliability Tests of Low Power Wireless Sensing and Control Links in an Office Environment*. In: ARCS 2010 Workshop Proceedings of the 23th Int. Conf. on Architecture of Computing Systems 2010, pp 259-258.

[20] http://www.jennic.com/

[21] K. Holtman; P. van der Stok. *Routing recommendations for low-latency 802.15.4 control networks.* In: ARCS 2011 Workshop of the 24th Int. Conf. on Architecture of Computing Systems 2011, pp 285-290.

[22] P. Jurcik, R. Severino, A. Koubaa, M. Alves, E. Tovar, "Dimensioning and Worst-case Analysis of Cluster-Tree Sensor Networks", ACM Transactions on Sensor Networks, Volume 7, Issue 2, Article 14, August 2010.

[23] H. Raisigel, G. Chabanis, I Ressejac, M. Trouillon. Autonomous Wireless Sensor Node for Building Climate Conditioning Application, Sensor Technologies and Applications (SENSORCOMM) 2010.

Quantifying the Channel Quality for Interference-Aware Wireless Sensor Networks

Claro Noda†, Shashi Prabh†, Mário Alves†, Carlo Alberto Boano¶, Thiemo Voigt‡

†CISTER Research Unit, ISEP
Instituto Politécnico do Porto
Porto, Portugal
{cand,ksp,mjf}@isep.ipp.pt

¶Institute of Computer Engineering
University of Lübeck
Lübeck, Germany
cboano@iti.uni-luebeck.de

‡Swedish Institute of
Computer Science
Kista, Sweden
thiemo@sics.se

ABSTRACT

Reliability of communications is key to expand application domains for sensor networks. Since Wireless Sensor Networks (WSN) operate in the license-free Industrial Scientific and Medical (ISM) bands and hence share the spectrum with other wireless technologies, addressing interference is an important challenge. In order to minimize its effect, nodes can dynamically adapt radio resources provided information about current spectrum usage is available.

We present a new channel quality metric, based on availability of the channel over time, which meaningfully quantifies spectrum usage. We discuss the optimum scanning time for capturing the channel condition while maintaining energy-efficiency. Using data collected from a number of Wi-Fi networks operating in a library building, we show that our metric has strong correlation with the Packet Reception Rate (PRR). This suggests that quantifying interference in the channel can help in adapting resources for better reliability. We present a discussion of the usage of our metric for various resource allocation and adaptation strategies.

Categories and Subject Descriptors

C.4.3 [**Performance of Systems**]: Measurements Techniques. Reliability, Availability, and Serviceability

General Terms

Experimentation, Measurement, Performance, and Reliability.

Keywords

Channel Quality, Interference, ISM Bands, Dynamic Resource Adaptation, Wireless Sensor Networks.

1. INTRODUCTION

Wireless technologies have grown exponentially during the last decade and are progressively cast around for more applications. Many standardized technologies operate in crowded license-free Industrial Scientific and Medical (ISM) frequency bands. Wireless networks in these bands are now ubiquitous in residential and office buildings as they offer great flexibility and cost benefits. However, despite the extensive research, the issue of reliability of wireless networks remains a challenge. Medium access techniques such as TDMA and FDMA cannot be readily applied in the context of ISM bands [1], as they are not designed to tolerate inter-network interference. Instead, distributed multiple access schemes based on *carrier sense*, such as CSMA, are widely employed along with Spread Spectrum modulation techniques which provide some robustness as well as generate lower levels of interference. Although this bottom-up approach to unlicensed spectrum usage exacerbates the challenges to achieve reliability and predictability in low-cost wireless solutions, there are many gains for end users [2] and extensive opportunities for innovation [3]. It has also incubated new research directions, such as dynamic spectrum allocation for future wireless systems [4]. Inspired by this paradigm, we investigate mechanisms for interference avoidance within ISM bands for low-power radios.

Wireless Sensor Networks (WSN) are seen as a viable alternative for monitoring, control and automation applications, provided they are made appropriately reliable and delays are bounded. To this end, interference and coexistence pose a major challenge. In this paper, we present the *Channel Quality* (CQ) metric that provides a quick and accurate estimate of interference by capturing a channel's availability over time at a very high resolution. This metric is useful towards achieving better reliability and lower latency through dynamic radio resources allocation.

Interference from coexisting networks in ISM Bands is typically referred as Cross Technology Interference (CTI). Even though CTI represents a well known problem [5–8] it has not been adequately addressed in WSN. This problem is hard to resolve for two reasons: a) efficient cooperative schemes for spectrum access are not possible with currently deployed technologies and b) there are large RF power and spectrum footprint asymmetries. CTI could be avoided by sophisticated communication protocols that are sensitive to instantaneous spectrum occupation. However, low-cost hardware and limited energy-budget of the nodes make the typical spectrum sensing techniques as proposed for non resource constrained systems [9] unsuitable for WSN.

This paper has the following contributions:

- A novel channel quality metric that is based on channel availability and is agnostic to the interference source.
- An analysis of the parameter space and validation of the metric's performance with real-world interference traces.

The rest of this paper is organized as follows. Section 2 provides further motivation for this work and in Section 3 we derive the expression for our CQ metric. Section 4 describes how we use the energy detection (ED) feature in IEEE-802.15.4 compliant radios to measure evolution of signal (interference) strength in 802.15.4 channels, our experimental setup and our data collection experi-

ments. We then discuss results of our evaluations and conclude the paper in Section 5.

2. MOTIVATION

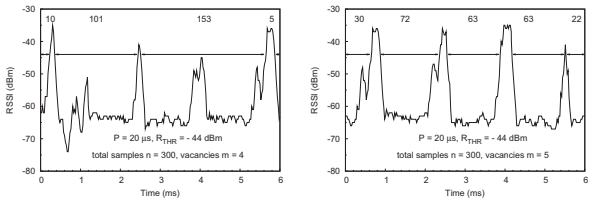
Any given network configuration at deployment phase, like channel selection, is typically not enough as the network may experience communication interruptions or simply fails at some point. We need WSN that seamlessly adapt resources and self-organize to maintain their integrity in a changing environment. Several recent studies have addressed burstiness and interference in wireless links. Srinivasan et al. proposed a metric to quantify link burstiness and show impact on protocol performance and achievable improvements in transmission cost [10]. Also, Munir et al. investigated scheduling algorithms to improve reliability and provide latency bounds [11]. However, these solutions can not react to instantaneous changes in the channel condition. They rather select routes and channels using long-term observations.

There are aggressive techniques to deal with interference in wireless systems. Successive Interference Cancellation (SIC) has been partially demonstrated for 802.15.4 in Software Defined Radios [12]. Nevertheless, there are practical limitations to advance with it. For example, it is known that SIC requires highly linear amplifiers in the receiver (large dynamic range) and also excellent adjacent channel suppression, because residual energy put in the front-end causes it to underperform and desensitizes the radio. Both of these requirements lead to expensive solutions. Furthermore, it is questionable whether SIC's demand for signal processing could outweigh its benefits compared to other approaches, in view of available technology, inexpensive hardware and energy budget constrains. Finally, these ideas are not trivially applied to CTI because a large heterogeneous set of possible signals to disentangle further complicate SIC-based solutions.

Alternatively, we advocate modest improvements in low-power receiver architecture can enable energy efficient spectrum sensing, which is necessary for nodes to form smart reactive networks that eliminate the need for highly complex radios. Spectrum occupation can change rapidly in time and space, yet under unfavourable channel conditions nodes adapt resources or find better channels to maintain communications. Dynamic resource adaptation can lower latency bounds and boost reliability but in order to encompass this information into protocols one needs accurate spectrum sensing. In this paper we show that sufficiently accurate spectrum sensing is feasible with sensor nodes.

Currently, the radio transceivers in WSN nodes are mostly based on the IEEE-802.15.4 standard that is intended for low-power operation. On reception, off-the-shelf radios require around 50 mW and consume 200 – 2000 μJ per packet received. This power is drawn by the PLL synthesizer, digital demodulator, symbol decoder and RF analog blocks for signal filtering, amplification and down-conversion among other functions, typically in this order. Recent incursions in 0.18 μm CMOS process of PLL realizations [13–16], targeted for these systems, report fairly appealing figures: power consumptions below 3 mW and lock-in times less than 30 μs. Since the PLL synthesizer is known to be by far the most power-hungry block in the receiver, these results suggest that the next generations of WSN radios would require, at least, one order of magnitude less chip energy per bit received.

Now, in order to support ED spectrum sensing only the PLL synthesizer, analog RF blocks plus AGC are necessary, while the demodulator can be turned off. Interestingly, among other optimizations, this further reduces energy consumption while the receiver is used exclusively to detect the RF energy in the channel, but we have not yet found any 802.15.4 radio chip providing this flexibility.



(a) Scenario 1 (b) Scenario 2

Figure 1: Channel vacancies: two scenarios with the same Channel Availability (CA).

3. CHANNEL QUALITY METRIC

The sources of interference in wireless networks are typically very diverse. Interference causes a decrease in the Signal-to-Noise plus Interference Ratio (SNIR) which can result in packet losses. Any device that produces RF signals with spectral components within or near the receiver passband is a potential interferer. Average energy in a channel has been used as an indicator of channel usage in the previous literature [17–20]. Unfortunately, this metric is unable to distinguish between a channel where the traffic is bursty with large inactive periods and a channel that has very high frequency periodic traffic with the same energy profile. Clearly, the first scenario is preferable. It may well be the case that the traffic in the second case consists entirely of short-duration peaks resulting in much lower average energy but unusable channel. Motivated by this observation, we propose a metric that is based on the fine-grained availability of the channel over time and ranks in a more favourable way channels with larger inactive periods or vacancies.

