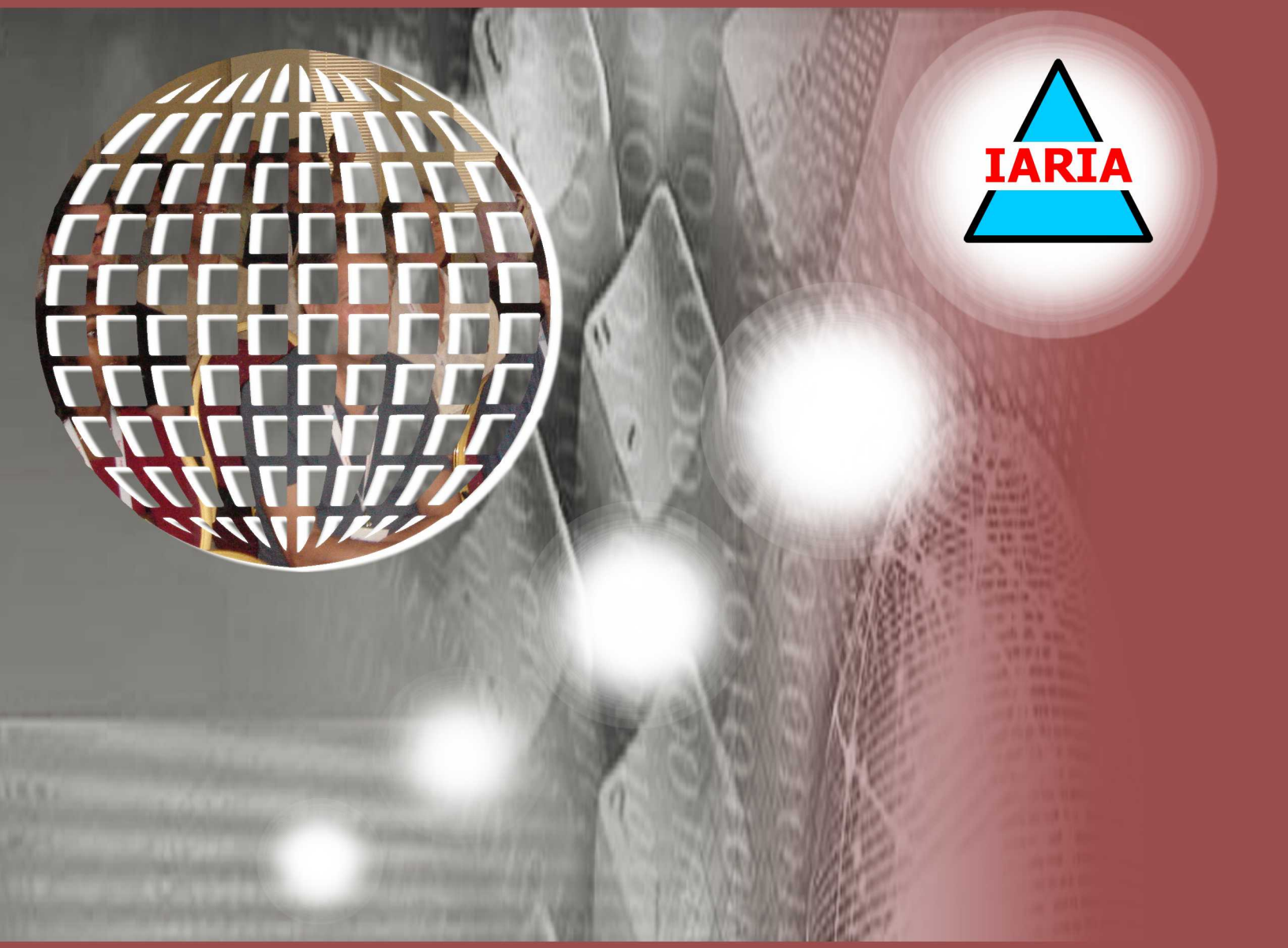
# International Journal on Advances in Telecommunications




2010 vol. 3 nr. 1&2

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



Sponsored by IARIA
www.iaria.org

**International Journal on Advances in Telecommunications**
**Volume 3, Number 1 & 2, 2010**

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

**Communication Theory, QoS and Reliability**

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

**Wireless and Mobile Communications**

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

**Systems and Network Communications**

- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Anne-Marie Bosneag, Ericsson Ireland Research Centre, Ireland
- Huaqun Guo, Institute for Infocomm Research, A*STAR, Singapore
- Jong-Hyouk Lee, INRIA, France
- Elizabeth I. Leonard, Naval Research Laboratory – Washington DC, USA
- Sjouke Mauw, University of Luxembourg, Luxembourg
- Reijo Savola, VTT, Finland

**Multimedia**

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

**Space Communications**

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

**Additional reviewers**

- Vasilis Stilianakis, University of Patras, Greece

**International Journal on Advances in Telecommunications**
**Volume 3, Numbers 1 & 2, 2010**

**CONTENTS**

# Narrowband Interference Suppression for MIMO MB-OFDM UWB Communication Systems

Georgi Iliev
Department of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
e-mail: gli@tu-sofia.bg

Zlatka Nikolova
Department of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
e-mail: zvv@tu-sofia.bg

Vladimir Poulkov
Department of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
e-mail: vkp@tu-sofia.bg

Miglen Ovtcharov
Department of Telecommunications
Technical University of Sofia
Sofia, Bulgaria
e-mail: miglen.ovcharov@jci.com

***Abstract*—Ultrawideband (UWB) systems show excellent potential benefits when used in the design of high-speed digital wireless home networks. The constantly-increasing demand for higher data transmission rates can be satisfied by exploiting both multipath- and spatial-diversity, using multiple-input multiple-output (MIMO) together with proper modulation and coding techniques. Unlike conventional MIMO OFDM systems, the performance of MIMO MB-OFDM UWB systems does not depend on the temporal correlation of the propagation channel although narrowband interference (NBI) is still a problem. In this paper we put forward a technique for suppressing NBI by the use of adaptive narrowband filtering. The method is compared experimentally with other algorithms for the identification and cancellation/suppression of complex NBI in OFDM single-band and multiband (MB) UWB systems. The study shows that the different schemes offer slightly differing performances, depending on the parameters of the MIMO UWB system. The proposed complex adaptive narrowband filtering technique is an optimal solution which offers a good balance between NBI suppression efficiency and computational complexity.**

***Keywords*** **- Narrowband interference (NBI); Multiband orthogonal frequency-division multiplexing (MB-OFDM); Multiple-input multiple-output (MIMO); Ultrawideband (UWB); Variable complex filters; Adaptive complex filter banks.**

## I. INTRODUCTION

In recent years, ultrawideband (UWB) signals have been employed extensively in communications and ranging applications. Depending on how the available bandwidth of the system is used, UWB can be divided into two groups: single-band and multiband.

Conventional UWB technology is based on single-band systems and employs carrier-free communications [1]–[3]. It is implemented by directly modulating information into a sequence of impulse-like waveforms; support for multiple users is by means of time-hopping or direct sequence spreading approaches.

The UWB frequency band of multiband UWB systems is divided into several sub-bands, with the bandwidth of each of them being at least 500 MHz [4] [5]. By interleaving the symbols across sub-bands, multiband UWB can maintain the power of the transmission as though a wide bandwidth were being utilized. The advantage of the multiband approach is that it allows the information to be processed over a much smaller bandwidth, thereby reducing overall design complexity as well as improving spectral flexibility and worldwide adherence to the relevant standards. In order to capture the multipath energy efficiently, the orthogonal frequency division multiplexing (OFDM) technique is used to modulate the information in each sub-band. The major difference with MB-OFDM, as opposed to more traditional OFDM schemes, is that the MB-OFDM symbols are not continuously sent on one frequency-band; instead they are interleaved over different sub-bands across both time and frequency. Multiple access of multiband UWB is enabled by the use of suitably-designed frequency-hopping sequences over the set of sub-bands.

Most UWB applications are used indoors, thus providing an excellent transmission environment for MIMO implementation. Moreover, the GHz center frequency of UWB systems makes the spacing between antenna array elements less critical. In consequence, the combination of UWB and MIMO technology becomes an effective method of achieving the very high data rates required for high-speed short-range communications. Multi-antenna UWB technology has been well-explored in conventional single-band UWB systems [6]–[8]. Conversely, research into multiband UWB systems employing multiple antennae is as yet not complete, so the full benefits of UWB-MIMO communications systems have therefore not been entirely explored.

The main difference between MIMO OFDM and MIMO MB-OFDM lies in the channel characteristics. The block diagram for MIMO MB-OFDM is shown in Figure 1.

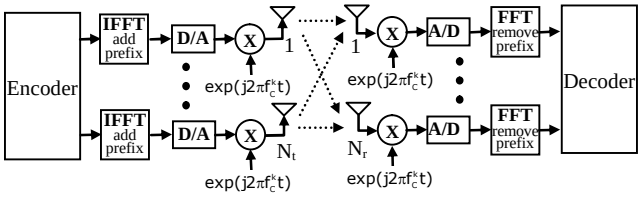



Figure 1. MIMO MB-OFDM UWB system.

In this paper, we look at UWB MIMO systems with MB-OFDM, a leading technology for many broadband communication systems [9]. Due to their relatively low transmission power, such systems are very sensitive to narrowband interference (NBI). Because of the spectral leakage effect caused by Discrete Fourier Transform (DFT) demodulation at the OFDM receiver, many subcarriers near the interference frequency suffer from serious Signal-to-Interference Ratio (SIR) degradation, which can adversely affect or even block communications [10].

The issue of NBI suppression in wideband OFDM systems is of primary importance in such systems and has been studied extensively in the last few years [11]. Two main types of approach are generally adopted. The first involves various frequency excision methods, whereby the affected frequency bins of the OFDM symbol are excised or their usage avoided [12]. The second approach is related to "cancellation techniques" which are aimed at eliminating or mitigating the effect of the NBI on the received OFDM signal [13]. In most cases, the degradation in a MIMO MB-OFDM-based receiver is beyond the reach of the frequency excision method when the SIR is less than 0 dB. Thus, mitigation techniques employing cancellation methods, one of which is based on complex adaptive filtering and the other on NBI identification and cancellation, are recommended as an alternative [14]-[17].

In this paper, a method for NBI cancellation based on adaptive complex digital filtering, using the Least Mean Squares (LMS) algorithm to adapt to the central frequency of the NBI [18], is presented and the method is compared to other mitigation techniques. The study shows that all schemes give different performances, depending on the parameters of the MB-OFDM MIMO UWB system, but the proposed method offers considerable benefits, including low computational complexity.

The rest of the paper is organized as follows: In Section II the adaptive filtering scheme using variable complex adaptive narrowband filter is considered. An adaptive complex notch filter bank for the cancellation/enhancement of multiple complex signals is also proposed. The simulation model for NBI suppression in UWB systems is described in Section III. Section IV presents the simulation results for NBI suppression in UWB channels and MIMO MB-OFDM systems. Finally, Section V concludes the paper.

## II. NBI Suppression using Adaptive Complex Filtering

In comparison with the information wideband signal, the interference occupies a much narrower frequency band but has a higher-power spectral density [19]. On the other hand, the wideband signal usually has autocorrelation properties quite similar to those of AWGN (Adaptive Wide Gaussian Noise), so filtering in the frequency domain is possible.

### A. Variable Complex Narrowband Digital Filter

The filtering process is carried out at the input of the OFDM demodulator by the use of a variable complex filter with independent tuning of the central frequency and bandwidth. This is then turned into an adaptive narrowband filter to be implemented in an OFDM receiver.

A variable complex bandpass (BP) first-order digital filter designated LS1 (Low Sensitivity) is designed [20] (Figure 2). The transfer functions of the LS1 section, all of BP type, are:

$$H_{RR}(z)=H_{II}(z)=\frac{\hat{\beta}+2\hat{\beta}^2\cos\theta z^{-1}+\hat{\beta}(2\hat{\beta}-1)z^{-2}}{1+2(2\hat{\beta}-1)\cos\theta z^{-1}+(2\hat{\beta}-1)^2 z^{-2}},$$
$$H_{RI}(z)=-H_{IR}(z)=\frac{2\hat{\beta}(1-\hat{\beta})\sin\theta z^{-1}}{1+2(2\hat{\beta}-1)\cos\theta z^{-1}+(2\hat{\beta}-1)^2 z^{-2}}. \quad (1)$$

The composed multiplier $\hat{\beta}$ is $\hat{\beta}=\beta+2\beta(\beta-1)$.

The bandwidth can be tuned by trimming the single coefficient $\beta$, whereas $\theta$ controls the central frequency $\omega_0$.

This design of variable complex digital filter has two very important advantages: firstly, an extremely low passband sensitivity which offers resistance to quantization effects; secondly, independent control of central frequency and filter bandwidth over a wide frequency range.

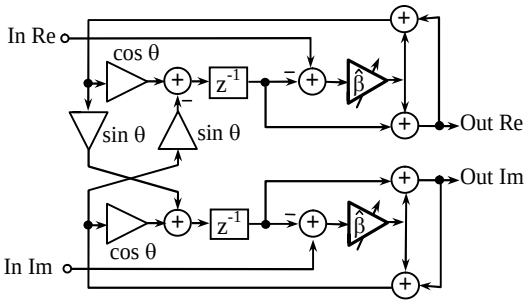



Figure 2. Variable complex BP LS1 filter.

### B. Adaptive Complex Filtering

In Figure 3, an adaptive complex notch/BP narrow-band system based on the LS1 variable complex filter is shown [18].

In the following, we consider the input/output relations for corresponding BP/notch filters Eq.(2)-(9). For the BP filter we have the following real output:

$$y_R(n)=y_{R1}(n)+y_{R2}(n), \quad (2)$$

where

$$y_{R1}(n) = -2(2\beta-1)\cos\theta(n)y_{R1}(n-1) - \\ -(2\beta-1)^2 y_{R1}(n-2) + 2\beta x_R(n) + \\ +4\beta^2\cos\theta(n)x_R(n-1) + 2\beta(2\beta-1)x_R(n-2), \quad (3)$$

$$y_{R2}(n) = -2(2\beta-1)\cos\theta(n)y_{R2}(n-1) - \\ -(2\beta-1)^2 y_{R2}(n-2) - \\ -4\beta(1-\beta)\sin\theta(n)x_I(n-1). \quad (4)$$

$y_R$ is the real output and $x_R$ is the real input.

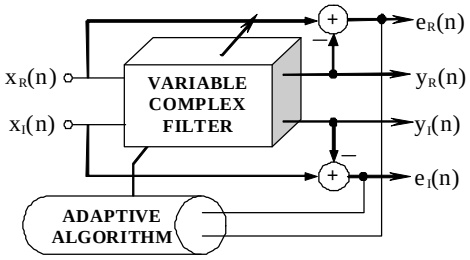



Figure 3. Block-diagram of a BP/notch adaptive complex filter section.

The imaginary output is given by the following equation:

$$y_I(n) = y_{I1}(n) + y_{I2}(n). \quad (5)$$

$$y_{I1}(n) = -2(2\beta-1)\cos\theta(n)y_{I1}(n-1) - \\ -(2\beta-1)^2 y_{I1}(n-2) + \\ +4\beta(1-\beta)\sin\theta(n)x_R(n-1), \quad (6)$$

and

$$y_{I2}(n) = -2(2\beta-1)\cos\theta(n)y_{I2}(n-1) - \\ -(2\beta-1)^2 y_{I2}(n-2) + 2\beta x_I(n) + \\ +4\beta^2\cos\theta(n)x_I(n-1) + 2\beta(2\beta-1)x_I(n-2), \quad (7)$$

where $y_I$ is the imaginary output and $x_I$ is the imaginary input.

For the notch filter we have a real output:

$$e_R(n) = x_R(n) - y_R(n), \quad (8)$$

and imaginary output:

$$e_I(n) = x_I(n) - y_I(n). \quad (9)$$

The cost function is the power of the notch filter output signal:

$$[e(n)e^*(n)], \quad (10)$$

where

$$e(n) = e_R(n) + je_I(n). \quad (11)$$

The LMS algorithm is then applied to update the filter coefficient responsible for the central frequency as follows:

$$\theta(n+1) = \theta(n) + \mu\,\mathrm{Re}[e(n)y'^*(n)]. \quad (12)$$

$\mu$ is the step size controlling the speed of convergence, (*) denotes complex-conjugate, $y'(n)$ is the derivative of $y(n) = y_R(n) + jy_I(n)$ with respect to the coefficient subject of adaptation, where

$$y'_R(n) = 2(2\beta-1)\sin\theta(n)y_{R1}(n-1) - \\ -4\beta^2\sin\theta(n)x_R(n-1) + \\ +2(2\beta-1)\sin\theta(n)y_{R2}(n-1) - \\ -4\beta(1-\beta)\cos\theta(n)x_I(n-1) \quad (13)$$

and

$$y'_I(n) = 2(2\beta-1)\sin\theta(n)y_{I1}(n-1) + \\ +4\beta(1-\beta)\cos\theta(n)x_R(n-1) + \\ +2(2\beta-1)\sin\theta(n)y_{I2}(n-1) - \\ -4\beta^2\sin\theta(n)x_I(n-1). \quad (14)$$

In order to ensure the stability of the adaptive algorithm, the range of the step size $\mu$ should be set according to [21]

$$0 < \mu < \frac{P}{L\sigma^2}. \quad (15)$$

In this case L is the filter order, $\sigma^2$ is the power of the signal $y'(n)$ and P is a constant which depends on the statistical characteristics of the input signal. In most practical situations K is approximately equal to 0.1.