Consider the energy levels (or RSSI) in some channel are measured periodically with period P. Suppose, the acceptable noise level and interference threshold is R_{THR}. Therefore, the channel can be considered idle when $RSSI < R_{THR}$. For example, Figure 1 shows RSSI samples over time along with idle intervals, which we refer to as *channel vacancies* (CV). Let m_j denote the number of CV consisting of j consecutive idle samples and n the total number of samples. Then $m_1 + m_2 + \ldots + m_n = m$ is the total number of observed CV. Notice that j consecutive clear channel samples imply that the channel was idle for at least $(j-1)P$ time units. We define the average *Channel Availability* (CA) as:

$$CA(\tau) = \frac{1}{n-1} \sum_{j|(j-1)P>\tau} jm_j \qquad (1)$$

where $\tau > 2P$ is the time window of interest, which could be the duration of packets. As we argued earlier, a channel where $m_{2j} = k$ is more desirable than a channel where $m_j = 2k$, although jm_j is the same for both cases. Therefore, we want to rank a channel with larger vacancies higher even though the sum of the idle durations might be the same. Hence, we define the *Channel Quality* metric as:

$$CQ(\tau) = \frac{1}{(n-1)} \sum_{j|(j-1)P>\tau} j^{(1+\beta)} m_j \qquad (2)$$

where $\beta > 0$ is the bias. CQ in equation (2) take values between 0 and n^β, where the larger values indicate better channels. Observe that this expression is agnostic to the interference source.

Figure 1 shows the amount of channel vacancies in two scenarios with a similar channel availability ($CA_a = 0.88$ and $CA_b = 0.83$) computed with a $R_{THR} = -44$ dBm. Due to collisions, the probability of correct reception is higher in the scenario shown in Fig-

(a) (b)

Figure 2: The experimental setup used to collect energy level traces on IEEE-802.15.4 channels deployed at the Library of the Faculty of Engineering at the University of Porto (a) and detail of TelosB motes arranged in a USB hub (b).

ure 1(a) than in the one depicted in Figure 1(b).

4. EVALUATION

In this section, we first describe our experimental set-up used for data collection followed by an analysis of our metric when applied to the data. We devise off-line experiments and implement them in Python [21] scripts to be run over the traces. This has the advantage of producing a naturally controlled environment, e.g. isolating channel effects that are present in an online experiment. We show that our metric is highly correlated with PRR.

4.1 Experimental Setup

In order to experimentally investigate our proposal we need traces of interference signals that help understand channel degradation in real-world settings. More specifically, we want to find out how our metric can help identifying a usable channel and eventually establish which alternative techniques can be applied to employ it effectively. Therefore, we have designed an experimental setup to study interference in the 2.4 GHz ISM band. This band is available globally; there are thousands of certified devices on the market that operate in it and coexistence problems are well known [5, 6], which ultimately facilitates the task of collecting interference traces. Our setup has no limitations to study any kind of interference, but given that Wi-Fi has been identified as the most critical interference source to affect WSN [6] and it is also widely available, in this paper we report experiments with traces where interference stems solely from Wi-Fi networks.

In our setup, we employ a set of 17 TelosB sensor nodes to scan all sixteen IEEE-802.15.4 channels simultaneously. In order to do simultaneous channel readings, we use one of the motes to transmit a scanning beacon on channel 26, which instructs all other nodes to switch to their respectively assigned channels and begin scanning. The motes are connected via USB hubs to a laptop as shown in Figure 2. We sample the RSSI from the CC2420 transceiver at 40 kS/s [22], and store the data in a memory buffer. After completing 5600 samples in approximately 130 ms, i.e., the largest possible amount of samples that can be stored in the constrained memory of TelosB nodes, all nodes return to listen on channel 26 and wait for the next scanning beacon. Scanning beacons are sent every 8 seconds, which guarantees enough time to dump all the RSSI readings to a file. Having one node per channel enables us to increase the pace at which data is collected and makes the logging operation easier.

A large density of Access Points capable of producing notorious spectrum occupation is mainstream in many metropolitan areas today and particularly in university campus. However, it is the density of users and the overall volume of data been transferred that actually produces congestion. Thus, we used our ensemble to collect measurements in our laboratory, which has moderate traffic on a few 802.15.4 channels. Then we conducted a measurement campaign at the Library of the Faculty of Engineering of the University of Porto, where we found very heavy traffic from 802.11 Wi-Fi networks. In our experiments, signals are well above the noise floor (10 - 70 dB), but more relevant is the time distribution of burst patterns that varies from a few microseconds to tens of milliseconds. To examine our metric proposal we then perform off-line experiments, upon a set of traces from a four hour capture.

4.2 Sampling Time

One of the questions we seek to answer is *how long* should we sample a channel in order to have a meaningful CQ value. Sampling too shortly leads to uncertainty about the near future state of the channel. Notice that the *clear channel assessment* (CCA) used in Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism would not help here given the asymmetric scenario in transmission power and spectral footprint [see 8]. Basically, such asymmetries in the PHY layer among different radios, make the distributed coordination approach fail. WSN nodes employ orders of magnitude less RF power than other channel contenders, which makes them more vulnerable to packet corruption since it is improbable that other nodes would detect an ongoing transmission and thus defer theirs. For this reason, it is necessary to sample for longer time, definitely larger than a CCA accounting for 8 symbol periods or 128 μS, in order to capture a sequence of events large enough to estimate the probability of successful packet reception.

On the other extreme, sampling too long introduces a cumulative effect that misses the dynamics of the channel availability and leads to poor prediction of the next state of the channel. The more distant in time the events are the more likely is that their probabilities are independent and therefore does not help to estimate the channel condition either. Furthermore, during the sampling period the radio is turned on which consumes energy.

In practice, this means that we need to find a compromise for the sampling time that is intrinsically dependent on the system observed. In order to understand this compromise, we progressively compute the CQ, up to 120 ms, over all traces. Figure 3 illustrates the results. The R_{THR} threshold value is primarily chosen based on the RSSI levels of packets from other nodes we are interested in receiving.

Actually there is an SNIR margin, specific for each radio and related to the *Co-Channel Rejection Ratio* (CoCRR), which needs to be considered here. In the CC2420 radio, this value corresponds to 3 dB for a target $PER = 10^{-3}$, and must be accounted to fine-tune R_{THR}.

Since we are not interested here in any specific packet duration, we chose $\tau = 0.2$ ms, small enough so that most CV contributions count in Equation 2. Notice that the sum is computed over CV larger than τ only. For practical reasons, we perform data binning on all CV observations, i.e., all values which fall in a small interval, $bin = 0.5$ ms, are represented by the same value. This quantization affects the absolute values obtained in Equation 2. Suppose an empty channel with one single vacancy of length $j = n - 1$. As mentioned in Section 4.1, we are sampling at approximately 40 kS/s, i.e., we take one RSSI sample every 25 μS. Therefore, we divide j by a factor $k = 20$ and thus, $CQ < (\frac{j}{k})^{\beta}$. This curve is the upper bound for absolute values shown in all graphs in Figure 3. Similarly, the values represented in the abscissas are divided by 40 to obtain the corresponding scanning time in milliseconds.

One common trend in all graphs is that CQ stabilizes after some

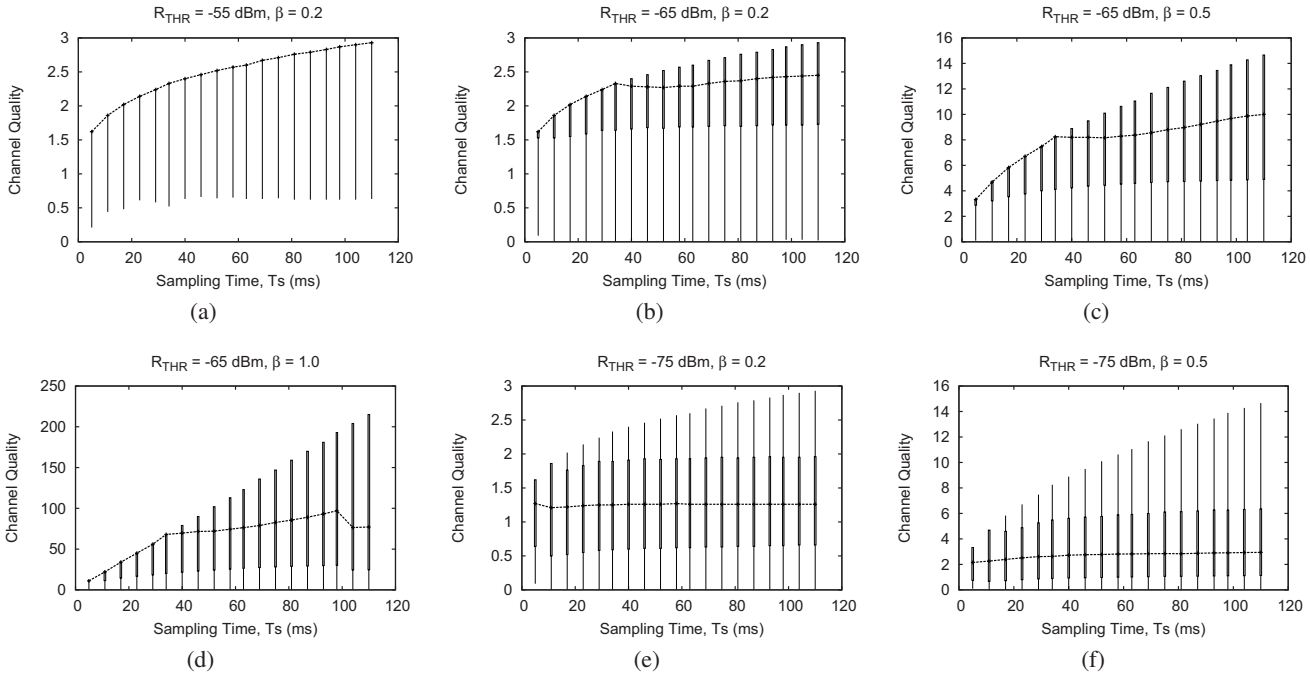


Figure 3: CQ computed over all traces for different sampling times. Curves represent the median, boxes represent the interquartile range (IQR), and bars stand for the rest of the values.

time, provided there is sufficient interference. This indicates that Equation 2 converges toward a value that is proportional to an average number of vacancies during the sampling period and, clearly, also depends on β and the values of j.

Figure 3 shows that for lower R_{THR} values, which corresponds to heavier interference, the median of CQ stabilizes faster. On the contrary, if the channel is mostly idle, CQ grows with very high probability (e.g., if $R_{THR} = -55$ dBm as in 3(a), CQ grows as $(2 \cdot T_s)^{0.2}$ and the IQR shrinks over the maximum). An intermediate case, as when CQ is computed for -65 dBm, see 3(b)-3(d), demonstrates that the metric typically grows to a certain value until it finally stabilizes. Hence, these sampling times are much smaller than the timescale of the interference pattern present in the channel. Based on this behaviour, an optimum sampling time would be as long as it is necessary to have the median of CQ stabilized.

In a system where this metric is computed online the sampling time, represented by n in Equation 2, could be dynamically maintained at this turning point where the median of CQ stabilizes, or below a certain maximum value. We defer the development of a control algorithm for this purpose to future work. For the rest of our experiments we use a hand-picked scanning time of 40 ms.

4.3 Correlation with PRR

Packet Reception Rate (PRR) is a well known reliability metric. When PRR is high, the wireless channel and the link are optimum. However, PRR reflects all forms of signal distortion in the wireless channel, including interference. Thus, a medium or low PRR does not provide enough information to identify factors responsible for poor performance and yet intermediate quality links, that display a medium PRR, may account for up to 50% of all links observed in WSN testbeds [10]. Moreover, PRR and other useful metrics, such as packet's RSSI and LQI, require packet transmissions. Instead, our CQ metric specifically accounts for interference and has no side effect on the channel, as it relies exclusively on the receiver channel energy detection, and therefore scales with node density and channel usage.

We now investigate how the channel availability as described by our CQ metric is related to the probability of successful packet receptions. For this experiment we use a third of each RSSI trace, lasting 130 ms, to compute the metric and the remaining to check for the presence of interference that may lead to packet corruption. If the energy levels in the channel remain 3 dB below the RSSI of packets (CoCRR mentioned in 4.2), during the duration of each packet, then the packet is considered successfully received.

Multiple packets are transmitted over each trace and the average is computed to derive the PRR. Packets are transmitted periodically and transmissions are separated by an Inter-Packet Interval time (IPI) of 2 ms. In this way, we conduct an experiment with more than 240.000 off-line packet verifications on traces obtained from the deployment at the library.

As shown in the previous section, Equation 2 provides a range for CQ values that depends on n and β. However, in order to compare among CQ values computed with different parameter values we rewrite CQ as:

$$CQ(\tau) = \frac{1}{(n-1)^{(1+\beta)}} \sum_{j|(j-1)P>\tau} j^{(1+\beta)} m_j, \qquad (3)$$

CQ in Equation (3) now take values between 0 and 1, regardless of n and β values. For example, if we compute Equation 3 with $\beta = 0.3$ for an IEEE-802.15.4 ACK frame lasting 352 μs, in the scenarios shown in Figure 1, we obtain $CQ_a = 0.65$ and $CQ_b = 0.50$. The difference between CQ_a and CQ_b is three times higher than the difference between CA_a and CA_b (obtained using Equation 1 in Section 3), and it therefore highlights the difference in quality between the two channels.

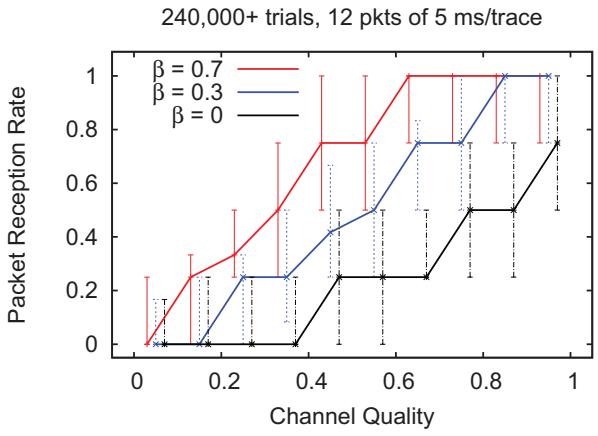


Figure 4: PRR and CQ computed according to Equation 3

Figure 4 shows the correlation between CQ and PRR. The curves correspond to the CQ median and the error bars represent the interquartile range computed over the entire set of traces, at $R_{THR} = -65$ dBm. We compute CQ for bias (β) values 0, 0.3, and 0.7 to highlight that $\beta = 0.3$ linearises the curves and better expand useful range of CQ values.

Higher values of β increase the weight of larger CV in the definition of the CQ. Therefore, having $\beta = 0.7$ will promote the selection of channels with larger CV. On the other hand, observe how for $\beta = 0$, which is equivalent to compute the average channel availability as described by Equation 1, CQ values increase up to 0.4 but almost no packets are received, since vacancies are not large enough. In this case, CQ values grow faster than PRR which indicates that average channel availability does not capture well-enough the complexity of the channel, as we discussed in Section 3. Being able to tune β, as shown in the graph in Figure 4, helps us in maximizing the correlation among PRR and CQ. This makes our metric an accurate indicator of the channel condition, which is an interesting result. In our experiments, we find the β value that maximises the correlation among CQ and PRR to be approximately $\beta = 0.3$. Similar to the scanning time, finding the optimum β value, for an arbitrary interference scenario, is outside the scope of this paper.

4.4 Discussion

In this section we revisit some resource adaptation techniques and discuss how they could be dynamically applied to leverage our CQ metric for interference-aware communication protocols.

Lin et al. demonstrated a novel pairwise transmission power control for WSN that performs significantly better than node-level or network-level power control methods [23]. They improve PRR and energy consumption by dynamically adapting the RF transmission power to maintain the minimum level required to guarantee a good link. This is a clever approach to compensate for the non-linear pathloss. However, it does not account for two important aspects: a) the irreducible error floor [24, Ch. 6] produced by fading can not be removed by increasing transmission power and b) it does not address external interference. A solution to both these problems is dynamic frequency and power adaptation, simultaneously.

In this regard, one can augment such pairwise power control mechanism with CQ, directly establishing a dynamic lower bound for the RF power to use in the transmitter, previous to actual transmissions. Besides, since a maximum transmission power can not be exceeded, an alternative such as a moving to a different channel may be inferred immediately. Starting from the RSSI samples in memory, we could ask the question: which signal level would result in a CQ value that satisfies a given requirement for channel usage under the current interference level?

In general, protocols designed for multichannel operation can maintain good links by using well ranked channels by distributed CQ computations among neighbour nodes, provided a control channel among them is stable. Additionally, these CQ values could aid route changes when an interferer's spectral footprint is very large, as in 802.11n, to take advantage of the irregular coverage, common in some indoor environments.

Successful transmissions in the scenarios in Figure 1 also depend on the packet size. Certain packet size would maximize throughput or minimize the time to deliver a data object over the channel, for a given interference level. Observe that shorter packets have better chances of avoiding collisions (and hence retransmissions) but also result in higher overhead due to fixed packet headers and delays due to acknowledgement timeout. One could look into the relationship between these optimum packet sizes and the CQ values computed on the channel. Based on observed CQ values protocols can then tune packet size to save energy or transfer data in a minimum time.

FEC techniques pose a trade-off between data recovery capacity and its inherent payload and computation overhead. Recently, Liang et al. demonstrated the Reed-Solomon (RS) correcting codes performs well while recovering packets affected by 802.11 interfering signals [8]. Since interference levels may vary extensively it is interesting to see if this solution can benefit from simple CQ based optimizations.

On the other hand, energy cost to compute the CQ metric must be further explored in view of overall energy balance in dynamic link adaptation. In future work we plan to extend our experiments and later implement the metric on WSN hardware.

5. CONCLUSIONS

We introduced a new channel quality metric that is based on the availability of the channel over time. The metric is useful for interference aware protocols in WSN. We described our experimental setup for collecting real-world interference traces in the 2.4 GHz ISM band. Using this data, we showed that our metric has strong correlation with PRR. Thus, our metric's characterization of a channel is reliable and applicable in practice. We also discussed dynamic resource allocation techniques for interference-aware protocols in WSN for which our metric can prove to be useful. We are currently working on a software implementation of CQ for WSN hardware to further validate its performance in online experiments.

6. ACKNOWLEDGEMENTS

Authors acknowledge support from the Cooperating Objects Network of Excellence (CONET), a EU-funded project under ICT, Framework 7 and the Portuguese Science Foundation (FCT). We also are grateful to the staff at the Library of the Faculty of Engineering of the University of Porto.

References

[1] Kenneth R. Carter. Unlicensed to kill: a brief history of the Part 15 rules. *Info*, 11(5):8–18, 2009.

[2] Henry Goldberg. Grazing on the commons: the emergence of Part 15. *Info - The journal of policy, regulation and strategy for telecommunications*, 11(5):72–75, 2009.

[3] Vic Hayes and Wolter Lemstra. Licence-exempt: the emergence of Wi-Fi. *Info - The journal of policy, regulation and strategy for telecommunications*, 11(5):57–71, 2009.

[4] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13):2127 – 2159, 2006.

[5] Axel Sikora and Voicu F. Groza. Coexistence of IEEE 802.15.4 with other systems in the 2.4 GHz-ISM-Band. In *IEEE Instrumentation and Measurement Technology*, pages 1786–1791, Ottawa, Canada, May 2005.

[6] Marina Petrova and Lili Wu and Petri Mähönen and Janne Riihijärvi. Interference Measurements on Performance Degradation between Colocated IEEE 802.11g/n and IEEE 802.15.4 Networks. In *Proc. of International Conference on Networking (ICN)*, Sainte-Luce, Martinique, April 2007.

[7] Jan-Hinrich Hauer, Andreas Willig, and Adam Wolisz. Mitigating the Effects of RF Interference through RSSI-Based Error Recovery. In *Proc. of the 7th European Conference on Wireless Sensor Networks (EWSN)*, pages 224–239, Coimbra, Portugal, February 2010.