This approach can easily be extended to the complex adaptive narrow-band filter bank (Figure 4) [22].

The notch filter bank output signal is described by the following formulae:

$$e_{R_{FB}}(n) = x_R(n) - \sum_{i=1}^{M} y_{Ri}(n), \quad (16)$$

$$e_{I_{FB}}(n) = x_I(n) - \sum_{i=1}^{M} y_{Ii}(n), \quad (17)$$

$$e_{FB}(n) = e_{R_{FB}}(n) + je_{I_{FB}}(n), \quad (18)$$

where $M=N_R$ is the number of the receiver's antennae.

The LMS algorithm is applied to adapt the filter bank coefficients [21]:

$$\theta_i(n+1) = \theta_i(n) + \mu\,\mathrm{Re}[e_{FB}(n)y'^*_i(n)], \text{ for } i = 1 \div M. \quad (19)$$

The main advantages of both the adaptive structure and the filter bank lie in their low computational complexity and fast convergence. The very low sensitivity of the variable complex filter section ensures a high tuning accuracy, even with severely quantized multiplier coefficients and the general efficiency of the adaptation [22].

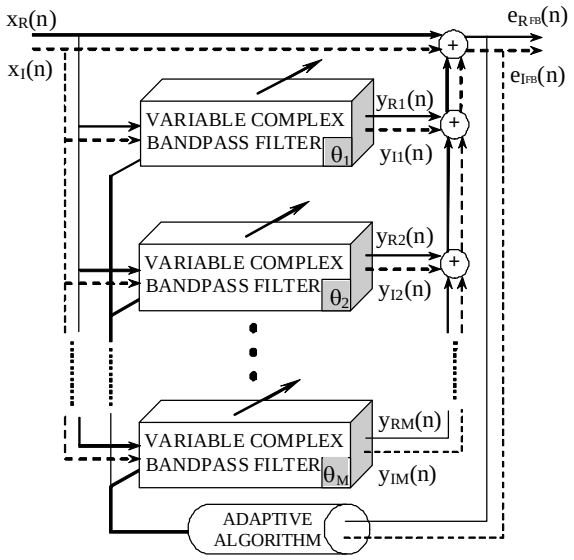



Figure 4. Adaptive complex notch filter bank for the cancellation/enhancement of NBI.

## III. SIMULATION MODEL FOR NBI SUPPRESSION IN UWB SYSTEMS

In order to compare the proposed method with other NBI suppression methods, such as Frequency Excision (FE) [23] and Frequency Identification and Cancellation (FIC) [17], a number of simulations relative to complex baseband presentation are performed.

The FE method is applied to the OFDM signal with a complex NBI at the input of the demodulator. The signal is then converted into the frequency domain by FFT, oversampled by 8 and the noise peaks in the spectra of the signal are limited to the determined threshold. The signal is subsequently converted back into the time domain and applied to the input of the demodulator. It should be noted that, for more precise frequency excision, FFT of a higher order than the one in the demodulator is applied.

The FIC method is implemented as a two-stage algorithm. First, the complex NBI frequency is estimated by finding the maximum in the oversampled signal spectrum. Next, using the ML approach, the NBI amplitude and phase are estimated. The second stage realizes the NLS optimization algorithm, where precise estimations of NBI complex amplitude, phase and frequency are performed.

For the realization of the NBI filtering method, the complex adaptive notch filter is connected at the receiver's input. The adaptation algorithm tunes the filter in such a way that its central frequency and bandwidth match the NBI signal spectrum. In the simulations, the central frequency of the notch filter is chosen to be equal to the NBI central frequency, while its bandwidth is equal to 20% of the bandwidth between two adjacent OFDM sub-carriers.

In the OFDM demodulator, the prefix and suffix guard intervals are removed and a 256-point FFT is applied. The pilot tones are removed and a channel equalization of the OFDM symbol is performed. Finally, the corresponding 64-QAM demodulation and decoding is carried out.

The information source is modeled by a generator of uniformly distributed random integers based on the modified version of Marsaglia's "Subtract with borrow algorithm" [24]. The method can generate all the double-precision values in the closed interval $[2^{-53}, 1\text{-}2^{-53}]$. Theoretically, this method can generate over $2^{1492}$ values before repeating itself.

The channel encoder is implemented as a convolutional encoder. In the simulation, the code rate: $R_C = 1/2$ is chosen. In the receiver, a Viterbi hard threshold convolutional decoder is implemented.

A block interleaver-deinterleaver is used in the simulation. The algorithm chooses a permutation table randomly, using the initial state input that is provided.

The digital modulator is implemented as 256-point IFFT. In the OFDM block demodulator, the prefix and suffix guard intervals are removed from each channel and 256-point FFT is also applied. The OFDM symbol consists of 128 data bins and 2 pilot tones. Each discrete piece of OFDM data can use different modulation formats. In the experiments, Grey-encoded 64-QAM modulation format is used. After the IFFT process, the prefix and suffix guard intervals are added.

For the wireless channel, a multi-ray model with direct and delayed (reflected) components is used. The delayed components are subject to fading, while the direct ones are not. To preserve total signal energy, the direct and delayed signal components are scaled by the square roots of K/(K+1) and 1/(K+1) respectively. To simplify simulations, a complex baseband representation of the system is used [25] [26].

The NBI is modeled as a single complex tone, the frequency of which is located centrally between two adjacent OFDM sub-carriers.

## IV. EXPERIMENTS AND SIMULATION RESULTS

### A. NBI Suppression for UWB Channels

Using the above general simulation model, different experiments were performed, estimating the Bit Error Ratio (BER) as a function of the SIR. Four types of channels are considered, i.e., AWGN, CM1, CM2 and CM3 [27] [28]. The CM1, CM2 and CM3 channels are subject to strong fading and additionally background AWGN is applied, so that the signal to AWGN ratio at the input of the OFDM receiver is 20 dB. In Figure 5, a complex AWGN channel is considered. The SIR is varied from -20 dB to 0 dB. It can be seen that for high NBI, where the SIR is less than 0 dB, all methods lead to a significant improvement in performance. The complex adaptive filtering scheme gives better performance than the FE method. This could be explained by the NBI spectral leakage effect caused by DFT demodulation at the OFDM receiver, when many sub-carriers near the interference frequency suffer degradation. Thus, filtering out the NBI before demodulation is better than frequency excision. The FIC algorithm achieves the best result because there is no spectrum leakage, as happens with frequency excision, and there is no amplitude and phase distortion as seen in the adaptive filtering case.

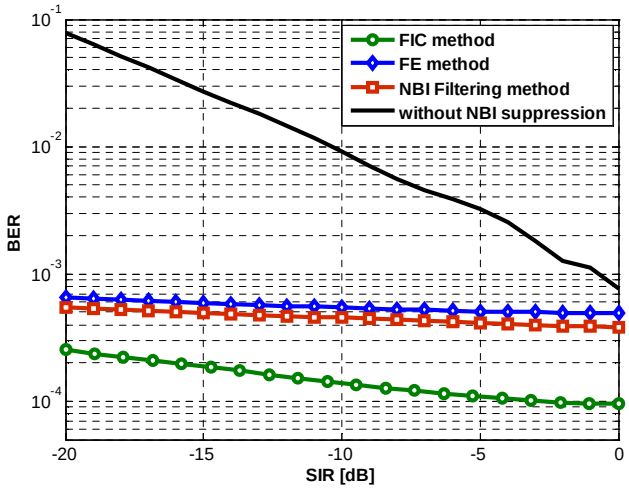



Figure 5. BER as a function of SIR for AWGN channel

In the case of CM1, CM2 and CM3 IEEE UWB channels (Figures 6, 7 and 8) it could be seen that similar results were obtained.

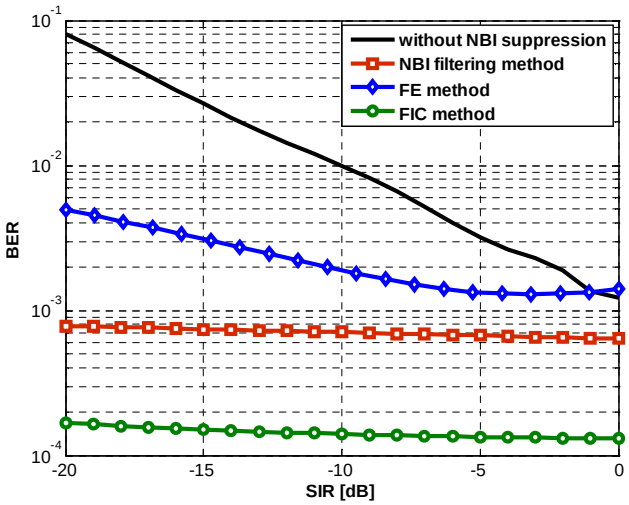



Figure 6. BER as a function of SIR for CM1 channel

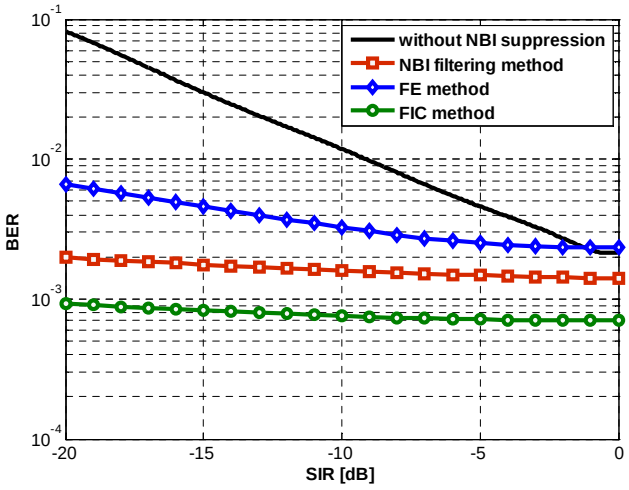



Figure 7. BER as a function of SIR for CM2 channel

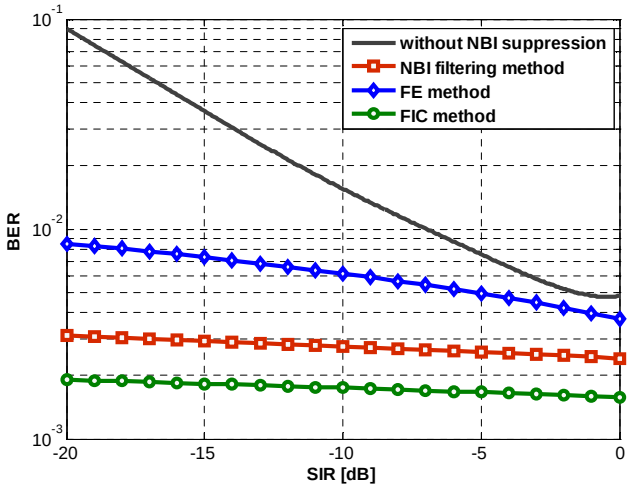



Figure 8. BER as a function of SIR for CM3 channel

It should be noted that the adaptive filtering scheme and frequency cancellation scheme lead to a degradation of the overall performance when SIR > 0. This is due either to the amplitude and phase distortion of the adaptive notch filter, or to a wrong estimation of NBI parameters during the identification. The degradation can be reduced by the implementation of a higher-order notch filter or by using more sophisticated identification algorithms. The degradation effect can be avoided by simply switching off the filtering when SIR > 0. Such a scheme is easily realizable as the amplitude of the NBI can be monitored at the BP output of the filter (Figure 3).

In Figure 9, the results of applying a combination of methods are presented. A multi-tone NBI (5 sine-wave interfering signal) is added to the OFDM signal. One of the NBI tones is 10 dB stronger than the others. The NBI filter is adapted to track the strongest NBI tone, thus preventing the loss of resolution and AGC saturation. It can be seen that the combination of frequency excision plus adaptive filtering improves the performance, and the combination of frequency cancellation plus adaptive filtering is even better.

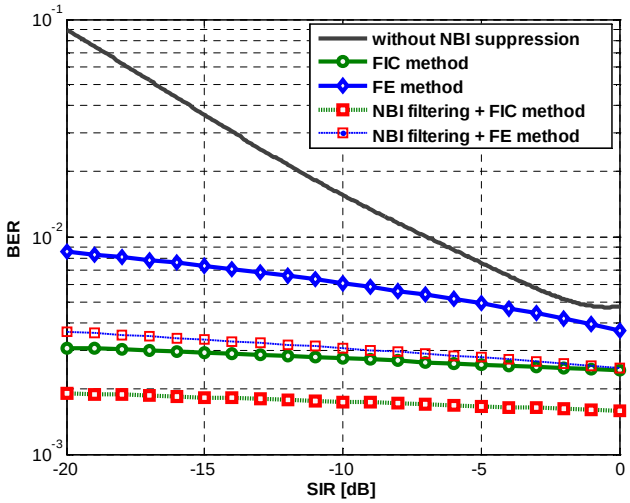



Figure 9. BER as a function of SIR for CM3 channel – multi-tone NBI

Figure 10 shows BER as a function of SIR for the CM3 channel when QPSK modulation is used, the NBI being modeled as a complex sine wave. It can be seen that the relative performance of the different NBI suppression methods is similar to the one in Figure 6 but the BER is higher due to the fact that NBI is QPSK modulated.

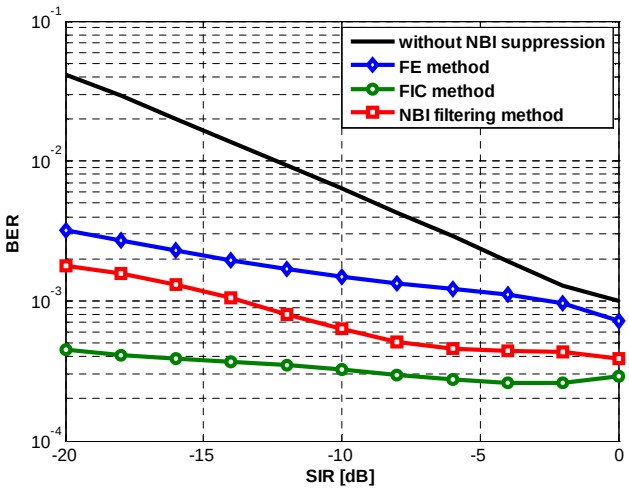



Figure 10. BER as a function of SIR for CM3 channel – QPSK modulated NBI

### B. *NBI Suppression for MIMO MB-OFDM systems*

To evaluate the performance of the three NBI suppression methods, (FE, FIC, and our proposed NBI filtering method), simulations relative to the complex baseband presentation are conducted. A standard MIMO OFDM receiver is assumed and the suppression methods are applied to the MIMO OFDM signal with a complex NBI at each input of the receiver independently.

For the NBI filtering method, the complex adaptive notch filter bank is used. The adaptation algorithm tunes the filter at each receiver input so that its central frequency and bandwidth match the NBI signal spectrum. The Frequency Identification and Cancellation method estimates the complex NBI frequency by determining the maximum in the oversampled signal spectrum per channel.

The OSTBC model for complex signals is realized using the methods described in [29], [30]. The number of transmit antennae $N_T$ can be set from 1 to 4 as long as the number of receive antennae $N_R$ can also be set from 1 to 4. For 2x2 MIMO system, the code rate is Rc=1 whereas for 3x3 and 4x4 MIMO systems the code rate is Rc=1/2.

The MIMO wireless flat fading channel is realized as given in [31]. A system with $N_T$ transmit antennae and $N_R$ receive antennae is considered. It is assumed that the complex channel gain $h_{i,j}$ is a complex Gaussian random variable: $h_{i,j} \sim Nc(0,1)$. As the MIMO channel matrix H is not known for the receiver, it must be estimated before the start of the decoding process. The channel estimation method based on the optimal training preamble [31] is adopted. OSTB decoding, 64-QAM demodulation and error correction decoding are carried out.

Using the above-mentioned simulation model of the Orthogonal Space-Time Block Coding (OSTBC) MIMO system, different experiments were performed to estimate the BER as a function of the SIR. Four types of systems are considered: SISO (1x1), MIMO (2x2), MIMO (3x3) and MIMO (4x4). The MIMO channels are subject to flat fading and, in addition, background AWGN is applied, so that the signal to AWGN ratio at the input of the OFDM receiver is 15dB. In Figure 11, a complex flat fading AWNG channel without NBI suppression is considered. The SIR is varied from -20 dB to 0 dB. It can be seen that 4x4 OSTBC MIMO system gives the best performance.

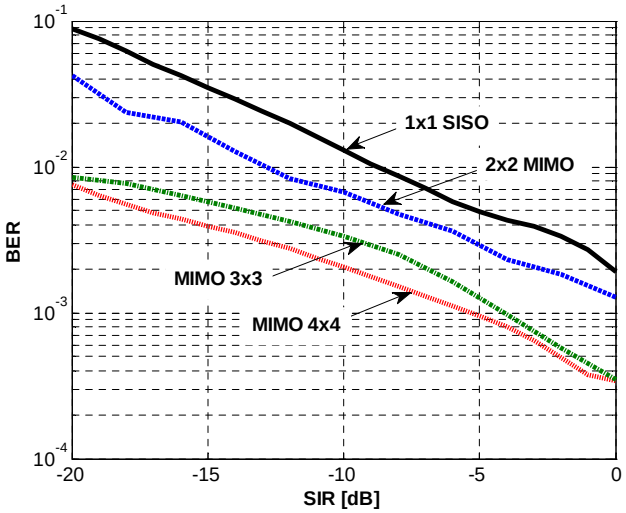



Figure 11. BER as a function of SIR for MIMO channel

Three NBI suppression techniques are then applied: FE, FIC, and the new NBI adaptive filtering method.