[8] Chieh-Jan Mike Liang, Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving Wi-Fi Interference in low-power ZigBee Networks. In *Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*, pages 309–322, Zurich, Switzerland, November 2010.

[9] T. Yucek and H. Arslan. A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications. *IEEE Communications Surveys Tutorials*, 11(1), 2009.

[10] Kannan Srinivasan, Maria A. Kazandjieva, Saatvik Agarwal, and Philip Levis. The beta-factor: measuring wireless link burstiness. In *Proc. of the 6th Conference on Embedded Networked Sensor Systems (SenSys)*, pages 29–42, Raleigh, NC, USA, November 2008.

[11] Sirajum Munir, Shan Lin, Enamul Hoque, S. M. Shahriar Nirjon, John A. Stankovic, and Kamin Whitehouse. Addressing Burstiness for Reliable Communication and Latency Bound Generation in Wireless Sensor Networks. In *Proc. of the 9th Conference on Information Processing in Sensor Networks (IPSN)*, pages 303–314, Stockholm, Sweden, April 2010.

[12] Daniel Halperin, Thomas Anderson, and David Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless LANs. In *Proc. of the 14th International Conference on Mobile Computing and networking (MobiCom)*, pages 339–350, San Francisco, CA, USA, 2008.

[13] M. Vamshi Krishna, Xie Juan, M.A. Do, K.S. Yeo, and C.C. Boon. A low power fully programmable 1 MHz resolution 2.4 GHz CMOS PLL frequency synthesizer. In *Proc. of the 2nd IEEE Conference on Biomedical Circuits and Systems (BIOCAS)*, pages 187–190, Marrakech, Morocco, November 2007.

[14] Louis-François Tanguay and Mohamad Sawan. An ultra-low power ISM-band integer-n frequency synthesizer dedicated to implantable medical microsystems. *Analog Integr. Circuits Signal Process.*, 58:205–214, March 2009.

[15] Wu Xiushan, Wang Zhigong, Li Zhiqun, Xia Jun, and Li Qing. Design and realization of an ultra-low-power low-phase-noise CMOS LC-VCO. *Journal of Semiconductors*, 31 (8):085007, 2010.

[16] Geng Zhiqing, Yan Xiaozhou, Lou Wenfeng, Feng Peng, and Wu Nanjian. A low power fast-settling frequency-presetting PLL frequency synthesizer. *Journal of Semiconductors*, 31 (8):085002, 2010.

[17] Federico Penna, Claudio Pastrone, Maurizio Spirito, and Roberto Garello. Measurement-based Analysis of Spectrum Sensing in Adaptive WSNs under Wi-Fi and Bluetooth Interference. In *Proc. of the 69th Vehicular Technology Conference (VTC)*, Barcelona, Spain, April 2009.

[18] Luca Stabellini and Jens Zander. Energy-efficient detection of intermittent interference in wireless sensor networks. *International Journal of Sensor Networks*, 8(1):27–40, 2010.

[19] R. Musaloiu-E. and A. Terzis. Minimising the Effect of WiFi Interference in 802.15.4 Wireless Sensor Networks. *International Journal of Sensor Networks (IJSNet)*, 3(1):43–54, December 2007.

[20] Junaid Ansari and Petri Mähönen. Channel Selection in Spectrum Agile and Cognitive MAC Protocols for Wireless Sensor Networks. In *Proc. of the 8th Workshop on Mobility Management and Wireless Access (MobiWac)*, Bodrum, Turkey, October 2010.

[21] Guido Van Rossum. Python for Unix/C Programmers. In *Proc. of the NLUUG najaarsconferentie. Dutch UNIX users group*, 1993.

[22] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Römer, and Marco Zúñiga. JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In *Proc. of the 10th Conf. on Information Processing in Sensor Networks (IPSN)*, pages 175–186, Chicago, USA, April 2011.

[23] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A. Stankovic, and Tian He. ATPC: adaptive transmission power control for wireless sensor networks. In *Proc. of the 4th Conference on Embedded Networked Sensor Systems (SenSys)*, pages 223–236, Bolder, Colorado, USA, November 2006.

[24] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

Existing offset assignments are near optimal for an industrial AFDX network

Xiaoting Li
INP-ENSEEIHT IRIT,
Université de Toulouse
2 Rue Camichel
Toulouse, France
Xiaoting.Li@enseeiht.fr

Jean-Luc Scharbarg
INP-ENSEEIHT IRIT,
Université de Toulouse
2 Rue Camichel
Toulouse, France
Jean-Luc.Scharbarg@enseeiht.fr

Frédéric Ridouard
LISI-ENSMA, University of Poitiers
1 Avenue Clément Ader
Futuroscope, France
Frederic.Ridouard@ensma.fr

Christian Fraboul
INP-ENSEEIHT IRIT,
Université de Toulouse
2 Rue Camichel
Toulouse, France
Christian.Fraboul@enseeiht.fr

ABSTRACT

Avionics Full DupleX Switched Ethernet (AFDX) has been developed for modern aircraft such as Airbus 380. Due to the non-determinism of switching mechanism, a worst-case delay analysis of the flows entering the network is a key issue for certification reasons. Up to now most existing approaches (such as Network Calculus) consider that all the flows are asynchronous and they do not take into account the scheduling of flows generated by the same end system. It is then pessimistic to take into account such a synchronous scenario. Each end system can be considered as an offset free system, thus the main objective of this paper is to evaluate existing offset assignments in the context of an industrial AFDX network. Existing offset assignments are adapted to take into account specific characteristics of an AFDX network. Worst-case delay results are obtained according to these offset heuristics. It is shown that some existing heuristics are not efficient while some are near optimal for the studied industrial AFDX network.

1. INTRODUCTION

Avionic Full DupleX Switched Ethernet (AFDX [1]) has been proposed in order to satisfy the growing requirements of avionics application. Such a network is defined based on static network configuration and routing. The demonstration of a determined upper bound for end-to-end (ETE) communication delays on such a real-time network plays a key role. Different methods [5, 2, 10, 4] have been presented for the worst-case delay analysis on the AFDX network. Among them, the Network Calculus [3] has been used for the certification of Airbus 380.

Since each end system of the AFDX network schedules its flows according to a local clock, it is pessimistic to consider that all frames arrive simultaneously (synchronous scenario) on this network. This issue has been addressed in [9], in which a computation method integrating the offsets of flows based on the Network Calculus approach has been developed. However, only one offset assignment originally designed for the CAN network in [8] was applied to an industrial AFDX network. It is interesting to consider other existing offset assignments [6, 7] in order to find the best algorithm for an industrial AFDX network. Moreover, the existing algorithms can be adapted in order to take into account specific characteristics of an AFDX network.

The objective of this paper is to evaluate and compute offset assignment algorithms for an industrial AFDX network. The goal of the evaluation is to measure the gap between offset assignment based on heuristics and the optimal assignment, which is intractable on an industrial AFDX network. An upper bound on this gap is computed, based on an optimal scenario.

This paper is organized as follows. Section 2 shortly introduces the context of the studied industrial AFDX network and existing offset assignments. Section 3 derives an ideal offset assignment which gives an optimal scenario for the scheduled flows. In Section 4, new heuristics integrating the AFDX characteristics are proposed. The existing and proposed offset assignments are applied to the industrial AFDX network, and their results are compared and analyzed in Section 5. Section 6 concludes and indicates directions for future research.

2. CONTEXT

2.1 Introduction of the industrial AFDX network

An AFDX network [1] is composed of end systems and switches. The inputs and outputs of the AFDX network, called *end systems* (ES), are connected by several interconnected AFDX switches. Each end system can be connected to only one port of an AFDX switch and each port of an AFDX switch can be connected at most to one end system. Links between switches work in full-duplex mode.

A *Virtual Link* (VL) standardized by ARINC-664 is a concept of virtual communication channel, which statically defines the flows. A connection defined by a Virtual Link is unidirectional, including one source end system and one or more paths leading to different destination end systems (multicast nature). A VL is characterized by:

- *Bandwidth Allocation Gap* (BAG), the minimum delay between two consecutive frames of corresponding VL ranging in powers of 2 from 1 ms to 128 ms, and
- S_{min} and S_{max}, the minimum and maximum frame lengths which respect the standard Ethernet frame.

An AFDX network architecture is illustrated by Figure 1. According to this architecture, there are five end systems and two AFDX switches. On the example, v_1 has a unique path $\{e_1 - S_1 - S_2 - e_4\}$ and v_5 has multi-paths $\{e_3 - S_2 - e_4\}$ and $\{e_3 - S_2 - e_5\}$.

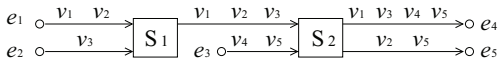


Figure 1: Example of an AFDX configuration

The industrial AFDX network interconnects aircraft functions in the avionics domain. It is composed of two redundant networks. Each network includes 123 end systems, 8 switches, 984 Virtual Links and 6412 VL paths (due to VL multicast characteristics). The left part in Table 1 gives the dispatching of VLs among BAGs. The right part in Table 1

BAG (ms)	Number of VL
2	20
4	40
8	78
16	142
32	229
64	220
128	255

Frame length (bytes)	Number of VL
0-150	561
151-300	202
301-600	114
601-900	57
901-1200	12
1201-1500	35
> 1500	3

Table 1: BAGs and frame lengths

gives the dispatching of VLs among frame lengths, considering the maximum length S_{max}. The majority of VLs considers short frames. Table 2 shows the number of VL paths per length (i.e. the number of crossed switches).

This industrial AFDX network works at 100 Mb/s and the technological latency of an AFDX switch is 16 μs. The overall workload (utilization) of the industrial network is about 10%. Actually, the industrial AFDX network is lightly

Nb of crossed switches	Number of paths
1	1797
2	2787
3	1537
4	291

Table 2: VL paths lengths

loaded in order to guarantee that buffers will never overflow. Both sporadic VLs and periodic VLs exist on the AFDX network, and offsets can be assigned to periodic VLs. There is no global clock in an AFDX network. Consequently, frame releases of different end systems are independent. However, each end system schedules its flows. This scheduling can be integrated in the worst-case delay analysis thanks to offsets. The next paragraph gives an overview of existing offset assignments.