In the case of 2x2, 3x3 and 4x4 MIMO channels (Figures 12, 13 and 14) better results in terms of NBI filtering are obtained for higher values of antenna diversity. The FE method manifests good performance for high SIR.

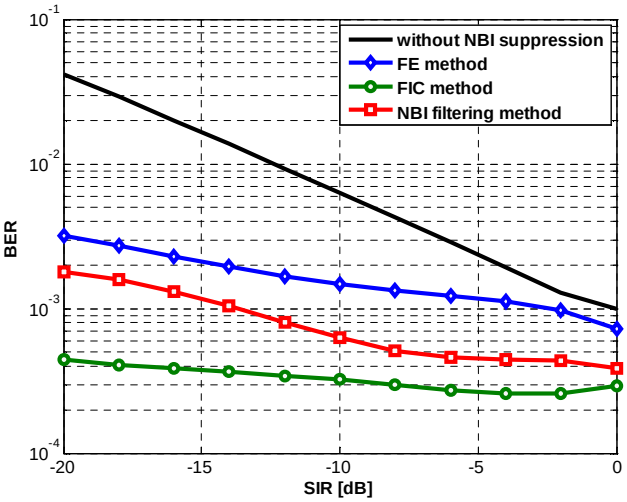



Figure 12. BER as a function of SIR for 2x2 MIMO channel

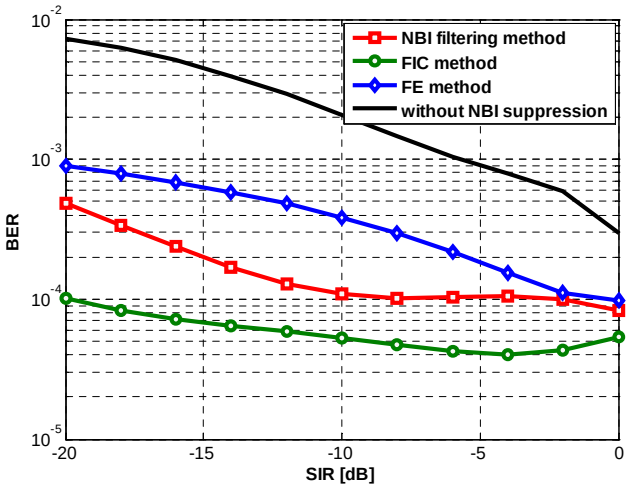



Figure 13. BER as a function of SIR for 3x3 MIMO channel

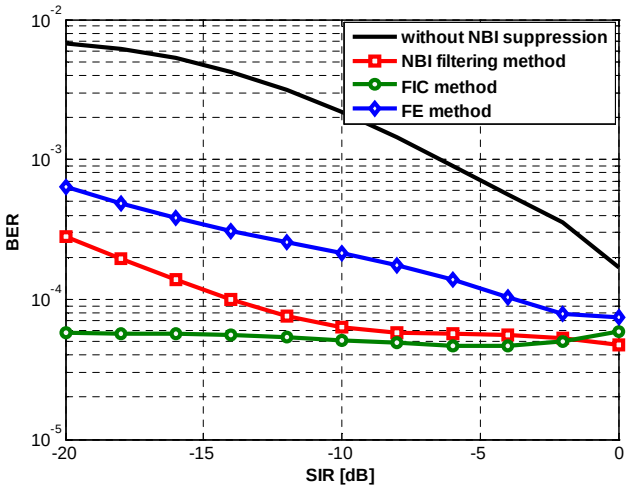



Figure 14. BER as a function of SIR for 4x4 MIMO channel

The experimental results show that the frequency identification and cancellation method achieves the highest performance. On the other hand, the extremely high computational complexity limits its application in terms of hardware resources. In this respect, the adaptive notch filter turns out to be the optimal NBI suppression scheme, as it offers very good performance and reasonable computational complexity. The frequency excision method shows relatively good results and its main advantage is its computational efficiency.

TABLE I. COMPUTATIONAL COMPLEXITY COMPARISON

| Suppression Method | Number of Additions | Number of Multiplications | Complexity |
|---|---|---|---|
| Frequency Excision | 6MN | 4MNlog(N) | ~O(Nlog(N)) |
| NBI Filtering | KMN | $28MN+KMN^2$ | ~O($N^2$) |
| Frequency Identification and Cancellation | $2MN^2$ | $M(N+2)N^3$ | ~O($N^4$) |

The computational complexity per iteration of the modeled NBI suppression methods are listed in Table I, where: K is a positive integer constant, N is the number of samples to be processed, M is the number of receiver's antennas and O() is the complexity estimation function.

## V. CONCLUSIONS

In this paper a method for NBI suppression in MIMO MB-OFDM UWB communication systems, using adaptive complex narrowband filtering, is proposed. In relation to this, a comparison with two other schemes for suppression of complex NBI is performed. The first is a frequency excision method while the second employs frequency identification and cancellation based on ML and NLS algorithms. The experiments show that for high NBI, where the SIR is less than 0 dB, all three suppression methods lead to a significant improvement in performance.

An optimal solution is the adaptive narrowband filtering method, which offers a trade-off between outstanding NBI suppression efficiency and computational complexity. An alternative approach is to implement a combination of the NBI filtering and frequency cancellation methods, thus improving overall performance.

## ACKNOWLEDGMENT

This work was supported by the Bulgarian National Science Fund – Grant No. ДО-02-135/2008 "Research on Cross Layer Optimization of Telecommunication Resource Allocation".

## REFERENCES

[1] M. L.Welborn, "System considerations for ultra-wideband wireless networks," in IEEE Radio Wireless Conf., Aug. 2001, pp. 5–8.

[2] J. R. Foerster, "The performance of a direct-sequence spread ultrawideband system in the presence of multipath, narrowband interference, and multiuser interference," in IEEE Conf. Ultra Wideband Systems Tech., May 2002, pp. 87–91.

[3] N. Boubaker and K. B. Letaief, "Ultra wideband DSSS for multiple access communications using antipodal signaling," in IEEE Int. Conf. Commun., vol. 3, May 2003, pp. 11–15.

[4] E. Saberinia and A. H. Tewfik, "Pulsed and nonpulsed OFDM ultra wideband wireless personal area networks," in IEEE Conf. Ultra Wideband Systems Tech., Nov. 2003, pp. 275–279.

[5] A. Batra, J. Balakishnan, G. R. Aiello, J. R. Foerster and A. Dabak, "Design of a multiband OFDM system for realistic UWB channel environments," IEEE Transactions Microwave Theory and Techiques, vol. 52, No. 9, pp. 2123–2138, Sep. 2004.

[6] W. P. Siriwongpairat, M. Olfat, and K. J. R. Liu, "Performance analysis and comparison of time hopping and direct sequence UWB-MIMO systems," in EURASIP J. Appl. Signal Process. Special Issue on "UWBState of the Art", vol. 2005, Mar. 2005, pp. 328–345.

[7] L. Yang and G. B. Giannakis, "Analog space-time coding for multiantenna ultra-wideband transmissions," IEEE Trans. Commun., vol. 52, no. 3, pp. 507–517, March 2004.

[8] M. Weisenhorn and W. Hirt, "Performance of binary antipodal signaling over the indoor UWB MIMO channel," in IEEE Int. Conf. Commun., vol. 4, May 2003, pp. 2872–2878.

[9] A. Bara, J. Balakrishnan, A. Dabak, R. Gharpurey and J. Lin, "Multi-band OFDM physical layer proposal", IEEE P802.15-04/0493r1-TG3a, Sept. 2004.

[10] A. Giorgetti, M. Chiani and M.Z.Win, "The effect of narrowband interference on wideband wireless communication systems", IEEE Trans. Communications, vol. 53, No.12, pp. 2139-2149, Dec. 2005.

[11] Z. Li, A. M. Haimovich, and H. Grebel, "Performance of ultra-wideband communications in the presence of interference," in Proc. IEEE Int. Conf. Communications, vol. 10, June 2001, pp. 2948–2952.

[12] Alan J. Coulson, "Narrowband interference in pilot symbol assisted OFDM systems", IEEE Trans on Wireless Communications, vol. 3, No. 6, Nov. 2004, pp. 2277-2287.

[13] M. Biagi, A. Buccini and L. Taglione, "Advanced methods for ultra wide band receiver design", Proceedings of the First International Workshop "Networking with UWB", Rome, Italy, 21 Dec. 2001.

[14] C. Carlemalm, H. V. Poor and A. Logothetis, "Suppression of multiple narrowband interferers in a spread-spectrum communication system", IEEE J. Select. Areas Communications, vol.3, No. 5, pp. 1431-1436, Sept 2004.

[15] L. B. Milstein, "Interference rejection techniques in spread spectrum communications", Proc. IEEE, vol. 76, No. 6, pp. 657-671, June 1988.

[16] J. D. Laster and J. H. Reed, "Interference rejection in digital wireless communications", IEEE Signal Proc. Mag., vol. 14, No. 3, pp. 37 – 62, May 1997.

[17] E. Baccareli, M. Baggi and L. Tagilione, "A novel approach to in-band interference mitigation in ultra wide band radio systems", IEEE Conference on Ultra Wide Band Systems and Technologies, 2002.

[18] G. Iliev, Z. Nikolova, G. Stoyanov and K. Egiazarian, "Efficient design of adaptive complex narrowband IIR filters", Proceedings of XII European Signal Processing Conference, EUSIPCO 2004, Vienna, Austria, pp. 1597 - 1600, 6 - 10 Sept. 2004.

[19] S. Y. Park, G. Shor, and Y. S. Kim, "Interference resilient transmission scheme for multi-band OFDM system in UWB channels," in Proc. IEEE Int. Circuits and Systems Symp., vol. 5, Vancouver, BC, Canada, May 2004, pp. 373–376.

[20] G. Stoyanov, M. Kawamata and Z. Valkova, "New first and second-order very low-sensitivity bandpass/bandstop complex digital filter sections", Proc. IEEE Region 10th Annual Conf. "TENCON'97", Brisbane, Australia, vol. 1, Dec. 2-4, 1997, pp. 61-64.

[21] S. Douglas, "Adaptive filtering, in digital signal processing handbook", D. Williams and V. Madisetti, Eds., Boca Raton: CRC Press LLC, pp. 451-619, 1999.

[22] Z. Nikolova, G. Iliev, G. Stoyanov and K. Egiazarian, "Design of adaptive complex IIR notch filter bank for detection of multiple complex sinusoids", Proc. International Workshop SMMSP'2002, Toulouse, France, pp. 155-158, Sept. 2002.

[23] J.-C. Juang, C.-L. Chang and Y.-L. Tsai, "An interference mitigation approach against pseudolites", The 2004 International Symposium on GNSS/GPS, Sydney, Australia, 6 – 8 Dec. 2004.

[24] S. Tezuka, P. L'Ecuyer and R. Couture, "On the lattice structure of the add-with-carry and subtract-with-borrow random number generators", ACM Transactions on Modeling and Computer Simulation (TOMACS), vol. 3, Issue 4, pp. 315 – 331, Oct. 1993.

[25] R. van Nee and R. Prasad, OFDM for wireless multimedia communications, Artech House, 2000.

[26] H. Shulze and C. Luders, Theory and applications of OFDM and CDMA, John Wiley, 2005.

[27] M.-O. Wessman, A. Svensson and E. Agrell, "Frequency diversity performance of coded multiband-OFDM systems on IEEE UWB Channels", COST 289 Workshop, Gothenburg, Sweden, April 2007.

[28] A. F. Molisch and J. R. Foerster, "Channel models for ultra wideband personal area networks", IEEE Wireless Communications, pp. 524-531, Dec. 2003.

[29] S. Alamouti, "Simple transmit diversity technique for wireless communications", IEEE Journal Select. Areas Commun, vol. 16, No. 8, pp. 1451–1458, Oct. 1998.

[30] I. Bernguer and Dong, "Space-time coding and signal processing for MIMO communications", J Comput. Sci &Technol., vol. 18, No 6, pp. 689-702, Nov. 2003.

[31] M. Ovtcharov, V. Poulkov, G. Iliev and Z. Nikolova, "Narrowband interference suppression for IEEE UWB channels", Proc. ICDT 2009, Colmar, France, pp. 43-47.

# Efficient Variable Block Size Selection for AVC Low Bitrate Applications

Ihab Amer and Graham Jullien
Advanced Technology Information Processing Systems (ATIPS)
2500 University Drive, NW
Calgary, AB, Canada, T2N 1N4
e-mails: {amer, jullien}@atips.ca

Wael Badawy
IntelliView Technologies Inc.
808-55 Avenue NE
Calgary, AB, Canada, T2E 6Y4
e-mail: badawy@intelliview.ca

Adrian Chirila-Rus and Robert Turney
DSP Systems Engineering, DSP Division, Xilinx Inc
115 South 4th street
Watertown, WI, USA, 53094
e-mails: {adrian.chirila-rus, robert.turney}@xilinx.com

Rana Hamed
German University in Cairo (GUC)
Main Entrance, Fifth District
New Cairo City, Egypt
email: rana.magdy-hamed@guc.edu.eg

*Abstract*— **The Advanced Video Coding (AVC) standard proposes the usage of Variable Block Size (VBS) motion-compensated prediction and mode decision aiming for an optimized Rate-Distortion (R-D) performance. Unlike Fixed Block Size (FBS) motion-compensated prediction, where all regions of the pictures are treated similarly in terms of temporal prediction, VBS increases the efficiency of encoding by allowing more active regions to be represented with more bits than less active ones. The main concern regarding the usage of VBS motion-compensated prediction is the dramatic increase it adds to the encoder computational requirements, which not only prevents the encoder from satisfying real-time constraints, but also makes it impractical for hardware implementation. This paper presents an efficient VBS selection scheme, which can be applied to any VBS Motion Estimation (ME) module, leading to significant reduction in its computational requirements with minor loss in the quality of the reconstructed picture. The computational requirements reduction is achieved by minimizing the number of required ME searches and simplifying the Mode Decision (MD) operation. In order to meet different applications' demands, the proposed algorithm can be adjusted to function at any of three operating points, trading off computational requirements with R-D performance. In the paper, the algorithm is described in detail, focusing on the theoretical computational requirements savings. This theoretical analysis is then supported with simulation results performed on three benchmark video sequences with various types of motion.**

***Keywords- H.264/AVC, motion estimation, variable block size.***

## I. INTRODUCTION

To achieve a high coding efficiency, AVC deploys a set of new features in addition to enhancing a set of previously used features. In the inter-frame motion prediction, AVC allows for the usage of variable block sizes motion estimation/compensation that can support block sizes of 16×16, 16×8, 8×16 and 8×8 resulting in significant performance improvement. Even more, in the case when an 8×8 mode is chosen, further smaller blocks of sizes 8×4, 4×8 and 4×4 can be used. This method improves the motion tracking capabilities of the encoder by allowing inactive regions and regular movements to be represented with an optimal amount of motion information (e.g., 16×16 mode is represented by one motion vector), while for fast and highly irregular movements, the finer blocks can be used at the cost of increased motion information, but optimal error representation. In terms of computational and memory requirements, motion estimation is by far the most complex module in the entire AVC encoder. As will be shown later, recent studies show that it represents between 70% and 90% of the entire encoder computational requirements. The increase in computational and operational requirements brought by the usage of such new features requires algorithmic and implementation enhancements so that the compression algorithm can become useful in real applications. As a step towards the design and implementation of an entire computationally-efficient AVC encoder, this paper proposes a solution to the increased computational requirements of the AVC variable block size motion estimation/compensation (and mode decision) module. A novel, computationally-efficient algorithm for variable block size motion estimation and mode decision (acting at three operating points that meet various applications' requirements) is developed and described. The

obtained simulation results and analysis show that, for the variety of tested benchmark sequences, the encoder computational requirements can be reduced to less than half, at the expense of "minor" degradation in the quality. The work in this paper further emphasises on the work that has been introduced in [1].

The remainder of this paper is organized as follows: Section II describes the key concept behind the adoption of VBS ME/MD, showing its necessity, and the computationally-expensive way it is implemented in the direct approach of the AVC software reference model. A survey of recent efforts to solve this computational requirements issue is also given. In Section III, the proposed VBS selection scheme is described, accompanied with some theoretical computational requirements analysis to show its effectiveness. Section IV shows and discusses the experimental analysis and results obtained by encoding various benchmark video sequences. Then finally, Section V concludes this paper.

## II. VBS MOTION ESTIMATION AND MODE DECISION

AVC defines seven block sizes for inter-prediction. The seven modes are shown in Figure 1. During the encoding process, an ideal encoder has to examine all these modes to achieve the best inter-prediction, which requires performing all the seven modes of searches, and then examining all the 259 possible combinations of macroblock (MB) partitioning schemes to choose the best one out of them. This VBS motion estimation and mode selection process result in significant performance improvement (in terms of rate-distortion). This is because, on the contrary to Fixed Block Size Motion Estimation (FBSME), where all sub-blocks in a MB have the same size, Variable Block Size (VBS) ME improves the motion tracking capabilities of the encoder by allowing it to give more "attention" to highly active sub-blocks (representing an active region with more sub-blocks would better describe it). However, this leads to dramatic increase in the computational requirements of the encoder compared to traditional FBSME-based encoders.

This section overviews the main concept of VBS ME/MD. It starts with Section A, where the necessity of VBS ME/MD is investigated. Then in Section B, a description of the AVC reference software implementation approach is given. Finally, Section C summarizes some recent efforts to reduce the computational requirements of VBS ME/MD.

### A. Necessity of VBS ME/MD

In this section the necessity of VBS ME is evaluated by first demonstrating its usage when encoding different sequences with different types of motion, then by comparing the reconstructed sequences R-D performance when VBS ME is fully or partially disabled.

Figure 2 shows the occurrence percentage of different block sizes as chosen by the AVC software reference model "Joint Model JM 10.2", running with all the seven block mode types enabled. Full search is used for motion estimation. The test has been performed for a wide range of sequences, starting from QCIF (176×144) up to HD 1080p (1920×1080) and for different motion types.

Table 1 summarizes the characteristics of the sequences that were used in this experiment.

The impact of choosing the variable block size can be clearly seen. For example, the traditional 16×16 has been chosen as the optimal mode for almost 50% of the cases in the "Shields" sequence, a sequence with regular pan movement, while it was as small as 5% for high irregular movement in "Football" sequence, for which the small block sizes (4×4) has been the best option in 40% of the cases.

In addition, a more informative test was performed by running the JM several times for each sequence, disabling one or more inter search/decision mode in each run, then comparing the R-D curves generated from all runs. Figure 3 shows the impact of disabling some of the searching modes on the overall encoder efficiency. The sequence used for this test was Foreman CIF (30 fps). For simplicity, the testing conditions for the encoder runs included the following restrictions: the usage of only one reference frame and the usage of "Low Complexity Mode" for Rate-Distortion optimization. Each of the curves was generated with a different inter-search/decision-mode configuration than the other. Table 2 provides a description of the different cases used to create Figure 3. Any single curve in the graph represents a typical relation between rate and distortion in digital video coding. As expected, the two parameters are inversely proportional. This is reasonable since the better quality needed to be preserved in an encoded video sequence, the more bits required to represent the generated bitstream.

Rate is typically represented by the minimum number of bits per seconds required to transmit the generated bitstream without affecting the continuity of the reconstructed sequence. It depends on two parameters: the size of the generated bitstream, and the resolution/frame rate of the specific sequence. The quality of the reconstructed sequences is measured by the most widely-used *Peak Signal to Noise Ratio* (PSNR) parameter. Due to its easy calculation, PSNR is the most famous objective quality measurement in video coding. It is measured on a logarithmic scale and depends on the "Mean Squared Error" (MSE) between an original and an impaired image or video frame as shown in (1) and (2). *F(i,j)* and *f(i,j)* are the values of pixel *(i,j)* in the original and reconstructed images or video frame of resolution $N \times M$, while $(2^n-1)$ is the highest-possible signal value in the video frame, where $n$ is the number of bits per frame sample.

$$MSE = \frac{\sum_{\forall i,j} [f(i,j) - F(i,j)]^2}{N \times M} \quad (1)$$

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (2)$$

Figure 3 shows that the blue curve (Case 0, which uses all the variable block sizes) is the optimum one in terms of R-D performance. The brown curve (Case 5, where only a fixed 4×4 search/decision mode is used) has a relatively poor performance, especially at low bitrates (around 1 Mb/s), as most of the residual data is cut-off by the coarse quantization process (high QP value), giving more influence to the size of the encoded motion info on the output bitrate. Besides, the light-blue curve (Case 3, where only a fixed 16×16 search/decision mode is used) also shows a poor performance, especially at high bitrates (around 6 Mb/s), when the savings achieved by representing each macroblock by only one motion vector becomes negligible when compared to the overhead of encoding the residuals, which becomes higher for macroblocks with more "activity" inside.

Figure 4 shows that the generated R-D curves for all the tested sequences tend to demonstrate a relatively similar behaviour to what has been discussed above at different bitrates (the higher the sequence resolution, the higher the required bitrate).

In conclusion, VBS ME/MD is an important tool that has been used in AVC to optimize the R-D performance of the encoder. It allows inactive regions and regular movements to be represented with an optimal amount of motion information (e.g., 16×16 blocks), while for fast and highly irregular movements, the finer blocks are used (e.g., 4×4 blocks), achieving optimal error representation, at the cost of increased motion information.

### B. Direct Implementation of VBS ME/MD

A feature of the way AVC reference software (JM) adopts *Rate-Distortion Optimization* (RDO) is by performing an exhaustive search for the "best" partitioning scheme (the one that gives the lowest R-D Cost) among all the 259 ($4^4$ (8×8 or smaller) sub-modes + 1 (8×16) + 1 (16×8) + 1 (16×16) modes) possible combinations. This can be shown in Figure 5 [2]. Hence, the JM searches exhaustively for the minimum possible distortion that can be achieved subject to a specific rate constraint. This can be expressed as follows: Find min(D) subject to $R<R_c$, which can be elegantly solved using *Lagrangian optimization* where a distortion term is weighted against a rate term as follows [3]:

$$\text{Find min(J), where } J = D + \lambda R \quad (3)$$

where J is the R-D Cost that needs to be minimized. D is the *Distortion*. JM uses the *Sum of Absolute Differences* (SAD) as the metric of distortion because of its less computational requirements compared to other metrics. SAD is calculated using (4), where F(i,j) and f(i,j) are the values of pixel (i,j) in the reference and the candidate block respectively. R represents the *Rate*. The way to estimate the rate differs based on the type of rate-distortion optimization scheme being used. For *high-complexity mode*, JM uses the exact number of bits for header, motion info, and transform coefficients (the way they look after the last encoding stage) as the metric of rate as shown in (5). λ (the Lagrange parameter that determines the importance of rate with respect to distortion) is exponentially related to the Quantization Parameter (QP) after multiplying it with a double-precision weight as shown in (6).

$$SAD = \sum_{\forall i,j} \left| f(i,j) - F(i,j) \right| \quad (4)$$

$$R = R_{\text{header (exact)}} + R_{\text{motion (exact)}} + R_{\text{coefficients (exact)}} \quad (5)$$

$$\lambda = \text{weight} \times 2^{(QP-12)/3} \quad (6)$$

For *low-complexity mode*, motion cost calculation is performed using a less complex scheme. A lookup table (LUT) is used to roughly estimate the number of bits needed to encode the difference between the motion vectors and the predictors as shown in (7). λ is also estimated roughly from QP by a lookup table as shown in (8).

$$R = LUT_{MV\_COST}[MV_{cand} - MV_{pred}] \quad (7)$$

$$\lambda = LUT_{QP2QUANT}[\max(0, QP-12)] \quad (8)$$

For simplicity reasons, and practical hardware implementation, any reference to rate-distortion optimization in the remaining part of this paper is a reference to the low-complexity mode, unless clearly mentioned otherwise. Recent complexity analysis was performed in [4] to estimate the distribution of complexity among different modules of the AVC encoder. A typical profile of the encoder complexity (by files) based on an Intel® Pentium™ III 1.0 *GHz* general purpose processor with 512 MB of memory is shown in the pie chart of Figure 6(a). Also, another run-time complexity analysis was performed in [6], where the reference software (Baseline profile) was executed on a Sun® Blade™ 1000 with UltraSPARC™ III 1 *GHz* processor. The run time percentage (approximated to the nearest integer value) of each module is shown in the pie chart of Figure 6(b). The figure shows that the computational requirements of motion estimation/compensation typically represent from 70% to 90% of the overall encoder computational requirements for a typical AVC encoder, which makes it a primary candidate for hardware acceleration. Besides, this module is the main target of encoder computational requirements reduction for most researchers. Most of the efforts aim at introducing "reasonable" quality degradation as an expense of computational requirements savings compared to the optimal (yet very complex) exhaustive-search solution. The next section surveys some of those efforts.

### C. Recent Efforts to Reduce Complexity of VBS ME/MD

Many algorithms for fast ME and MD have been proposed in the literature. Most of them rely on the fact that human's visual system is typically insensitive to the degradation in PSNR that is less than 0.2 *dB*. Hence, they tend to reduce the computational requirements, at the expense of minor "unrecognizable" quality degradation. In this section, an overview of the recent efforts to reduce the VBS ME/MD in AVC is given. Some of the algorithms that have been introduced recently rely on exploiting video

features such as texture and edge information to predict the best possible mode. As an example, Wu et al [7] proposed to adjust the block sizes based on the homogeneity of the region in the MB. They observed that homogenous regions, which are determined by using the magnitude of the edge vector, tend to move together, and hence should not be split into smaller blocks. The results they provided indicate that the technique is not as effective for dynamic sequences as it is for inactive ones. They achieved a maximum computational requirements reduction of 45% for inactive sequences, but it did not exceed 10% reduction for active sequences. The tests they performed were on low resolution sequences only (QCIF and CIF), which leaves the effectiveness of their algorithm with high resolution sequences questionable. Lin et al [2] developed a combined algorithm for fast motion estimation and mode decision by exploiting the motion and cost information available from blocks that have been processed prior to the current block. The authors proposed to predict the mode of a current MB based on previously coded MBs, then perform a fast ME for this predicted mode. If the resulting cost is less than an adaptively calculated threshold, the algorithm skips any further calculations for this block. Although the authors did not investigate the potential of their algorithm for HW implementation, the way they described it makes it less likely to be implemented in hardware. A possible candidate HW design should be able to handle the worst case scenario, which is to calculate all the complex R-D costs for all the seven modes. This task is too complex to be executed within a reasonable number of clock cycles that fits typical encoder systems requirements unless a huge (non realistic) design is implemented. The authors did not provide any computational requirements analysis to justify the time savings they are claiming. They also claim that they observed a maximum PSNR degradation of 0.4 *dB* in their experiments on CIF sequences, which is big enough to generate visually noticeable defects. Yu et al [8] proposed a strategy that incorporates temporal similarity detection as well as the detection of different moving features within a macroblock. Their experiments were performed on QCIF resolution only, and they showed up to 9% bitrate overhead with highly dynamic sequences. The authors did not provide any visual justification of the quality they are claiming. Also they did not show any potential for hardware implementation. In [9], the author proposed an algorithm that relies mainly on two predictive factors: intrinsic complexity of the MB and mode knowledge from the previous frame. The algorithm added more than 5% bitrate overhead for dynamic sequences, and the maximum achieved time saving for all the tested QCIF and CIF sequences was 30%.

Tourapis et al [10] proposed to extend and adapt the concept of the Enhanced Predictive Zonal Search (EPZS) motion estimation algorithms within the AVC standard. They proposed additional modifications to the predictor consideration, and introduced a new refinement pattern and a new iterative refinement process to improve the efficiency of the algorithms. The results of their experiments on selected CIF sequences showed a speedup in the motion estimation process (without providing enough computational requirements analysis to justify this speedup). However, in that paper, the authors did not introduce any enhancements in the mode decision process. In [11], the authors had more focus on mode decision. They proposed a fast mode decision mechanism by considering that the mode decision error surface is monotonic with partition block size. Their approach depends on searching specific block sizes first, then based on the values of the resulting cost parameters, a decision was made whether to perform an early termination or not. If not, all possible search modes are performed. This means that for highly dynamic sequences, the speedup of the mode decision process is expected to be negligible. The authors did not provide any computational requirements analysis or comparisons, and they reported a speedup in the encoding process of the examined CIF sequences, without describing the configurations of the reference (JM) they are comparing with. In [12], the authors proposed fast mode decision and motion estimation processes with main focus on MPEG–2/AVC transcoding. The algorithm utilizes the motion information from MPEG–2 for an AVC encoding using EPZS. The target application for this work had limited processing power; hence the authors ignored the small sub-partitions that AVC standard offers. This had its effect on the obtained results for the tested CIF sequences. Highly dynamic sequences showed more than 0.3 *dB* degradation in PSNR, with a speedup of an order of magnitude in the encoding time.

Lee et al [13] proposed an early skip mode decision as well as a selective intra mode decision. They reported a maximum improvement of 30% in the required encoding time. Ahmad et al [14] proposed a scheme that is based on a 3D recursive search algorithm and takes into consideration the motion vector cost and previous frame information. They also reported a maximum decrease of 30% in the required encoding time. However, they did not provide any quality assessments; neither did they provide any computational requirements analysis of their algorithm. In [15], Kim et al proposed an algorithm that uses the property of All Zero Coefficients Block (AZCB), obtained by quantization and coefficient thresholding, to skip unnecessary modes. The authors reported an average speedup of two times for the tested QCIF and CIF sequences. However, based on the algorithm flow that they provided, the algorithm is expected to lose its computational efficiency when applied to highly dynamic sequences, as no early skipping will be performed in this case, and all the inter-search modes will be required. Jiang et al [16] proposed a low complexity VBS ME algorithm for video telephony communication. Their technique included classifying macroblocks in a frame into types, and applying different searching strategies for each type. They based their cost calculations on SAD only, ignoring the rate component. They did not propose any modification to the MD module. Hence, the algorithm efficiency is limited to the usage of the proposed ME algorithm only. In spite of the achieved time savings compared to the full search approach, the reported results showed noticeable degradation in encoding time when compared to other fast ME algorithms. Also, the authors' experiments on QCIF and CIF sequences showed that the

algorithm is confined in applications with sequences of low motion and high temporal correlation.

Tanizawa et al [17] proposed a fast rate-distortion optimization method targeting low complex mode decision. The proposed method was based on a 2-step hierarchical mode decision. In the first step, a simple R-D cost without tentative coding is calculated. One or more coding candidates are selected using the obtained R-D cost. The number of candidates varies with the value of the quantization parameter (QP). In the second step, the conventional RDO method is applied to the candidates that have been chosen in the first step. The authors timing comparisons did not include the motion estimation operation. A disadvantage of this method is its unsuitability for hardware implementation due to the extremely complex RDO calculation that is applied for the successful candidates of the first step. Dai et al proposed an algorithm in [18] that limits the candidate modes to a small subset by pre-encoding a down-sampled small version of the image. They reported a 50% reduction in encoding time for the three tested CIF sequences. Kuo et al [19] also proposed a multi-resolution scheme and an adaptive rate-distortion model with early termination rules to accelerate the search process. In addition, the authors derived a rule to estimate the bits resulting from residual coding. The reported results showed significant improvement in the encoding time for the tested CIF sequences when compared with the exhaustive full search technique. However, the algorithm showed poor R-D performance when applied to highly dynamic sequences. Chen et al [20] proposed a fast bits estimation method for entropy coding to be used in RDO instead of calculating the actual bitrate, which is an extremely complex operation if performed for each candidate location. The algorithm is based on simplifying the CAVLC only, which makes it inapplicable to the Main- and High-profile encoders that are intended to be configured to use CABAC. The authors did not propose any simplification to the motion search and mode decision strategy, which makes a hardware implementation of their method unpractical.

Rhee et al [21] introduced one of the earliest proposals to use VBS ME. In their work, they proposed two algorithms. One of them was extremely computationally extensive and they proposed to use it as a reference by the following research work in the field, while the other was a simpler version that is based on heuristics. Their work relied on local motion information. A set of candidate motion vectors of each fixed size small block is first obtained by full search whose matching error is less than a prescribed threshold. Neighbouring blocks are then merged in a quad-tree manner if they have at least one motion vector in common. The only modes the algorithm supported were the square modes (4×4, 8×8, and 16×6). Tu et al extended this work in [22] by taking the quantization parameter and the Lagrange cost function into consideration to determine the threshold for comparing the distance between the motion vectors of the two blocks. The experimentation was performed on QCIF and CIF sequences. No timing or computational requirements analysis was introduced. The authors proposed to start with 8×8 searches, and then merge all the way up in the tree to obtain larger block sizes. Hence the supported modes were (8×8, 16×8, 8×16, and 16×16) only. The authors then proposed in [23] a merge and split scheme, which gave the algorithm the ability to represent all search modes. They proposed to perform a "simple" refinement search after every merge or split operation. This extra search would introduce relatively large complexity overhead to the motion estimation module if the complexity of the main motion estimation search is comparable. The obtained results for the tested QCIF and CIF sequences were obtained at QP = 30 only and no rate distortion curves were given. In [24], the authors used the same merging scheme, with the possibility of excluding some low-probability modes in the mode decision process. Again, this extra search that is required after each merging or splitting would introduce relatively large complexity overhead to the motion estimation module if the complexity of the main motion estimation search is comparable.

Ates et al [25] proposed a hierarchical block-matching-based motion estimation algorithm that uses a common set of SAD computations for motion estimation of different block sizes. Based on the hierarchal prediction and the median motion vector predictor of AVC, the algorithm defines a limited set of candidate vectors; and the optimal motion vectors are chosen from this common set. The authors showed their complexity analysis; however, they did not support it with experimental timing measurements. This algorithm showed acceptable PSNR degradation and increase in required bitrates for the tested QCIF, CIF, and SIF sequences. The next section describes the proposed VBS selection scheme, accompanied with some theoretical computational requirements analysis to show its effectiveness.

## III. THE PROPOSED VBS MODE DECISION ALGORITHM

The proposed variable block size motion estimation and mode decision algorithm is based on a merging scheme. It is a bottom-up approach approximation method that exploits the correlation of the smaller blocks motion vectors in a uniform or close motion vector field to build up the larger blocks. The algorithm uses the available typical motion estimation information such as predictor, refinement displacement, and cost. The cost here maybe directly considered as the R-D Cost described in (3) assuming that it is going to be pre-calculated in the motion estimation engine and passed as an input to the algorithm, or it might be considered as the SAD requiring the calculation of the R-D Cost within the algorithm module. A description of the main data structure of the algorithm is given in Section A. In Section B, a theoretical computational requirements analysis is introduced to emphasise the computational efficiency of the proposed algorithm. Finally Section C describes the flow of the proposed algorithm.