2.2 Existing offset assignments

The offset assignment has been studied in [6] in the context of periodic task sets executed in a uniprocessor. Each task τ_i is characterized by a period T_i, a hard deadline D_i, a processing time C_i and an offset O_i. In the context of uniprocessor, the systems can be classified into three classes in terms of offset:

- Synchronous system: all the tasks have the same fixed offsets, i.e., at time 0, all the tasks generate one request;
- Asynchronous system: an offset is allocated to each task due to application constraints;
- Offset free system: any offset can be allocated to each task in order to improve the system schedulability.

For the third class, a key point is the choice of an offset assignment. The number of possible offset assignments is exponential.

In [6], an *optimal offset assignment* is proposed to exhaust all possible non-equivalent offset assignments. Although this method reduces significantly the number of combinations, the number remains exponential. *Dissimilar offset assignment*, denoted GCD, is then defined in order to reduce computational complexity in the comparison with the *optimal offset assignment* by providing a single offset assignment for a task set. This method tries to move from the synchronous case as much as possible. It considers a minimal distance $\lfloor \frac{gcd(T_i,T_j)}{2} \rfloor$ between two requests of τ_i and τ_j, where $gcd(T_i, T_j)$ is the greatest common divisor of T_i and T_j. This method treats task pairs (τ_i, τ_j) by decreasing value of $gcd(T_i, T_j)$.

Near-optimal offset assignment heuristics are derived in [7] based on the study of GCD. This assignment considers four alternative offset allocations when GCD fails to generate a schedulable asynchronous situation. Since both these two approaches assign offsets to VLs pair by pair, they are called *PairAssign* in this paper. Besides the decreasing value of $gcd(T_i, T_j)$, other heuristics are proposed considering criteria

like utilization rate, i.e., $\frac{C_i}{T_i}$, and the value of $-gcd(T_i, T_j)$ to decide the order of flow pairs. These four heuristics are denoted and defined as follows:

- *RateAdd*: $\frac{C_i}{T_i} + \frac{C_j}{T_j}$,
- *RAGCD*: $(\frac{C_i}{T_i} + \frac{C_j}{T_j}) \times gcd(T_i, T_j)$,
- *RMGCD*: $max(\frac{C_i}{T_i}, \frac{C_j}{T_j}) \times gcd(T_i, T_j)$;
- *GCDMinus*: $-gcd(T_i, T_j)$;

In [8], the authors addressed that the offset assignments mentioned above are not efficient when applied to the scheduling of automotive message, and an offset assignment algorithm is tailored for automotive CAN network. This algorithm, called *SingleAssign* in this paper, aims at choosing offsets to maximize the distance between frames. For n flows emitted by one source node, sort them by increasing value of their periods and calculate $T_{max} = \max_{i \in [1,n]}\{T_i\}$. The assignments start with the flow having smallest period and process one flow after another. For a flow τ_k ($k \in [1, n]$), its offset O_k is decided as follows:

- first search for the least loaded interval in $[0, T_k)$;
- then set O_k in the middle of this interval;
- finally record all the frames of τ_k released in $[0, T_{max})$.

3. OPTIMAL SCENARIO OF SCHEDULED FLOWS OVER THE AFDX NETWORK

Considering an industrial AFDX configuration with about 1000 flows, the *optimal offset assignment* proposed in [6] is intractable. Thus approaches based on heuristics have to be used. Then, the evaluation of the gap between the *optimal offset assignment* and the assignment generated by each heuristic is an important issue. For a given flow, this gap can be defined as the difference between the worst-case ETE delays obtained by, on the one hand considering the *optimal offset assignment*, on the other hand considering the offset assignment based on a heuristic. On a whole configuration, the gap is the average of the gaps obtained for the flows. Obviously, it is not possible to compute the gap for the *optimal offset assignment* on an industrial AFDX configuration, since the *optimal offset assignment* is intractable. Then, a first idea is to compute an upper bound on this gap.

This upper bound can be obtained by considering an ideal offset assignment, which minimizes the worst-case ETE delay for all the flows. This ideal assignment may not exist for a given configuration, but it is sure that it gives worst-case delays which are not higher than the ones obtained by the *optimal offset assignment*. This ideal assignment, denoted *IdealAssign*, minimizes the maximum waiting delay of every frame in each output port it crosses. It corresponds to the following scenario:

- At its source ES, a frame f_i of a VL v_i is not delayed by any other frames emitted by the same ES, i.e., the frame f_i is transmitted immediately after its release;
- At each switch output port of its path, the frame f_i crosses VLs generated by several ESs. f_i can be delayed by exactly one frame coming from each of these ESs. The frame with the largest size S_{max} is considered. The delay encountered by f_i at each switch output port takes into account the serialization effect (i.e., two frames cannot be received at the same time from an input link, see [2] for details).

Indeed, since there is no common clock among the end systems, there is no relationship between the releases of two frames from different end systems. Consequently, there exist scenarios where the two frames arrive at their first common switch output port at the same time.

Let us illustrate this scenario on the example depicted in Figure 2. This sample network has 4 VLs v_1 and v_2 emitted by the ES e_1 as well as v_3 and v_4 emitted by the ES e_2. The network works at 100 Mb/s. The temporal characteristics of each VL are listed in Table 3.

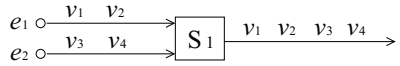


Figure 2: A sample AFDX network

v_i	BAG_i (ms)	S_{max_i} $(Byte)$	C_i (μs)
v_1	8	1000	80
v_2	8	1000	80
v_3	8	1000	80
v_4	8	500	40

Table 3: The Configuration of the sample example in Figure 2

The VL v_1 is focused on. The *IdealAssign* leads to scenario illustrated in Figure 3 where the arrow represents the frame arrival of VL v_i, a_i^h is the frame arrival of v_i at the node h, and the $\boxed{i}$ means the transmission of a frame of VL v_i. At the ES e_1, the frame f_1 is transmitted as soon as it is released due to the separation from v_2. Since the ESs are not synchronized, at the output port of the switch S_1, the frame f_1 of v_1 can arrive at the same time as the frame f_3 of v_3 and it is delayed by f_3, i.e., $a_1^{S_1} = a_3^{S_1}$. Only one frame (f_3) from the ES e_2 delays the frame f_1 at the output port of S_1 since v_3 and v_4 are separated far away from each other, and f_3 is considered due to the frame size $S_{max_3} > S_{max_4}$.

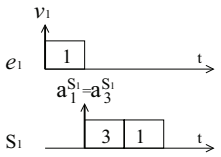


Figure 3: Scenarios of the VL v_1

The *IdealAssign* gives an upper bound on the reduction which can be obtained by an offset assignment algorithm. The next section proposes some offset assignment heuristics tailored for the AFDX network.

4. OFFSET ASSIGNMENTS IN THE CONTEXT OF AFDX NETWORK

In the context of a uniprocessor, a set of tasks shares a unique resource, i.e., the processor. The situation is different in the context of a switched Ethernet network, like the AFDX network, where a set of flows shares a set of output ports. Actually, each port is shared by a subset of all the flows. Consequently, the load can be different for each output port. The worst waiting time of a frame in an output port increases when the load of the output port increases. Then, it could be interesting to take into account the load of the output port in the offset assignment. This is illustrated in the example in Figure 4, where six VLs v_i ($i \in [1,6]$) are transmitted over the network. The temporal characteristics of each VL are given in Table 4. The network works at 100 Mb/s and the technological latency of switch is null.

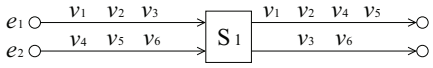


Figure 4: A small example of AFDX network

v_i	BAG_i (μs)	S_{max_i} ($Byte$)	C_i (μs)
v_1	400	500	40
v_2	800	750	60
v_3	400	750	60
v_4	400	500	40
v_5	800	750	60
v_6	400	750	60

Table 4: The Configuration of the small example in Figure 4

The offset assignment *SingleAssign* is applied to this example network. The three VLs emitted by the ES e_1 are considered. The offsets are assigned to these three VLs in order: $O_1 = 0\ \mu s$, $O_3 = 200\ \mu s$ and $O_2 = 100\ \mu s$. This case is drawn in part e_1 in Figure 5. Similar case at the end system e_2 is depicted in part e_2 in Figure 5. v_2 is focused on whose first frame f_2 is released at $O_2 = 100\ \mu s$. At the output port of the switch S_1 where v_2 visits, v_4 and v_5 from e_2 join the path of v_2 while v_3 has left. Then one possible scenario at this output port is depicted in part S_1 in Figure 5. It can be seen that when the frames f_1 and f_2 arrive at S_1, they are still separated far enough to avoid delaying each other. Similarly, the frames f_4 and f_5 from v_4 and v_5 are separated far enough when they arrive at S_1, consequently only one frame f_5 delays the studied frame f_2. Since the frame f_2 is released at the ES e_1 at time $O_2 = 100\ \mu s$ and the transmission of frame f_2 is finished at the switch S_1 at time 280 μs, the delay of the frame f_2 is $R_2 = 280 - 100 = 180\ \mu s$.