### A. The Main Data Structure of the Proposed Algorithm

The main data structure of the algorithm is shown in Figure 7. It is mainly a decision tree that is used to decide about the "suitable" neighbouring nodes to approximately merge them upwards forming a parent node. The algorithm is based on the observation that, if the cost of a parent block is higher than the sum of costs of the children blocks, then the even larger block-size modes can be excluded. Each node in the tree represents a "legal" block partition. A node is represented by its best-representing motion vector that have been calculated by block-based motion estimation (or interpolated by merging), the predictor used as the anchor to start searching around, and the R-D cost accompanied with this motion vector and motion vector predictor.

### B. Computational Complexity Analysis

The tree consists of four decision-levels and five nodes-levels as shown in Figure 7. Roughly, it can be assumed that all the *even* nodes-levels of the tree (level 0, level 2, and level 4) require approximately the same computational requirements (*1x* each level) to find the best motion vectors for all the nodes of a level and to calculate their accompanied costs, while the two *odd* nodes-levels of the tree (level 1 and level 3) require approximately a computational requirements of (*2x* each level) to find the best motion vectors and costs.

The overall ME/MD module computational requirements can be viewed as the sum of the VBS best motion vector search computational requirements and the VBS mode decision computational requirements. This is shown in (9).

$$Comp\_Req_{\_Total} = Comp\_Req_{\_MV\_Searches} + Comp\_Req_{\_VBS\_MD} \quad (9)$$

Hence, the overall inter-related computational requirements of the ME/MD module in the JM can be estimated using Figure 5 and Equation (9). Due to the exhaustive nature of the direct JM implementation, it is required to perform all the types of searches for all possible modes, which is equivalent to *7x*, in order to find the best match for each and every block type. Also, it is required to find the best partitioning scheme among all the 259 possible combinations, a relatively complex operation, whose computational requirements will not be described in details due to its dependence on the way it is implemented. Due to the dominance of the search computational requirements over the mode decision one, any reference to (JM 7x) in the remaining part of this paper is a reference to the exhaustive-search JM implementation of the ME/MD module.

By nature, the proposed merging tree makes the ME/MD module simpler than the JM implementation due to its hierarchical mode decision approximation scheme (simpler mode decision operation). However, as mentioned above, the dominant part that determines the overall module computational requirements is the influence it has on the computational requirements of the motion vector search part (first parameter of (9)). Therefore, the most effective cost reduction technique for the overall module would come by reducing the required number of searches. This led to the idea of allowing the algorithm to meet a wider base of market demands by making it adjustable to work at three different operating points, trading between computational requirements and R-D performance. The key concept is to reduce the required searches as much as possible, while still keeping the R-D degradation within the different allowance windows offered by different customers. Figure 8 describes the difference between the three operating points. The encircled levels refer to levels where the motion vectors of all their nodes are found by performing block-based searches. The arrows refer to the interpolated levels and where they are interpolated from. For *operating point 1*, the algorithm requires performing the 4×4, 8×8, and 16×16 types of searches. This corresponds to a search computational requirements of *3x* (based on the metric mentioned above), which is less than half the computational requirements of the JM exhaustive implementation (JM 7x). The motion vectors and the predictors of the other types of modes (4×8, 8×4, 8×16, and 16×8) are "interpolated" using the actually-calculated motion data of the child nodes in the levels that directly precede those levels in the merging tree structure. The algorithm working at this operating point would fit the requirements of high-resolution applications, such as SD and HD broadcasting. *Operating point 2* is the second mode of the algorithm. It requires performing 4×4 and 8×8 types of searches only, reducing the search computational requirements to *2x*, while the motion vectors and the predictors of the other types of modes (4×8, 8×4, 8×16, 16×8, and 16×16) are "interpolated" using the tree structure. *Operating point 3* is also introduced to meet the requirements of applications such as simple handheld devices where reduced computational requirements (and power consumption) is the main priority. This mode requires performing 4×4 searches only (level 0 of the tree), reducing the search computational requirements to *1x*, while the motion vectors and the predictors of all the other types of modes (4×8, 8×4, 8×8, 8×16, 16×8, and 16×16) are "interpolated" in a bottom-up fashion all the way through the tree structure.

Table 3 shows the schemes that are going to be referred to in the next section, with a brief description of each. Note that JM 1x, JM 2x, and JM 3x, are generated by running the JM, disabling some of inter search/decision modes. They are mainly included to evaluate the performance of VBS 1x, VBS 2x, and VBS 3x (operating point 3, 2, 1) respectively compared to their corresponding almost-same-computational-requirements JM implementations (in addition to the main comparison with the optimum R-D performance of JM 7x). A '√' in the above table refers to an actual MV search and mode decision. A '–' refers to an absence of this block-type search and mode decision, while an 'M' refers to a block-type that is interpolated by merging of child nodes. The number of comparisons required for VBS 1x, 2x, and 3x mode decision are derived assuming that each merging rule requires a single comparison. The MD computational requirements of VBS 2x and 3x also include

the additions and comparisons required for early mode skipping, while the computational requirements of the control logic are not stated. The MD computational requirements for JM 2x and 3x are not given in details because of being implementation-dependent.

### C. Description of the Proposed Algorithm

In this section, a description of the algorithm that has been implemented is provided. The coverage of the algorithm to the three operating points mentioned above is also discussed. Figure 9 shows the flowchart of the algorithm. As can be seen, at operating point 3, the tree structure is parsed (checked for merging) starting from its level 0 (L0) all the way up to its level 4 (L4). At operating point 2, the sum of costs (using (3)) of each four neighbouring 4×4 nodes (each L0 quad) is compared with the corresponding 8×8 cost to decide whether to start with 4×4 quads followed by merge-checking for one level up, or to start with the 8×8 level (L2) followed by merge-checking for two levels up. Operating point 1 differs from operating point 2 by having an additional check for possible early decision to choose 16×16 mode, which makes the last-level merge-checking stage unnecessary. In Figure 9, it is assumed that the nodes costs are calculated on the spot; however, they can be passed as inputs to the VBS module as they have already been calculated during the search for best motion vectors. A key element that determines the efficiency of the above merging scheme is how accurate the merging rules are. On the other hand, its computational requirements are very influential in determining the second parameter in (9). Note that based on Figure 7, different rules maybe used for merge-checking of different nodes. However, for simplicity, the same merging rule is initially presumed to be applicable for all pairs of blocks. Other varieties of the algorithm may be applicable, such as changing the rule according to the level it is located in, or giving some nodes more priority than others by subjecting them to more accurate merging rules. It is clear that a suitable rule would be an accurate, yet simple one. The steps of the merge-checking rule are provided in Figure 10.

It is worth mentioning that for generality, the shown pseudo-code is based on the assumption that each searched node has been searched using N_MV different motion vectors predictors (N_MV is assumed to be 4 as an example), or that N_MV successful candidates are elected by the ME search engine rather than only the best one per search. This means that each search node will be initially marked by 4 best motion vectors and 4 motion vector predictors, which would help avoiding falling into local minima. For simplicity, all the testing results mentioned in the next section have been generated after assigning N_MV to 1. Also it was found that using Th_x = Th_y = 0 is a reasonable choice. The next section shows the experimental analysis and results obtained by encoding various benchmark video sequences.

## IV. Experimental Analysis and Results

In this section, results of the performed experiments to evaluate the performance of the proposed algorithm are introduced. The main goal is to compare the algorithm at its three operating points to the different JM references mentioned in the previous section. Section A describes the method and experimentation environment, while Section B shows the detailed comparisons with the brutal-force method that is adopted by the reference software in terms of time complexity and R-D performance.

### A. Method and Experimentation Environment

The evaluation process is done by performing a comprehensive test where various video sequences are encoded by seven versions of the encoder (see Table 3) throughout a specific range of Quantization Parameter (QP). The goal is to plot the seven generated R-D curves for each sequence to sense the closeness of the R-D performance of the encoder working with the proposed VBS ME/MD algorithm, to the optimum exhaustive solution. Besides, the different encoding times of various sequences with the tested versions of the reference software will be given, and compared to the "theoretical" computational requirements analysis that has already been introduced in Section B of the previous section. The sequences were selected to represent a variety of motion types. This helps in creating a wide range of input stimuli to test the algorithm behaviour. In order to cover a wider range of rate and quality requirements during the testing process, QP was set to vary throughout a broad range of values.

### B. Obtained Results

This section starts by showing execution-time measurements and results in subsection 1, followed by the rate-distortion results in subsection 2.

#### 1) Execution Time Measurments and Results

All the results that are discussed in this section have been obtained by running the seven versions of the AVC encoder (defined in Table 2) on a unified platform. Table 4 shows the time spent by each of the seven versions of the software on encoding ten frames of each of the tested sequences (with QP = 30). The values between parentheses represent the savings in computational time with respect to the JM7x version. The table shows that using VBS 1x, VBS 2x, or VBS 3x, reduces the encoding time of all the tested sequences at least by more than half when compared to the required encoding time for the pure JM 7x version. This conforms to the theoretical computational requirements analysis that has been discussed in Section B of the last section. The total encoding time for VBS 1x, VBS 2x, and VBS 3x is almost the same as for JM 1x, JM 2x, and JM 3x respectively (with minor increase due to the extra comparisons and additions). However, the next section shows that the improvement of the VBS *n*x algorithms over

the corresponding JM *n*x ones in terms of R-D performance is clear enough to neglect this minor complexity overhead, especially when targeting low bitrate applications. Also, the table shows that most of the encoding time is being spent on the motion estimation calculation, which conforms to the encoder complexity profile that was shown in Figure 6. Figure 11 shows a graphical representation of the obtained results. The results show that the more motion a sequence contains, the more time it spends in motion estimation, which in turns translates to an overall increase in encoding time.

*2) Rate-Distortion Measurments and Results*

Figure 12 to Figure 14 show the R-D behaviour of the tested versions of the encoder, with an emphasis on the low-bitrate region. The figures show that the generated graphs are consistent with the expected behaviour of the algorithm. It is clear that for all the sequences, the lower the bitrate, the more effective the algorithm appears to act (at all operating points). This is because at lower bitrates, motion data have a comparable effect on bitrate to the residual data; hence any savings are highly sensible.

For all sequences, VBS 1x may be used as an optimized version of JM 1x. Though, its performance is relatively poor when compared to VBS 2x, VBS 3x, or JM 7x, it can be used as a reasonable compromise when the target application requires low complexity and low power system with reasonable R-D behavior. For all the examined sequences, JM 7x does not outperform VBS 3x by more than 0.2 *dB* at any bitrate (around the target bitrate that suits the sequence resolution).

VBS 1x, VBS 2x, and VBS 3x introduce enhancements over JM 1x, JM 2x, and JM 3x respectively. The enhancements become more sensible at low bitrates. For example, VBS 1x introduces huge enhancement (around 8 *dB*) over JM 1x when encoding the sequence "Mobile and Calendar QCIF 30 fps" (Figure 14) targeting as low bitrate as 29.5 Kbps. Also, the merging operation that was performed resulting in VBS 2x boosted the curve of JM 2x for the sequence "Miss America QCIF 30 fps" (Figure 13) by (6 *dB*) at 26.5 Kbps bitrate.

In summary, the experimental analysis and results demonstrate the main contribution of the proposed algorithm. It is mainly the ability to exhibit acceptable R-D behavior for different sequences with various types of motion. Nevertheless, the ME/MD computational requirements are less than half the computational requirements of ME/MD of JM 7x. This leads to faster encoding time on software platforms, as well as smaller (hence less expensive) implementations on hardware platforms.

## V. CONCLUSION

Having the VBS ME tool in the AVC standard improves its coding efficiency significantly. However, it also introduces extreme computational requirements to the encoder. The JM (AVC software reference model) has an exhaustive approach to implement VBS ME/MD. All seven types of motion estimation searches are performed, and then in the mode decision step, an exhaustive search follows to choose the best partitioning scheme among all possible combinations. Knowing that VBS ME and MD typically represent from 70% to 90% of the entire encoder computational requirements, many research efforts have been introduced in the literature to reduce their computational requirements. However, most of the solutions were local to specific types of simplified motion estimation searches.

In this paper a computationally-efficient VBS selection scheme was introduced. The scheme is applicable to any VBS ME module, leading to significant reduction in its computational requirements with minor loss in the quality of the reconstructed picture. Three versions of the proposed algorithm have been introduced in order to meet different applications' demands. Evaluation experiments were performed on three benchmark video sequences with various spatial and temporal characteristics ranging from smooth slow motion, up to random fast motion. Timing analysis of the performed experiments showed that the proposed algorithm (with its three versions) reduces the encoding time of all the tested sequences at least by half when compared to the required encoding time for the pure brutal-force solution. Objective quality measurement is represented by R-D performance. It has shown that, for all the performed tests, VBS 1x, VBS 2x, and VBS 3x introduce enhancements over JM 1x, JM 2x, and JM 3x respectively, with minor computational overhead. In general, the proposed algorithm is mostly effective with low-power decoder devices, with reduced computational resources, especially when targeting low-bitrate video applications.

## ACKNOWLEDGMENT

The authors would like to thank the Advanced Technology Information Processing Systems (ATIPS) Laboratory, Alberta Informatics Circle of Research Excellence (iCORE), the Alberta Ingenuity Fund (AIF), the Natural Sciences and Engineering Research Council of Canada (NSERC), CMC Microsystems, Micronet R&D, Canada Foundation for Innovation (CFI), the Department of Electrical and Computer Engineering at the University of Calgary, Xilinx Inc., and the German University in Cairo (GUC) for supporting this research.

## REFERENCES

[1] I. Amer, R. Hamed, W. Badawy, G. Jullien, and J. W. Haslett, "An Enhanced Lenient Merging Scheme for H.264 Variable Block Size Selection", proceedings of *International Conference on Advances in Multimedia (MMEDIA)*, Colmar, France, pp. 136-139, July 2009.

[2] S. Lin, C. Chang, C. Su, Y. Lin, C. Pan, and H. Chen, "Fast Multi-Frame Motion Estimation and Mode Decision for H.264 Encoders," Proceedings of *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, Vol. 2, pp. 1237-1242, June 2005.

[3] G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Magazine*, Vol. 15, pp. 74 - 90, November 1998.

[4] W. Chung, "Implementing the H.264/AVC Video Coding Standard on FPGAs," *A white paper*. [Online]. Available:

[5] www.xilinx.com/publications/solguides/be_01/xc_pdf/p18-21_be1-dsp4.pdf

[6] Y.-W Huang, B.-Y Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G Chen, "Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 4, pp. 507-522, April 2006.

[7] D. Wu, S. Wu, K. Lim, F. Pan, Z. Li, X. Lin, "Block INTER Mode Decision for Fast Encoding of H.264," Proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 181-184, May 2004.

[8] A. Yu and G. Martin, "Advanced Block Size Selection Algorithm For Inter Frame Coding in H.264/MPEG–4 AVC," Proceedings of *IEEE International Conference on Image Processing*, Vol. 1, pp. 95-98, October 2004.

[9] A. Yu, "Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG–4 AVC," Proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 3, pp. 169-172, May 2004.

[10] H. Tourapis and A. Tourapis "Fast Motion Estimation within the H.264 CODEC," Proceedings of *IEEE International Conference on Multimedia and Expo*, Vol. 3, pp. 517-520, July 2003.

[11] P. Yin, H. Tourapis, A. Tourapis, and J. Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," Proceedings of *IEEE International Conference on Image Processing*, Vol. 3, pp. 853-856, September 2003.

[12] X. Lu, A. Tourapis, P. Yin, and J. Boyce, "Fast Mode Decision and Motion Estimation for H.264 with a Focus on MPEG–2/H.264 Transcoding," Proceedings of *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 1246-1249, May 2005.

[13] J. Lee and B. Jeon, "Fast Mode Decision for H.264," Proceedings of *IEEE International Conference on Multimedia and Expo*, Vol. 2, pp. 1131-1134, June 2004.

[14] A. Ahmad, N. Khan, S. Masud, and M.A. Maud, "Efficient Block Size Slection in H.264 Video Coding Standard," *IEE Electronics Letters*, Vol. 40, No. 1, pp. 19-21, January 2004.

[15] Y.-H. Kim, J.-W. Yoo, S.-W. Lee, J. Shin, J. Paik, and H.-K. Jung, "Adaptive Mode Decision for H.264 Encoder," *IEE Electronics Letters*, Vol. 40, No. 19, pp. 1172-1173, September 2004.

[16] Y. Jiang, S. Li, and S. Goto, "A Low Complexity Variable Block Size Motion Estimation Algorithm for Video Telephony Communication," Proceedings of *IEEE International Midwest Symposium on Circuits and Systems*, Vol. 2, pp. 465-468, July 2004.

[17] A. Tanizawa, S. Koto, T. Chujoh, and Y. Kikuchi, "A Study on Fast Rate-Distortion Optimized Coding Mode Decision for H.264," Proceedings o*f IEEE International Conference on Image Processing*, Vol. 2, pp. 793-796, October 2004.

[18] Q. Dai, D. Zhu, and R. Ding, "Fast Mode Decision for Inter Prediction in H.264," Proceedings of *IEEE International Conference on Image Processing*, Vol. 1, pp. 119-122, October 2004.

[19] C.-H. Kuo, M. Shen, and C.-C. Kuo. "Fast Inter-Prediction Mode Decision and Motion Search for H.264," Proceedings of *IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 663-666, June 2004.

[20] Q. Chen and Y. He, "A Fast Bits Estimation Method for Rate-Distortion Optimization in H.264/AVC," Proceedings of *Picture Coding Symposium*, December 2004.