The illustration in Figure 5 shows an example where the offset assignment *SingleAssign* succeeds to distribute the workload even in the output port of a switch. It is interesting to demonstrate the case when the workload increases. The example AFDX network in Figure 4 is under study and the maximum frame sizes of VLs v_1 and v_4 are increased to $S_{max_1} = S_{max_4} = 750\ Bytes$ ($C_1 = C_4 = 60\ \mu s$). According to the *SingleAssign*, the releases of frames at e_1 and e_2 are depicted in Figure 6 (same as in Figure 5). One

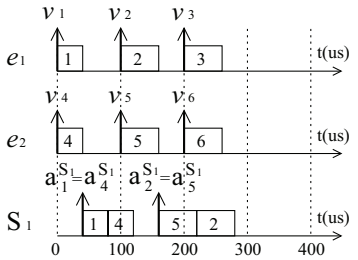


Figure 5: Illustration of the *SingleAssign* with low workload

possible scenario at the output port of S_1 is exhibited in part S_1 in Figure 6, where the studied frame f_2 finishes its transmission at time 300 μs. The delay of the frame f_2 is $R_2 = 300 - 100 = 200\ \mu s$, higher than the case in Figure 5 (180 μs). It increases due to the fact that when the frame f_2 arrives at S_1 at time $a_2^{S_1} = 160\ \mu s$, the transmission of frame f_4, delayed by the transmission of frame f_1, is not completed which delays the transmission of frame f_2. For this case the *SingleAssign* could not separate frames at a crossed switch.

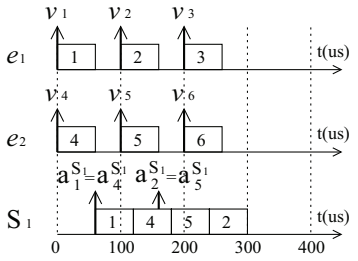


Figure 6: Illustration of the *SingleAssign* with high workload

Note that v_1, v_2 and v_3 emitted by e_1 visit three output ports: e_1 with the utilization $U_{e_1} = \sum(\frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3}) = 0.375$; the upper output port of S_1 with the utilization $U_{S_1} = \sum(\frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_4}{T_4} + \frac{C_5}{T_5}) = 0.45$; and the lower output port of S_1 with the utilization $U_{S_1'} = \sum(\frac{C_3}{T_3} + \frac{C_6}{T_6}) = 0.3$. Consequently, for these three VLs, the most loaded port is the upper output port of S_1, followed by e_1 and the lower output port of S_1. We could first assign offsets to v_1 and v_2 which visit the most loaded port of S_1, then pass to the v_3, leading to the offsets: $O_1 = 0\ \mu s$, $O_2 = 200\ \mu s$ and $O_3 = 100\ \mu s$. This case is illustrated in part e_1 in Figure 7. Similar case for the VLs emitted by e_2 is shown in part e_2 in Figure 7. Then one possible scenario for the frame f_2 at S_1 is identified in part S_1 in Figure 7, indicating that the delay of this frame is $R_2 = 380 - 200 = 180\ \mu s$, which is smaller than the one obtained by *SingleAssign* (200 μs). The reason is that at the most loaded output port of S_1 the workload is further evenly distributed to reduce the waiting time in the buffer.

A proposed algorithm considers separating the VLs by decreasing utilization of the output ports they share. The offsets are first assigned to the flows visiting the most loaded

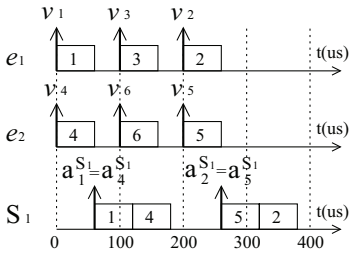


Figure 7: Illustration of the *MostLoadSA* with high workload

port using the assignment *SingleAssign*, then to the flows which are not yet handled in the secondly most loaded port till all the flows of one ES are assigned with offsets. This algorithm is developed based on the assignment *SingleAssign* and denoted *MostLoadSA*.

For the *PairAssign*, a similar heuristic is proposed to consider the load of the output ports. This heuristic, denoted *MostLoad*, sorts the VL pairs (v_i, v_j) by decreasing values of $Ld_i + Ld_j$, where Ld_i is the workload (utilization) of most loaded switch port crossed by v_i.

Due to the nature of the switched Ethernet, flows in one set can share several output ports. When flows share several common switches, the minimum interval between two frames decreases, which can increase the waiting time of a frame in the output port. Then the number of crossed switches can be considered in the offset assignments. For the *PairAssign*, a heuristic, denoted *CrossedS*, is proposed. It sorts the VL pairs (v_i, v_j) by decreasing values of $cs(v_i, v_j)$, where $cs(v_i, v_j)$ is the number of common switches crossed by v_i and v_j. For the *SingleAssign*, a similar heuristic, denoted *CrossedSSA*, is proposed which orders the VLs in one set by decreasing values of maximum number of crossed switch.

Besides the four new proposed heuristics, the existing offset assignment heuristics presented in Section 2.2 are applied to the AFDX network with the value of BAG as the period. The evaluation on each offset assignment is processed in the next section.

5. OBTAINED RESULTS

The existing and proposed offset assignments introduced in Section 4 are applied to the industrial AFDX network presented in Section 2.1. In this evaluation, all the VLs are assumed to be strictly periodic. The computation is processed using the Network Calculus approach integrating the offsets, which has been developed in [9]. The computed ETE delay upper bounds of each offset assignment are compared with those obtained from the network without offset constraints. The statistic reductions on ETE delay upper bounds of each algorithm are listed in Table 5. The columns *Average*, *Max* and *Min* give the average, maximum and minimum reductions, respectively.

The *SingleAssign* as well as its extended algorithms *MostLoadSA* and *CrossedSSA* outperform the *PairAssign* heuristics. Indeed, the average reductions obtained with the *PairAssign* heuristics are 23% (*GCD*) and 32% (*RateAdd*, *RAGCD*, *RMGCD*, *GCDMinus*, *MostLoad* and *CrossedS*). It is 49% for the *SingleAssign* and 51% for the *SingleAssign* based algorithms adapted to the AFDX network. On the considered example, the *SingleAssign* based algorithms are close to the *IdealAssign*, which gives an average reduction of 53%.

Heuristics	Average %	Max %	Min %
IdealAssign	53.48	83.29	21.00
GCD	23.00	70.24	4.01
RateAdd	32.89	73.50	5.08
RAGCD	32.51	72.99	8.85
RMGCD	32.29	70.77	9.99
GCDMinus	32.95	70.06	8.83
MostLoad	32.12	70.06	8.84
CrossedS	32.32	73.03	8.90
SingleAssign	49.67	83.29	18.84
MostLoadSA	51.32	82.94	18.84
CrossedSSA	51.29	82.94	18.84

Table 5: The comparative results

The *PairAssign* heuristics are not efficient in the studied context due to the limited different values of BAG, which lead to same values of $gcd(BAG_i, BAG_j)$ for different VL pairs. Here is a small example in Figure 8. Considering VLs v_1, v_2 and v_3 with $BAG_i = 4\ ms$ $(i \in [1,3])$ of e_1, there are three pairs: (v_1, v_2), (v_1, v_3) and (v_2, v_3). They have the same value of $gcd(BAG_i, BAG_j) = 4\ ms$ $(1 \leq i < j \leq 3)$. *GCD* leads to $O_1 = 0\ ms$, $O_2 = O_1 + \frac{gcd(BAG_1, BAG_2)}{2} = 2\ ms$, and $O_3 = O_1 + \frac{gcd(BAG_1, BAG_3)}{2} = 2\ ms$ $(O_2 = O_3)$. The releases of the first frames for both v_2 and v_3 overlap, and the frames have to wait in the queue. This case is depicted in Figure 9.

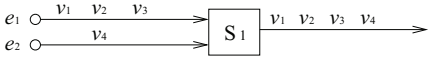


Figure 8: A small example of AFDX

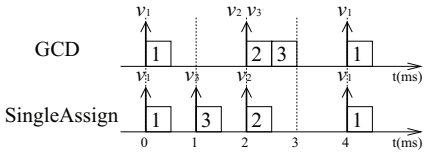


Figure 9: Comparison of GCD and SingleAssign

The situation is different when applying the offset assignment *SingleAssign* (Figure 9). With the same configuration, the offsets are set in order: $O_1 = 0\ ms$, $O_2 = 2\ ms$ and $O_3 = 1\ ms$. In this way, no frame has to wait in the output queue of e_1.

The analyzed problem of *GCD* for the industrial AFDX network exists for all the *PairAssign* heuristics because the computation of offsets mainly concerns the value of $gcd(BAG_i, BAG_j)$ even if the order of pairs varies based on different criteria.

The results are further studied by a normalized method. For one path $\mathcal{P}_x$, the computed ETE delay upper bound without offset assignment is considered as the reference (denoted rf_x) and normalized as 100. The computed result with one offset assignment (denoted cp_x) is taken as the comparison and normalized as Ncp_x:

$$Ncp_x = 100 + (\frac{cp_x - rf_x}{rf_x} \times 100)$$

All the 6412 VL paths are sorted by increasing order of Ncp_x. Three offset assignments are taken into account: *IdealAssign*, *SingleAssign*, and *MostLoadSA*. The comparative results are presented in Figure 10.

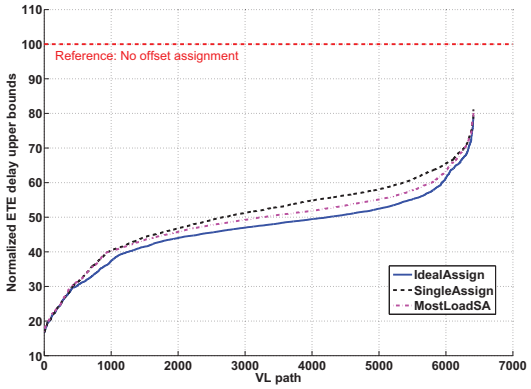


Figure 10: Comparative results of *IdealAssign*, *SingleAssign* and *MostLoadSA*

It can be seen in Figure 10 that the *MostLoadSA* curve is close to the *IdealAssign* curve, which reveals that this algorithm taking into account the AFDX properties works well on this industrial AFDX network. The gap between the *SingleAssign* curve and the *IdealAssign* curve is also small (although bigger than the gap with *MostLoadSA* curve). It suggests that a simple algorithm could be efficient to separate the flows of the industrial AFDX network.