[21] I. Rhee, G.R., Martin; S. Muthukrishnan, and R.A. Packwood, "Quadtree-Structured Variable-Size Block-Matching Motion Estimation with Minimal Error," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, No. 1, pp. 42-50, February 2000.

[22] Y.-K. Tu, J.-F. Yang, Y.-N. Shen, and M.-T. Sun, "Fast Variable-Size Block Motion Estimation Using Merging Procedure with an Adaptive Threshold," Proceedings of *IEEE International Conference on Multimedia and Exp*o, Vol. 2, pp. 789-792, July 2003.

[23] Z. Zhou, M.-T. Sun, and Y.-F. Hsu, "Fast Variable Block-Size Motion Estimation Algorithms Based on Merge and Split Procedures for H.264/MPEG–4 AVC," Proceedings of *IEEE International Symposium on Circuits and Systems*, Vol. 3, pp. 725-728, May 2004.

[24] Z. Zhou and M.-T. Sun, "Fast Macroblock Inter Mode Decision and Motion Estimation for H.264/MPEG–4 AVC," Proceedings of *International Conference on Image Processing*, Vol. 2, pp. 789-792, October 2004.

[25] H.F. Ates and Y. Altunbasak, "SAD Reuse in Hierarchical Motion Estimation for the H.264 Encoder," Proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 905-908, March 2005.

[26] Y.-W. Huang, T.-C. Wang, B.-Y. Hsieh, and L.-G. Chen, "Hardware Architecture Design for Variable Block Size Motion Estimation in MPEG–4 AVC/JVT/ITU-T H.264," Proceedings of *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 796-799, May 2003.

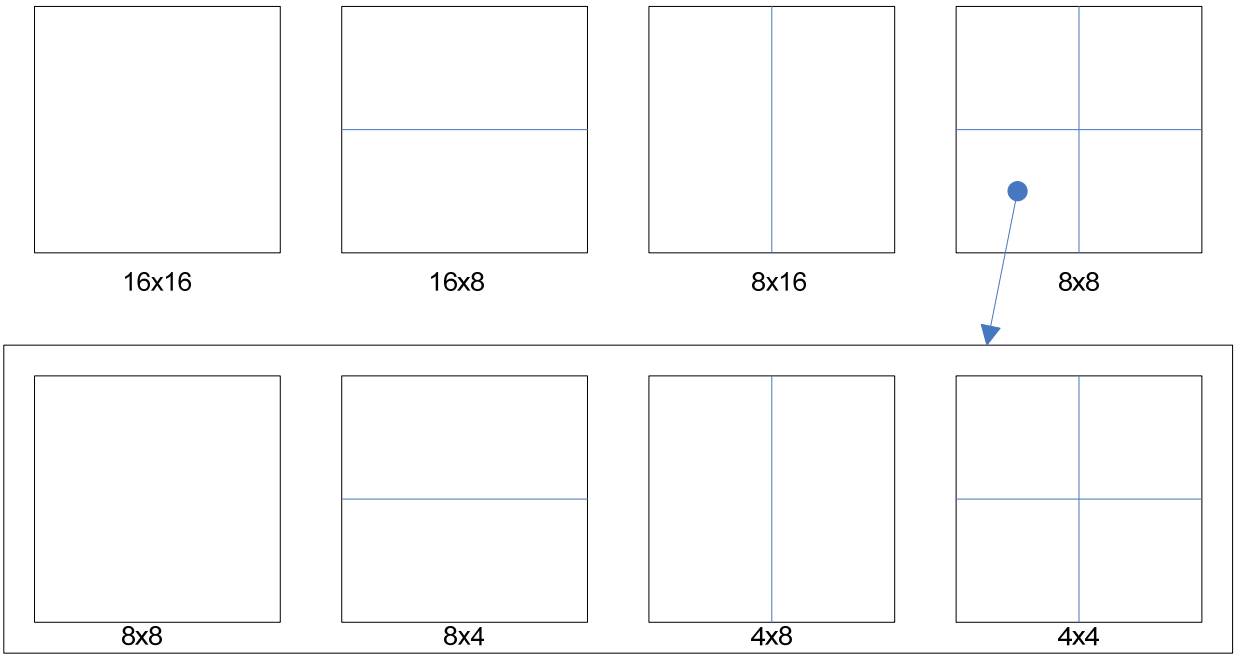



Figure 1. Variable block sizes defined in AVC

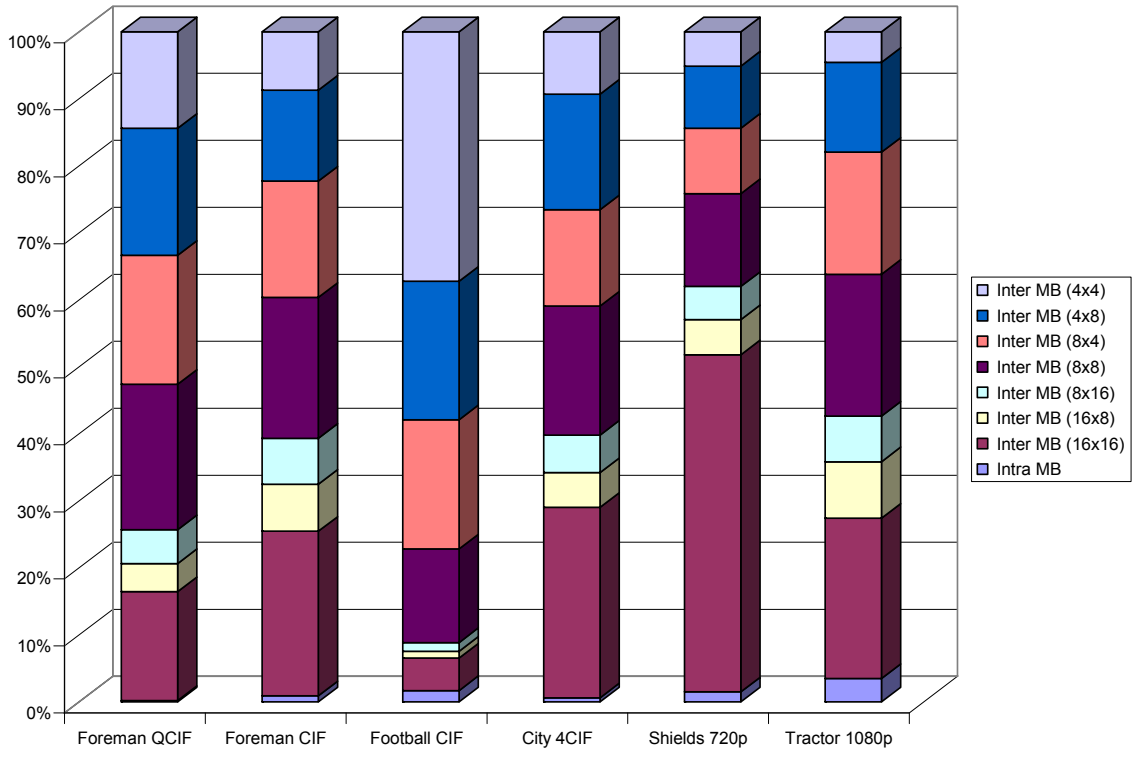



Figure 2. Optimal distribution of the various block sizes in the inter-predicted frames

Table 1. Summary of the characteristics of the tested sequences

| Sequence | Type | Resolution | Frame Rate | Description of the first ten frames | Type of motion |
|---|---|---|---|---|---|
| Foreman | QCIF | 176×144 | 30 fps | A man talking to a still camera | Slow limited motion |
| Foreman | CIF | 352×288 | 30 fps | A man talking to a still camera | Slow limited motion |
| Football | CIF | 352×288 | 30 fps | A part of a football game | Extensive motion |
| City | SD (4CIF) | 704×576 | 30 fps | A scene of a city taken with a panning camera | Regular motion |
| Shields | HD (720p) | 1280×720 | 60 fps | A person pointing at a group of shields while the camera is panning | Camera shooting of highly textured scenes |
| Tractor | HD (1080p) | 1920×1080 | 60 fps | A Tractor working at field | Camera shooting of very high resolution |

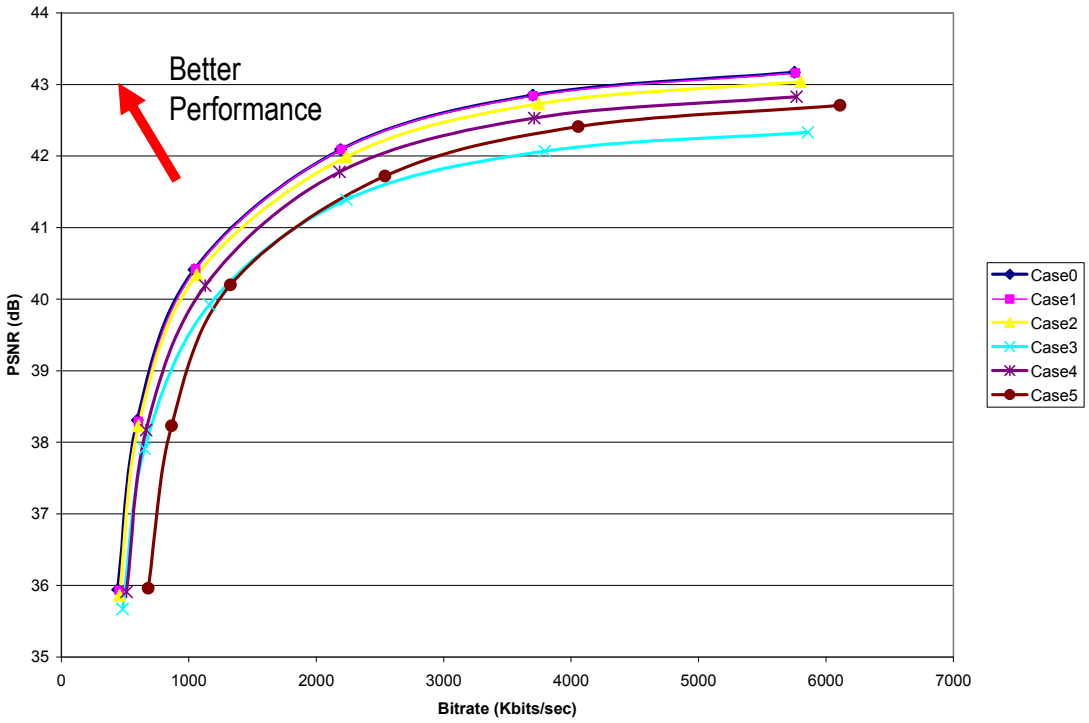



Figure 3. R-D behaviour of Foreman CIF (30 fps) with various inter searches/decision-modes enabled

Table 2. Description of the different curves in Figure 3

| Curve | Description |
|---|---|
| Case 0 | 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4 search/decision modes are enabled |
| Case 1 | 16×16, 8×8, 8×4, 4×8, and 4×4 search/decision modes are enabled |
| Case 2 | 16×16, 8×8, and 4×4 search/decision modes are enabled |
| Case 3 | Only 16×16 search/decision mode is enabled |
| Case 4 | Only 8×8 search/decision mode is enabled |
| Case 5 | Only 4×4 search/decision mode is enabled |

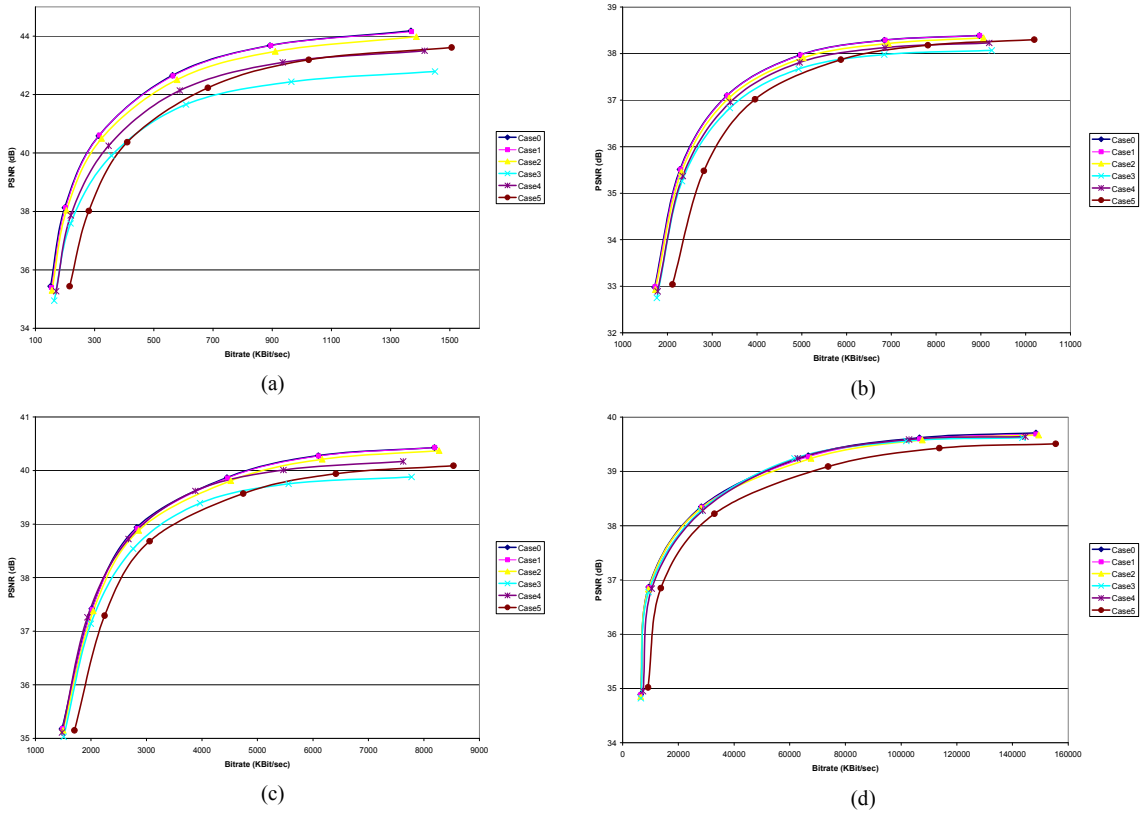



Figure 4. R-D behaviour of different sequences with different inter searches/decision-modes enabled
(a) Foreman QCIF (30 fps)
(b) Mobile CIF (30 fps)
(c) Football CIF (30 fps)
(d) Shields HD 720p (60 fps)

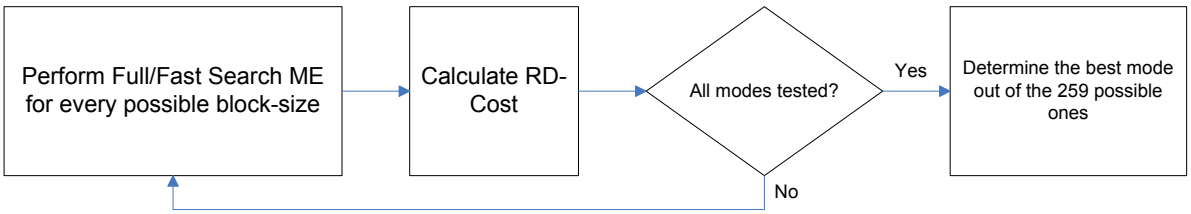
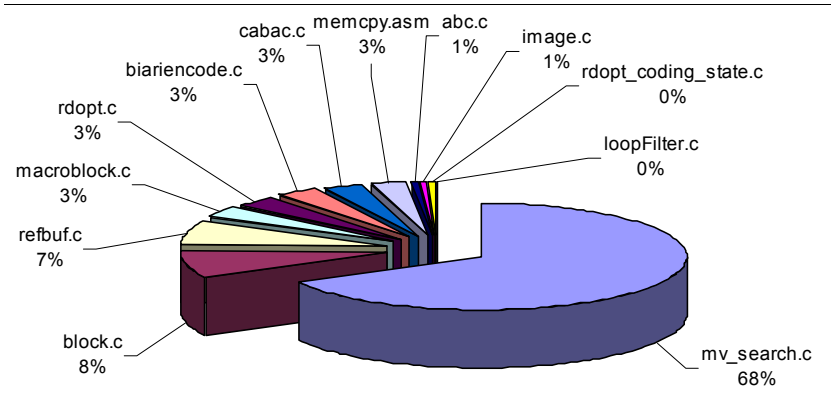
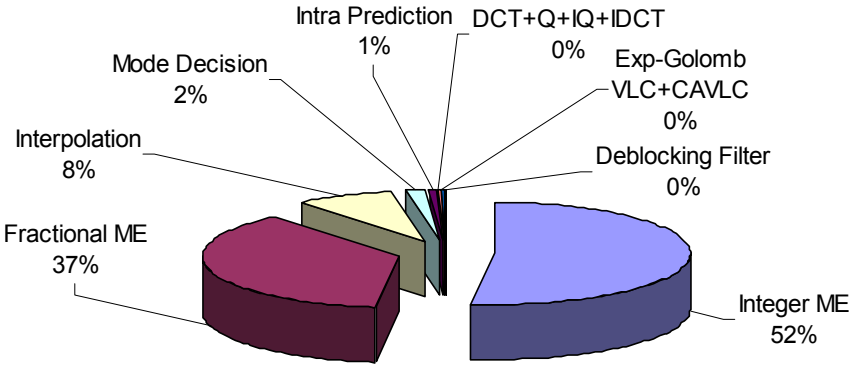



Figure 5. Exhaustive search for best partition scheme as adopted in AVC reference software




(a)




(b)

Figure 6. AVC encoder computational complexity
(a) profiled by files (from [4])
(b) profiled by functional modules (from [6])

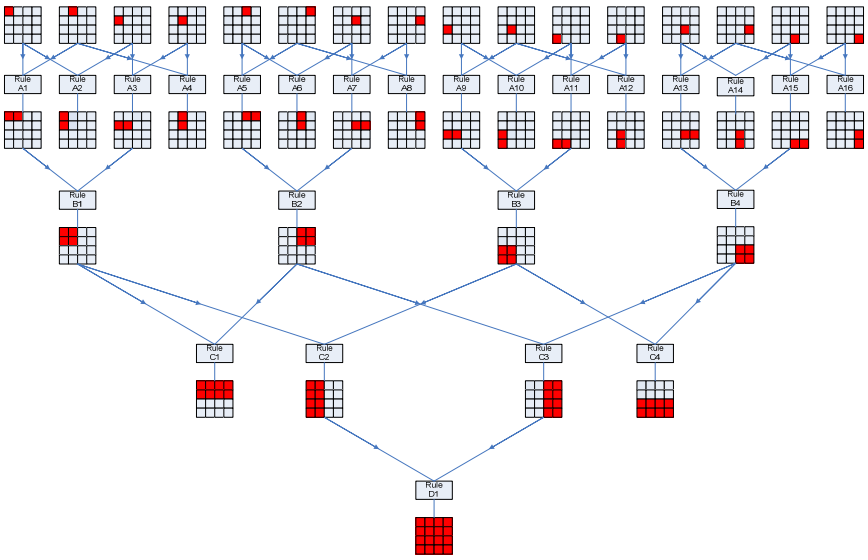


Figure 7. The main data structure of the algorithm: The merging tree

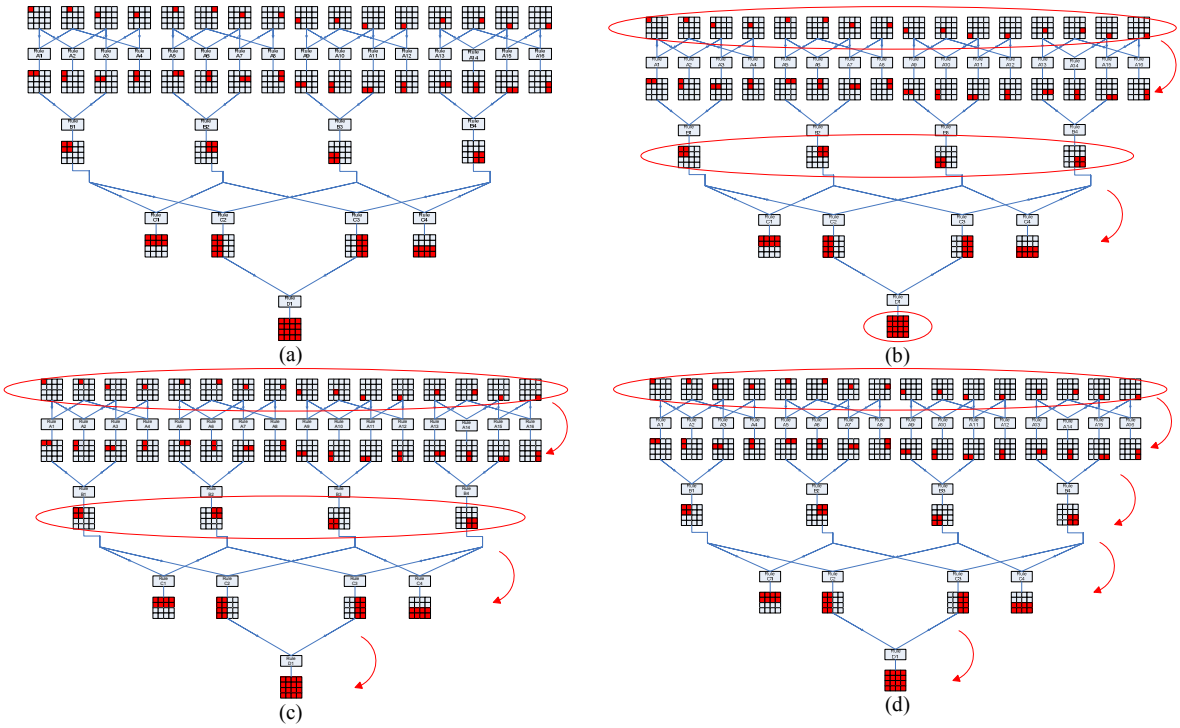



Figure 8. Required searches and interpolations for the algorithm's three operating points

(a) The merging tree (b) Operating point 1 (VBS 3x)
(c) Operating point 2 (VBS 2x) (d) Operating point 3 (VBS 1x)

Table 3. Required searches and computational complexities for the tested references

| Ref. | Search/Decision Modes | | | | | | | Computational Requirements |
|---|---|---|---|---|---|---|---|---|
| | 4×4 | 8×4 | 4×8 | 8×8 | 16×8 | 8×16 | 16×16 | |
| **JM 7x** | √ | √ | √ | √ | √ | √ | √ | 7x MV search + searching 259 combs for MD |
| **JM 3x** | √ | – | – | √ | – | – | √ | 3x MV search + searching 17 combs for MD |
| **JM 2x** | √ | – | – | √ | – | – | – | 2x MV search + searching 16 combs for MD |
| **JM 1x** | √ | – | – | – | – | – | – | 1x MV search + No mode decision |
| **VBS 3x** | √ | M | M | √ | M | M | √ | 3x MV search + 26 comps + 18 adds (for MD) |
| **VBS 2x** | √ | M | M | √ | M | M | M | 2x MV search + 25 comps + 12 adds (for MD) |
| **VBS 1x** | √ | M | M | M | M | M | M | 1x MV search + 25 comps (for MD) |

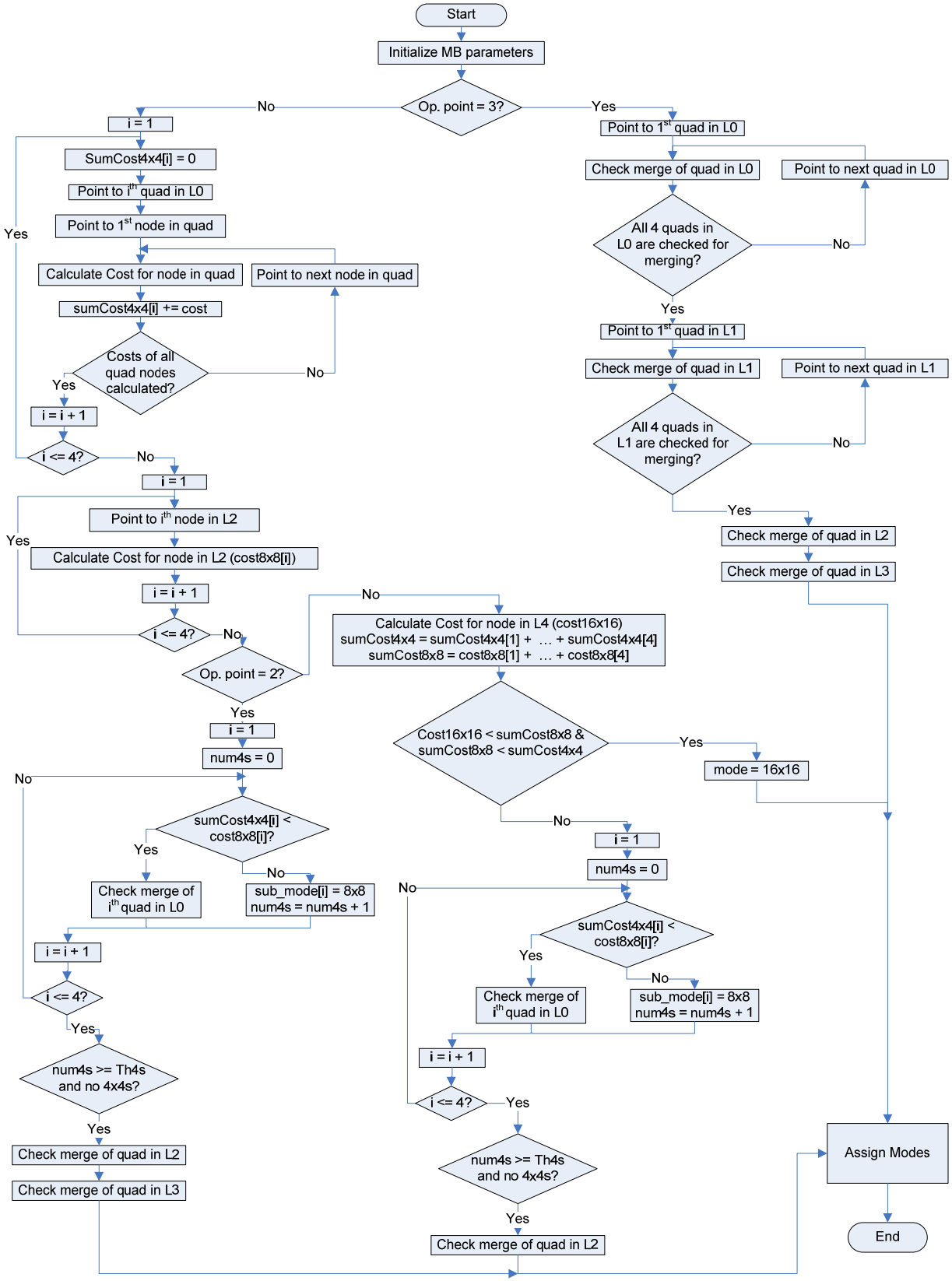



Figure 9. Flowchart of the proposed algorithm

1. If one of the nodes is "unavailable", then the parent node is also "unavailable", else:

2. Calculate the number of identical (or *semi-identical*) MVs in the pool of candidate MVs of each of the two nodes (4×4 = 16 possible pairs of MVs combinations). A simple rule to identify semi-identical MVs would be:

```
D_MV_x = abs(MV_x1 - MV_x2);
D_MV_y = abs(MV_y1 - MV_y2);

n_semi_iden_MV = 0;
If ((D_MV_x <= Th_x) && (D_MV_y  <= Th_y)){
    semi_identical = true;
    n_semi_iden_MV++;
    }
  Else
semi_identical = false;
```

This operation should be repeated to count the number of semi-identical MVs between the two nodes under test (n_semi_iden_MV).

3. Decide whether to merge the two nodes under test or not based on the following rule:

```
If (0<QP<12)                  merge = ((n_semi_iden_MV >12)?true:false);
  Else if (13<QP<25)merge = ((n_semi_iden_MV>9)?true:false);
  Else if (26<QP<38)merge = ((n_semi_iden_MV >6)?true:false);
Else                          merge = ((n_semi_iden_MV _MV>3)?true:false);
```

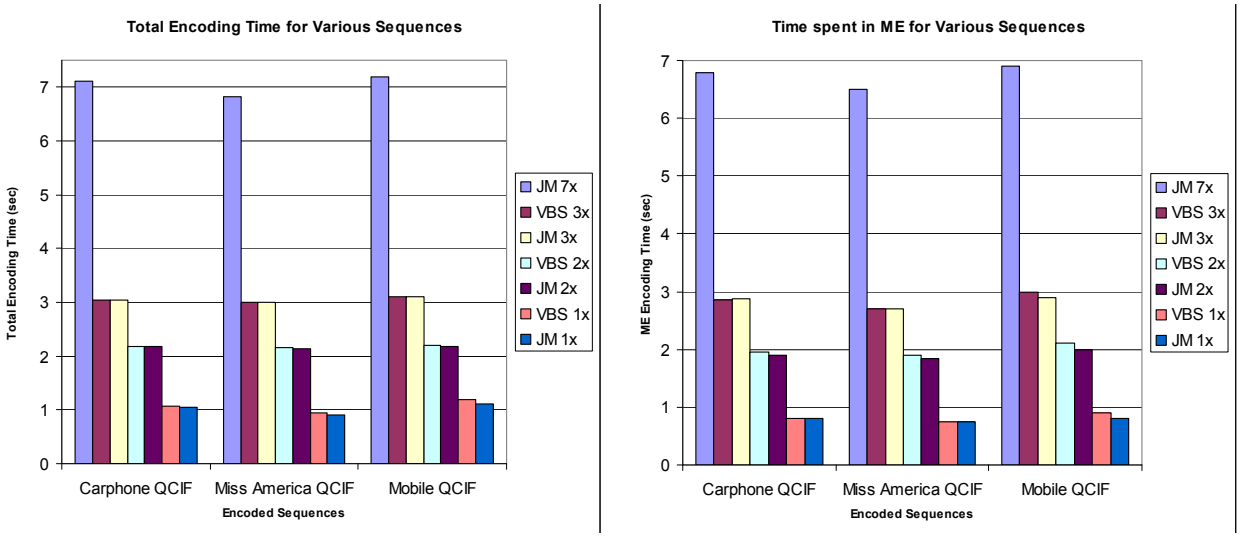
4. If the two nodes are chosen to be merged, then the parent node is marked as "available". The average MVs of the best 4 pairs of semi-identical MVs will be assigned to the parent node, This will be used to decide if this parent node is to be merged with its neighbour next-level node or not.

5. If the two blocks are <u>not</u> to be merged, then each of them will be marked with its best MV out of its 4 candidate ones, and the parent node will be marked as "unavailable".

Figure 10. Steps of the merge-checking rule

Table 4. Encoding time of the tested sequences via the seven versions of the reference software

| | | Carphone QCIF (30 fps) | Miss America QCIF (30 fps) | Mobile QCIF (30 fps) |
|---|---|---|---|---|
| JM 7x | Total Enc. time (sec) | 7.11 | 6.83 | 7.2 |
| | Time spent on ME (sec) | 6.794 | 6.5 | 6.9 |
| VBS 3x | Total Enc. time (sec) | 3.302 (53.56%) | 3.01 (59.93%) | 3.11 (56.81%) |
| | Time spent on ME (sec) | 2.861 (59.76%) | 2.7 (58.46%) | 3.0 (56.52%) |
| JM 3x | Total Enc. time (sec) | 3.049 (57.12%) | 2.99 (56.22%) | 3.1 (56.94%) |
| | Time spent on ME (sec) | 2.877 (57.65%) | 2.71 (58.31%) | 2.9 (57.97%) |
| VBS 2x | Total Enc. time (sec) | 2.172 (69.45%) | 2.153 (68.48%) | 2.2 (69.44%) |
| | Time spent on ME (sec) | 1.952 (72.55%) | 1.9 (70.77%) | 2.1 (69.57%) |
| JM 2x | Total Enc. time (sec) | 2.172 (69.45%) | 2.14 (68.67%) | 2.18 (69.72%) |
| | Time spent on ME (sec) | 1.892 (72.15%) | 1.85 (71.54%) | 2.0 (71.01%) |
| VBS 1x | Total Enc. time (sec) | 1.061 (85.08%) | 0.95 (86.09%) | 1.2 (83.33%) |
| | Time spent on ME (sec) | 0.813 (88.33%) | 0.75 (88.46%) | 0.9 (86.96%) |
| JM 1x | Total Enc. time (sec) | 1.046 (85.29%) | 0.9 (86.83%) | 1.1 (84.72%) |
| | Time spent on ME (sec) | 0.797 (88.27%) | 0.74 (88.62%) | 0.8 (84.06%) |




(a) (b)

Figure 11. Time spent by the seven versions of the software on encoding the tested sequence
(a) Total encoding time (b) ME encoding time