Further evaluations have been conducted, leading to the same conclusions. They consider the same industrial AFDX architecture described in Section 2.1 and the overall workload 10% is kept. For each VL, the S_{min} and S_{max} are randomly chosen from 72 *bytes* to 1526 *bytes*, and the BAG value is randomly chosen from 1 ms to 128 ms as the powers of 2. The results show that the average ETE delay reduction brought by the $IdealAssign$ is 45%. The $PairAssign$ heuristics bring average reductions ranging from 24% to 31%, which are far from the $IdealAssign$. The algorithms based on the $SingleAssign$ bring average reductions ranging from 39% to 40%, which are closer to the $IdealAssign$.

6. CONCLUSION

In this paper, the offset assignments for the industrial AFDX network are studied. Since the *optimal offset assignment* is intractable in this context, an optimal scenario is built based on a presumed ideal assignment in order to upper bound the gap between the *optimal offset assignment* and each offset assignment heuristic. New heuristics considering the AFDX characteristics are proposed. Using the Network Calculus approach, the improvement on ETE delay upper bound bought by each heuristic is compared to the ideal algorithm. It is demonstrated that *PairAssign* heuristics are not efficient when applied to the industrial AFDX network due to the limited different values of BAG. The *SingleAssign* turns out a near optimal algorithm in the studied context. Although the heuristics integrating specific AFDX characteristics bring slight improvements in contrast to the *SingleAssign*, they are of increased complexity.

The industrial AFDX network considered in this paper is lightly loaded. The offset assignment for a switched Ethernet with heavier workload remains an open question, which is the subject of our ongoing work.

7. REFERENCES

[1] Arinc 664, 2002-2008.

[2] H. Bauer, J.-L. Scharbarg, and C. Fraboul. Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach. *IEEE Trans. Ind. Informat.*, 6(4):521–533, November 2010.

[3] J.-Y. L. Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer-Verlag, Berlin Heidelberg New York, 2001.

[4] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul. Method for bounding end-to-end delays on an AFDX network. In *Proc. the 18th Euromicro Conference on Real-Time Systems (ECRTS'06)*, pages 192–202, July 2006.

[5] F. Frances, C. Fraboul, and J. Grieu. Using network calculus to optimize the AFDX network. In *Proc. 3th European Congress on Embedded Real Time Software (ERTS'06)*, January 2006.

[6] J. Goossens. Scheduling of offset free systems. *Real-Time Systems*, 24(2):239–258, March 2003.

[7] M. Grenier, J. Goossens, and N. Navet. Near-optimal fixed priority preemptive scheduling of offset free systems. In *Proc. 14th International Conference on Real Time Network and Systems (RTNS'06)*, pages 35–42, May 2006.

[8] M. Grenier, L. Havet, and N. Navet. Pushing the limits of CAN - scheduling frames with offsets provides a major performance boost. In *Proc. 4th European Congress on Embedded Real Time Software (ERTS'08)*, January 2008.

[9] X. Li, J.-L. Scharbarg, and C. Fraboul. Improving end-to-end delay upper bounds on an AFDX network by integrating offsets in worst-case analysis. In *Proc. IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'10)*, pages 1–8. IEEE, September 2010.

[10] J.-L. Scharbarg, F. Ridouard, and C. Fraboul. A probabilistic analysis of end-to-end delays on an AFDX avionic network. *IEEE Trans. Ind. Informat.*, 5(1):38–39, February 2009.

Networking in Modern Avionics: Challenges and Opportunities

Sérgio D. Penna
EMBRAER
Avenida Brigadeiro Faria Lima, 2170
12227-901 - São José dos Campos - Brazil
+55-12-3927 4067
sdpenna@embraer.com.br

ABSTRACT
This paper provides an overview of three important emerging standards in modern Avionics - ARINC-664 Part 7 (AFDX), TTP and ARINC-653 - and addresses challenges and opportunities arising from their adoption. Beyond several potential benefits offered by these standards, additional effort should be expected by practitioners when system design requires, for instance, timing analysis for estimating data transmission jitter. This paper presents some of these design challenges and suggests new academic research opportunities for, for instance, extending current timing analysis methods for distributed systems.

Categories and Subject Descriptors
J.2 [**Computer Science**]: Physical Sciences and Engineering – *Digital Data Communication Networks*. C.2.2 [Computer Systems Organization]: Computer Communication Networks - Network Protocols.

General Terms
Avionics, Networking.

Keywords
Networking, Avionics, Digital Data Bus, Time-Triggered Architecture, Operating Systems, Partitioning.

1. INTRODUCTION
The introduction of "Integrated Modular Avionics" (IMA) by the Radio Technical Commission for Aeronautics (RTCA DO-297) in November 2005 [1] gave focus to new industry standards. "Avionics Full Duplex Switched Network" (ARINC 644 Part 7 "AFDX") [2], "Time-Triggered Protocol" (TTA Group "TTP") [3] and "Application Executive interface" (ARINC 653 "APEX") [4] emerged offering new levels of modularity and communality to avionic systems. These standards present new challenges for system manufacturers and integrators, but offer new opportunities to improve current analytical methods for predicting system behavior during the design phase.

2. CURRENT AVIONICS DESIGN
Previous avionics systems were dominated by what was commonly called “Federated Architecture”, whereby one function or application was confined in one “black box”. In federated systems, these “black boxes” communicate through digital buses such as ARINC-429, which provided a point-to-point single-channel communication, or MIL-STD-1553, which provided an arbitrated data bus.

Such architecture led sometimes to excessive cabling, for one box needed to be physically connected to multiple other boxes, so applications could exchange data.

More recently, with the introduction of “IMA - Integrated Modular Avionics”, concept formalized in 2005 as the Radio Technical Commission for Aeronautics (RTCA) standard DO-297 “Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations”, things began to change.

With IMA, one box could host more than one application, so boxes became cabinets populated with multiple processing and input-output modules. New digital data buses, such as AFDX and TTP, appeared offering better ways of exchanging data among different applications. The use of COTS technologies, in particular Ethernet/IP networking became ubiquitous for obvious reasons, given the multitude of hardware and software resources and academic research around it.

However, not only data communication became a concern under IMA. Software Certification issues and the desire to free application from underlying proprietary operating system software interface favored the advent of another standard ARINC-653 “Avionics Application Software Standard Interface”, which enforces not only one single software interface between applications and operating system services, but also impose a strict logical separation between applications running in the same processing module.

The next paragraphs will detail more these three important standards: AFDX, TTP and ARINC-653 at the new challenges system designers and opportunities for academic researchers.

3. THE “AFDX” STANDARD
The “AFDX” standard is the Part 7 of the ARINC-664 “Aircraft Data Network” standard, called “Avionics Full Duplex Switched Ethernet Network” introduced formally in 2005. It describes a “more deterministic” switched Ethernet/IP network, that is, a switched network where a few constraints are applied. On the transmitting end, called “End-System”, a data transmission rate is associated to one virtual multicast unidirectional communication channel called “Virtual Link”, or VL in terms of a “Bandwidth Allocation Gap”, or BAG measured in milliseconds, and a

maximum frame size, called "Lmax" measured in bytes. Thus, the rate is defined as "Lmax/BAG" and is defined per VL.

Since a switched network implies the use of a switch, this piece of hardware became crucial to the design of the network. An AFDX Switch does not only perform packet forwarding as usual, but also enforces Traffic Policing at its input ports. This feature is based in the "Token Bucket" algorithm and discards packets that arrive in a pace faster than "Jswitch" milliseconds. This "Jswitch" quantity is programmed in the AFDX switch and is defined per VL. With Traffic Policing, the AFDX switch protects the network from what is usually called "babbling idiot", a misbehaved node that transmits more that than it is designed to.

The AFDX data frame uses a suffix (lower 16 bits) in its multicast MAC Destination Address to define a VL. The remainder of the frame takes its model from UDP/IP with one difference: the last byte is reserved for count frames from 1 to 255, a quantity used in AFDX'es "Redundancy Management". The AFDX network uses two physically separated channels, so each AFDX "End- System" transmits the same data frame in these two channels at the same time. In the receiving "End- System", the "Sequence Number" is used by the "Redundancy Manager" to discard frame copies that arrive too late.

Designers of an AFDX network need to take into account several potential sources of transmission jitter. On the transmitting "End-System", frames are queued before reaching the physical medium. Inside the AFDX switch, forwarded frames are queued in the output port before they depart to their destination node. The measure of jitter is relevant to the AFDX network design, for the "Jswitch" quantity must be correctly estimated and programmed in the switch for each VL.

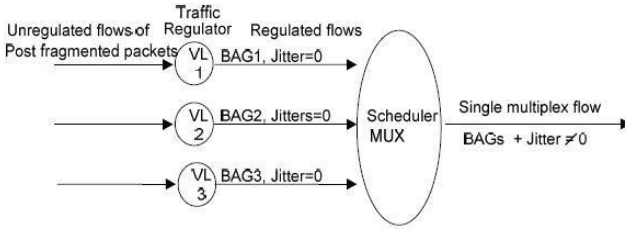


Figure 1. Virtual Link Flow Regulation in AFDX.

Adapted from ARINC-664 Part 7, Page 12 – Figure3-6.

Figure 1 shows how traffic flow is regulated on a transmitting AFDX End-System. Note that transmission jitter should be expected after messages are produced at a rate equal to BAG for each VL.

4. THE "TTP" STANDARD

The "Time-Triggered Protocol" (TTP) was introduced in 2003 by the "Time-Triggered Architecture Group" (TTA-Group) and has been recently adopted by SAE as the AS6003 standard.

The basic principle of TTP communication is "Time-Division Multiple Access" (TDMA). TTP nodes are time-synchronized and are allowed to transmit using the full speed of the physical medium for a limited time period. TTP node transmits in turns according to a precise schedule recoded in the "Message Description List" copied in each TTP node before operation starts. Each TTP node has one reserved transmission "Slot", many "Slots" form a "Round", and many "Rounds" form a "TTP cluster". "Rounds" are usually short in the order of a few milliseconds and "TTP Clusters" are as long as a few tens of milliseconds.

Messages transmitted in each "Slot" can be as long as 240 bytes and the format is application dependent. Clock synchronization is achieved using an distributed clock correction algorithm described as "Fault Tolerant Average" (FTA) [5]. The TTP network has two physically separated channels that can be used in redundancy or as two independent channels.

Designers of a TTP network need to take into account the period and the execution time of the applications that transmit and receive data, for its is essential for time-critical applications, such as a Flight Control System, that the data produced is consumed as early as possible.

Figure 2 shows a typical TTP Cluster taken from TTTech's "Brake-by-Wire" book example, as displayed by TTPlan®, TTTech's own TTP configuration tool. It has 7 slots and 4 rounds of 2,500 microseconds each. Note that, in this case, the two physical channels are used independently to transmit different size messages.

5. THE ARINC-653 STANDARD

The ARINC-653 standard was formally introduced in 2005. In its words: "The primary objective of this Specification is to define a general-purpose APEX (APplication/EXecutive) interface between the Operating System (O/S) of an avionics computer resource and the application software". This standard introduces two important concepts: "Temporal Partitioning" and "Spatial Partitioning".

"Temporal Partitioning" is realized by the introduction of a fixed periodic scheduling of "Partitions", a limited time-window where applications are allowed to execute. "Spatial Partitioning" is realized by the definition of each "Partition" virtual address space at system startup. This logical separation allows applications of different criticality levels (as defined in the ARP-4754 standards) to run in the same processing module.

Furthermore, the ARINC-653 standard defines a complete set of operating system services for managing partitions, processes and data communication within a partition and between partitions. The latter introduces the concept of "ports", which in turn are immediately associated to UDP ports as used by AFDX networks.

System designers that decide for the ARINC-653 "APEX" face the challenge of not only having to estimate the execution time of tasks, but also to decide how long should be the duration of each partition where these tasks are expected to execute. Should an application exceed the allotted time-window of its partition, it will be suspended and resumed only in the next partition period.

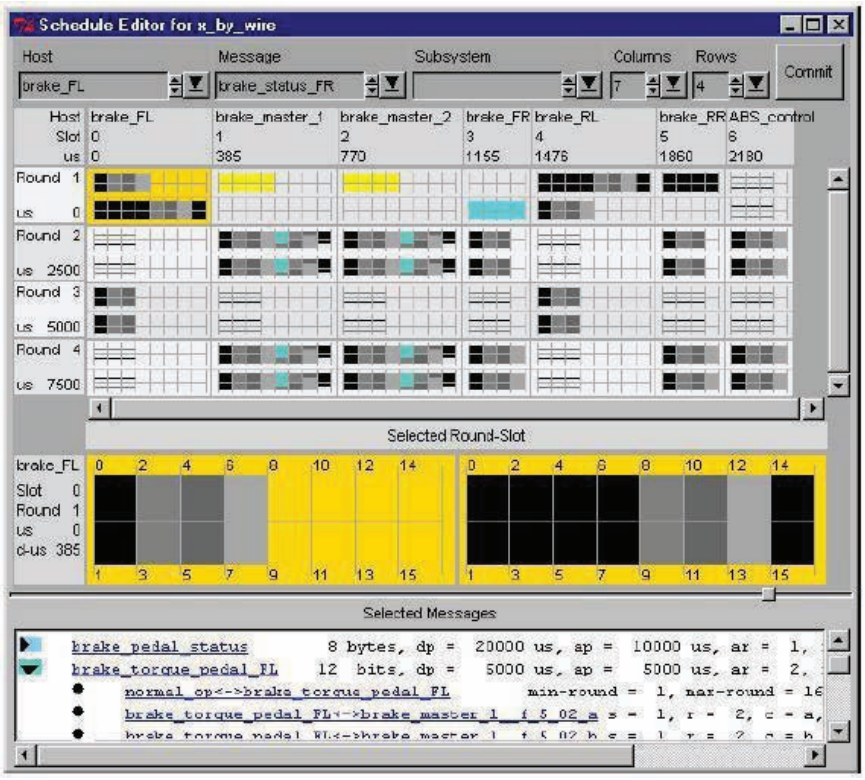


Figure 2. A typical TTP Cluster shown in TTPlan®.
Adapted from TTTech Computer Technik AG.

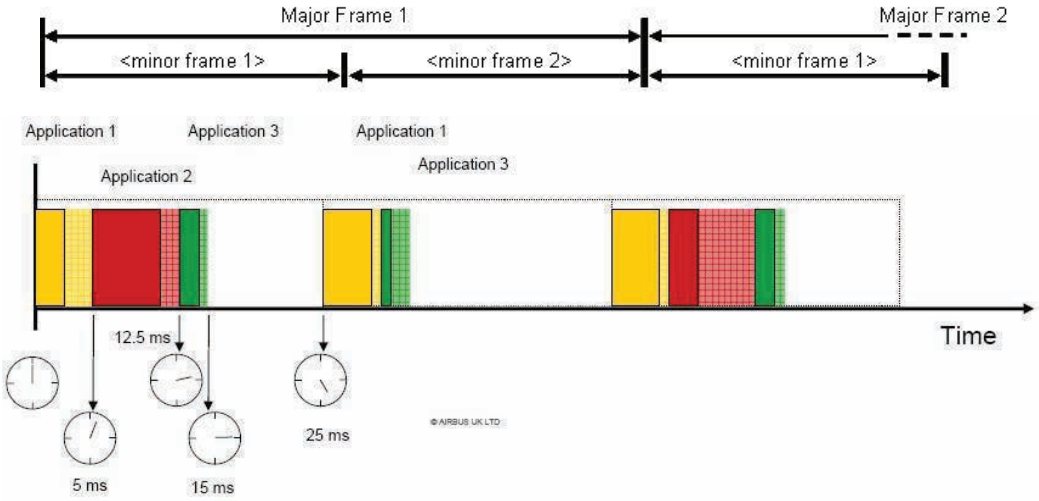


Figure 3. A typical temporal partitioning in ARINC-653.
Adapted from ARTIST2 – Integrated Modular Avionics A380.

Figure 3 shows one temporal partition configuration [6], where three applications (in colors yellow, red and green) occupy limited time windows within what is frequently called "minor frames", where they are allowed to run until they either finish execution (spare time is indicated by a "checkered" colored pattern) or become suspended in favor of the next partition. Note that application "red" is allowed to run for time 5 to time 12.5 milliseconds in the first "minor-frame", while applications "yellow" and "green" are allowed to run in the first and second "minor-frames". Not also that the temporal behavior repeats itself every two "minor-frames" in what the standard calls "Major-Frame".

6. CHALLENGES AND OPPORTUNITIES

System designers willing to adopt AFDX, TTP and ARINC-653 standards need to decide:

- ✓ For the AFDX network configuration: How many Virtual Links? How many AFDX Switches? What Bandwidth Allocation Gaps (BAGs) to choose?
- ✓ For the TTP Cluster configuration: How many Slots? How many Rounds? How big the messages should be?
- ✓ For the ARINC-653 processing load distribution: How many processing modules? How many partitions per module? How long should be the duration of each partition?

Further, issues about and time, as well as task synchronization should also be addressed and decided upon.

As for researchers, new design challenges offer great opportunities for:

- ✓ Extending current analysis methods: "Timing Analysis" should be able to analyse the entire distributed system, from task scheduling in processing modules to delays in message transmission and reception.
- ✓ Creating configuration and optimization tools: The bigger the system, the more automation should be in place for configuring and optimizing it (a lengthy analysis process is never practical).
- ✓ Pursuing further studies on time synchronization in distributed systems: Should hardware support for time synchronization always be in place? How good time synchronization using exclusively software should be?

In addition, the fact that the ARINC-653 standard offers little guidance for "extra-cabinet" communication offers another great opportunity for studies that eventually may fill in this important gap, avoiding OS supplier specific implementation and securing, true application portability.

7. CONCLUSION

With the introduction of IMA, new industry standards such as AFDX, TTP and APEX present new challenges for system manufacturers and integrators, but more important, opportunities for new academic endeavors, such as:

- ✓ Composition of AFDX, TTP and ARINC-653, extending current "Holistic Timing Analysis" methods for evaluating task WCRT/BCRT and message transmission jitter;
- ✓ Include "Data Aging" in the analysis;
- ✓ "Fill-in-the-blanks" where ARINC-653 left physical I/O unmapped (materialize "Pseudo Partitions");
- ✓ Formally evaluate and recommend standard Time Synchronization Protocols for distributed systems, such as Network Timing Protocol (NTP) and IEEE-1588 Precision Time Protocol in time-critical applications;
- ✓ Could a "Virtual Machine Hypervisor" be an alternative to ARINC-653?

In the years to come, more theoretical studies shall be required as time-critical applications developed using AFDX, TTP and ARINC-653 mature.

8. REFERENCES

[1] RADIO TECHNICAL COMMISION FOR AERONAUTICS - RTCA DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations. Washington D.C., USA, November 2005.

[2] AERONAUTICAL RADIO INCORPORATED - ARINC specification 664: Aircraft Data Networks – Part 7, Avionics Full-Duplex Switched Ethernet (AFDX) Network, June 27th, 2005.

[3] TTA-GROUP. Time-Triggered Protocol (TTP/C) High-Level Specification Document – Protocol Version 1.1, Edition 1.4.3, Vienna, Austria, November 19th, 2003.

[4] AERONAUTICAL RADIO INCORPORATED. ARINC specification 653P1-2: Avionics Application Software Standard Interface Part 1 – Required Services, December 1st, 2005.

[5] Kopetz H. and Ochsenreiter W. 1987. Clock Synchronization in Distributed Real-Time. In IEEE TRANSACTIONS ON COMPUTERS, Vol. C-36, No. 8 (August 1987), 933-940.

[6] Itier J.B. 2007. A380 Integrated Modular Avionics. http://www.artist-embedded.org/docs/Events/2007/IMA/Slides/ARTIST2_IMA_Itier.pdf. Accessed in November 24th.