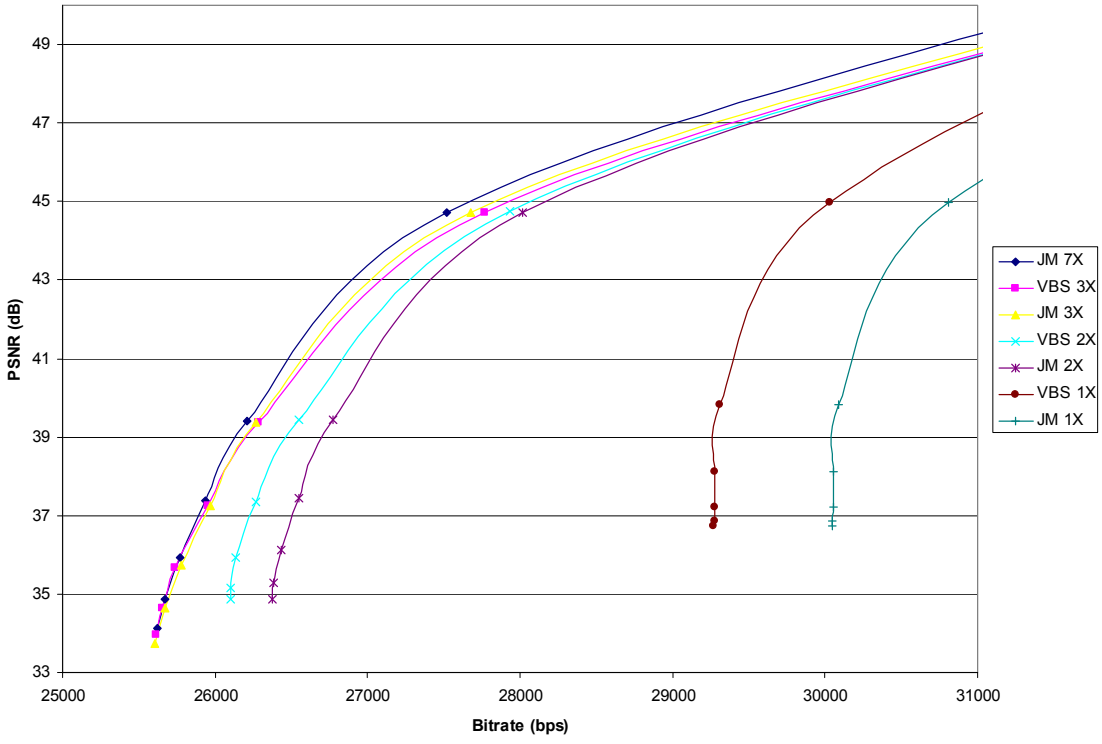



Figure 12. R-D behaviour for Carphone QCIF (30 fps) at low bitrate

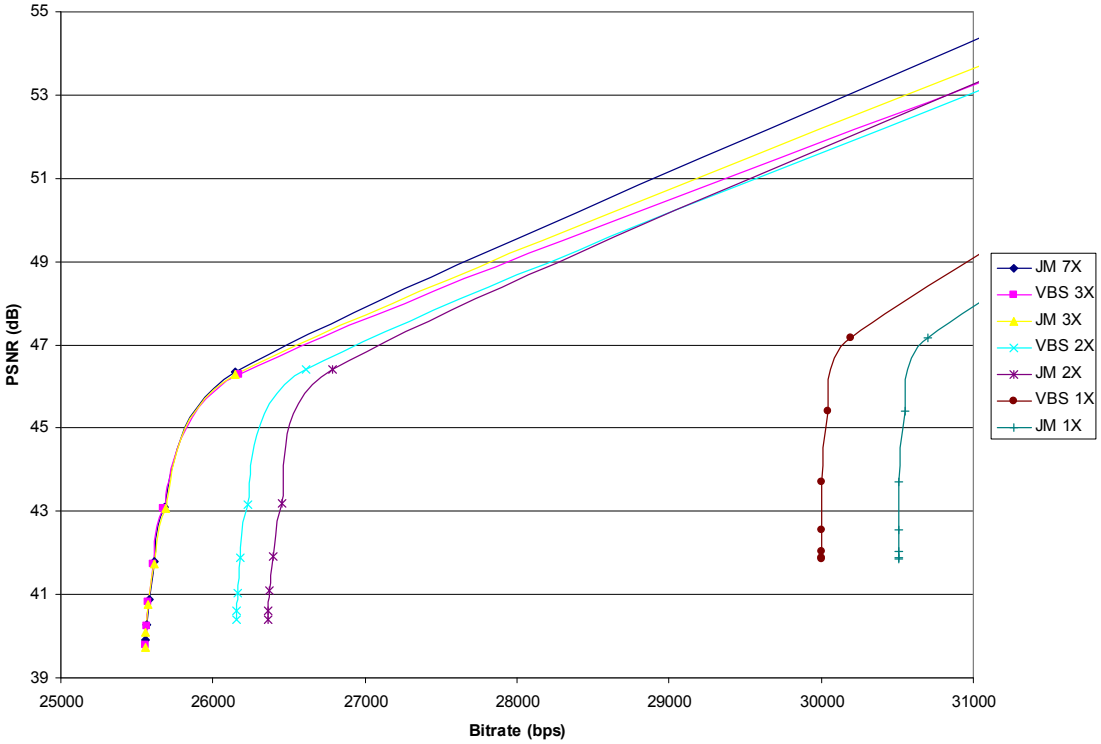



Figure 13. R-D behaviour for Miss America QCIF (30 fps) at low bitrate

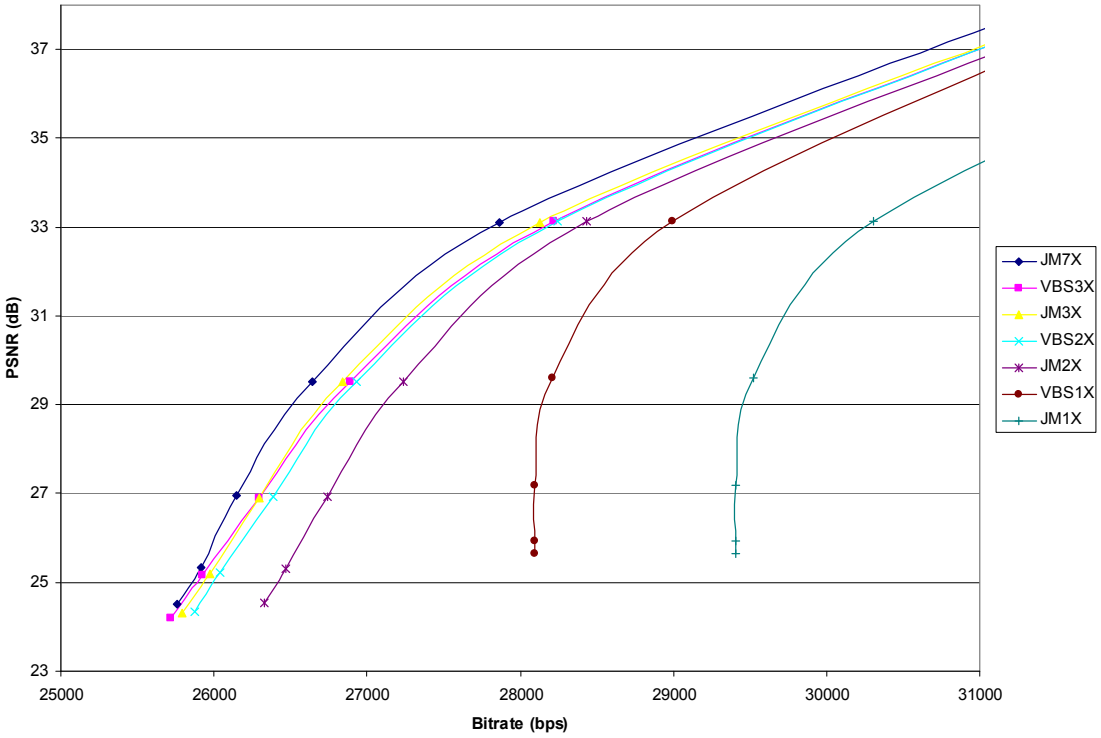



Figure 14. R-D behaviour for Mobile and Calendar QCIF (30 fps) at low bitrate

# Exploiting Concatenation in the Design of Low-Density Parity-Check Codes

Marco Baldi, Giovanni Cancellieri, and Franco Chiaraluce
*DIBET, Polytechnic University of Marche,*
*Ancona, Italy*
*Email: {m.baldi, g.cancellieri, f.chiaraluce}@univpm.it*

***Abstract*—This paper describes a method to exploit the concatenation of very simple component codes in order to obtain good low-density parity-check codes. This allows to design codes having a number of benefits, as high flexibility in length and rate and low encoding complexity. We focus on two kinds of concatenation: the former is classic serial concatenation, in which redundancy is progressively added to the encoded word, whereas the latter is the special case of concatenation coinciding with a bi-dimensional product code. The proposed design technique allows to obtain codes characterized by parity-check matrices with a low density of 1 symbols and free of short cycles in their associated Tanner graph; so efficient algorithms based on the belief propagation principle can be adopted for their decoding. In addition, the systematic form of the component codes can ensure rate compatibility; so the proposed codes can be adopted in type-II hybrid automatic repeat request schemes. We analyze their properties through theoretical arguments and provide some design examples to assess their performance.**

***Keywords*-LDPC codes; concatenated codes; product codes.**

## I. INTRODUCTION

This paper deals with the design of a family of structured low-density parity-check (LDPC) codes based on serial concatenation [1]. The introduction of concatenation in the design of efficient schemes for forward error correction (FEC) is due to Dave Forney in 1966 [2]. Since then, several forms of concatenation have been exploited in the design of codes, providing better and better performance, until the introduction of turbo codes [3], based on concatenated convolutional codes and Bahl, Cocke, Jelinek and Raviv (BCJR) decoders [4].

In recent years, LDPC codes [5] have become the state of the art in FEC techniques, due to their capacity-approaching performance under belief propagation decoding [6]. Since their recent rediscovery [7], many design techniques for LDPC codes have been proposed, that can be classified as structured and non-structured approaches. Structured approaches permit to design codes characterized by rather low implementation complexity that, however, must obey a number of constraints in terms of length and rate, as in the case of quasi-cyclic (QC) LDPC codes [8]. Non-structured designs, instead, are able to produce good LDPC codes with very fine length and rate granularity, as occurs by adopting the progressive edge growth technique [9]. However, non-structured techniques generally produce codes that are less prone to hardware implementation, due to the lack of structure in their characteristic matrices.

As a first aim of this paper, we study how to design structured LDPC codes that can be represented as the serial concatenation of very simple components [10], [11]. We adopt, as component codes, a class of polynomial codes having a binomial generator polynomial. This approach permits us to design codes characterized by a very simple intrinsic structure, that allows to adopt low complexity encoder circuits. The proposed codes can also be shortened arbitrarily, thus obtaining fine length and rate granularity.

We show that the proposed technique can be used to design sets of rate compatible codes [12]. Rate compatibility is important, for example, in the implementation of Type-II Hybrid Automatic Repeat-reQuest (T-II HARQ) schemes, where packets are initially encoded with a high rate code, and then redundancy is transmitted incrementally until successful decoding is achieved. T-II HARQ schemes are particularly useful in packet switched communication networks, since they allow the achievement of capacity-approaching unequal error correction. Serial concatenation is a consolidated procedure to design rate compatible codes [13], [14].

When the code length increases, the advantages of adopting serial concatenation can be limited by the need of rather large component codes. For this reason, we consider a further form of concatenation, coinciding with the structure of a bi-dimensional product code. At the cost of some loss in performance, the adoption of a product structure allows to keep small the size of the component codes while designing concatenated codes with large blocks. We will show that both the serial concatenation and the product structure are able to ensure the LDPC nature of the codes when adopting the special class of component codes we consider. We will also develop some examples aimed at assessing the effect of different design choices in the combination of serial concatenation with the product structure.

The paper is organized as follows. Section II introduces the code design approach based on the serial concatenation of codes with binomial generator polynomial. In Section III the characteristics of the proposed codes are studied. Section IV reports some design examples of serially concatenated codes and their simulated performance, whereas Section V gives some examples of usage of such codes

in bi-dimensional product structures. Finally, Section VI concludes the paper.

## II. Code Design

### A. Component codes

In the considered scheme, the $i$-th component code ($i = 1, \ldots, M$) is a polynomial code with generator polynomial

$$g^{(i)}(x) = (1 + x^{r_i}), \quad (1)$$

where $r_i$, that is a suitably chosen positive integer, represents the code redundancy. We denote by $n_i$ the length of the $i$-th component code and by $k_i = n_i - r_i$ its dimension. Each component code can be seen as a shortened version of a binary cyclic code with length $N_i = \left\lceil \frac{n_i}{r_i} \right\rceil \cdot r_i \geq n_i$, where function $\lceil \cdot \rceil$ returns the smallest integer greater than or equal to its argument. It can be easily verified that:

$$\left(1 + x^{N_i}\right) = (1 + x^{r_i}) \left(1 + x^{r_i} + x^{2r_i} + \ldots + x^{N_i - r_i}\right); \quad (2)$$

so a valid parity polynomial for the cyclic code is:

$$h^{(i)}(x) = \left(1 + x^{r_i} + x^{2r_i} + \ldots + x^{N_i - r_i}\right). \quad (3)$$

Starting from the coefficients of the parity polynomial, it is easy to obtain a valid parity-check matrix for any binary cyclic code in its standard form [15]. For the considered cyclic codes, the parity-check matrix ($\mathbf{H}_i$) has a very regular structure, that is a single row of $\left\lceil \frac{n_i}{r_i} \right\rceil$ identity blocks with size $r_i \times r_i$. It follows that $\mathbf{H}_i$ has size $r_i \times N_i$.

The $i$-th cyclic code has dimension $K_i = N_i - r_i \geq n_i - r_i = k_i$. Each $K_i$-bit information vector can be associated to an information polynomial $m^{(i)}(x)$ as follows:

$$m^{(i)}(x) = m_0 + \ldots + m_{k_i - 1} x^{k_i - 1} + \ldots + m_{K_i - 1} x^{K_i - 1}, \quad (4)$$

where $m_0 \ldots m_{K_i - 1} \in \{0, 1\}$ are the information bits. The codeword corresponding to $m^{(i)}(x)$ can be expressed, in polynomial terms, as follows:

$$\begin{aligned} t^{(i)}(x) &= t_0 + \ldots + t_{n_i - 1} x^{n_i - 1} + \ldots + t_{N_i - 1} x^{N_i - 1} \\ &= m^{(i)}(x) g^{(i)}(x). \end{aligned} \quad (5)$$

We shorten the cyclic code by imposing $m_{k_i} = m_{k_i + 1} = \ldots = m_{K_i - 1} = 0$. This implies $t_{n_i} = t_{n_i + 1} = \ldots = t_{N_i - 1} = 0$, and the parity-check matrix can be accordingly shortened by eliminating its first $N_i - n_i$ columns. Figure 1 shows the structure of the parity-check matrix of the cyclic code and its shortened version. Black diagonals represent 1 symbols, whereas the other symbols are null. The shortened matrix corresponds to the sub-matrix marked in grey.

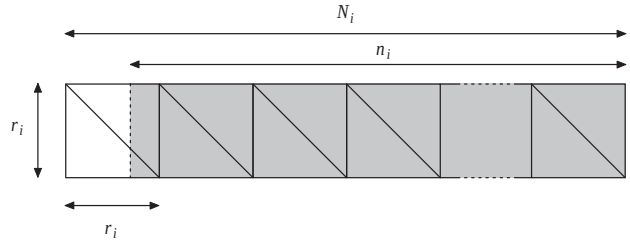



Figure 1. Parity-check matrix of the $i$-th component code.

### B. Serial concatenation

We consider a first family of codes that are obtained as the serial concatenation of $M$ component codes of the type described in Section II-A. We call this family of codes Multiple Serially Concatenated Multiple Parity-Check (M-SC-MPC) codes [16]. Each component code is in systematic form; so, we obtain a systematic serial concatenation, in which redundancy is incrementally appended at the end of the information vector.

The serially concatenated code has dimension $k$ and length $n$. If we set $n_0 = k$, the $i$-th component code has dimension $k_i = n_{i-1}$, redundancy $r_i$ and length $n_i = k_i + r_i$, with $i = 1 \ldots M$. The following relations hold:

$$\begin{aligned} n_1 &= n_0 + r_1 = k + r_1 \\ n_2 &= n_1 + r_2 = k + r_1 + r_2 \\ &\vdots \\ n_M &= n_{M-1} + r_M = k + \textstyle\sum_{i=1}^{M} r_i \end{aligned} \quad (6)$$

and the overall code has length $n = n_M$ and redundancy $r = \sum_{i=1}^{M} r_i$. The parity-check matrix of each component code, in the form of Figure 1, can be used to obtain a valid parity-check matrix for the serially concatenated code. Such matrix ($\mathbf{H}$) is in lower triangular form, and it is shown in Figure 2, for the case $M = 3$. Each column of $\mathbf{H}$ has maximum density $M/r = M/\sum_{i=1}^{M} r_i$ (that is the density of its leftmost $n_1$ columns); the values $r_i$, $i = 1 \ldots M$, must be chosen high enough as to make $\mathbf{H}$ sparse, thus obtaining an LDPC code. Furthermore, we will see in the following that, under suitable conditions, matrix $\mathbf{H}$ corresponds to a Tanner graph free of short cycles, that allows to adopt efficient LDPC decoding algorithms.

It should be noted that the proposed scheme can be seen as a generalization of the multiple serially concatenated single parity-check (M-SC-SPC) approach [17]. The latter, however, assuming $r_1 = r_2 = \ldots = r_M = 1$, does not permit to obtain LDPC codes. Moreover, the performance of M-SC-MPC codes can be better than that of M-SC-SPC codes [16].

A first requirement, when designing serially concatenated codes with the proposed technique, consists in making their parity-check matrix suitable for application of decoding

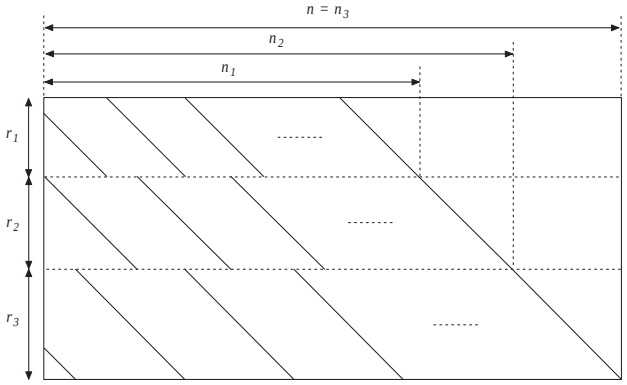



Figure 2. Parity-check matrix of the serially concatenated code.

algorithms based on the Belief Propagation (BP) principle. This can be achieved only if the associated Tanner graph is free of short cycles. For the considered codes, such condition can be easily ensured by following the rules established in Lemma 1 and Corollary 1, that are reported next.

*Lemma 1:* In order to obtain a Tanner graph representation free of length-4 cycles, the M-SC-MPC code must have length $n \leq n_{\max}$, with:

$$n_{\max} = \min_{\substack{i,j \in [1,M] \\ i<j}} \left\{ \mathrm{lcm}(r_i, r_j) + \sum_{l=i+1}^{M} r_l \right\}. \qquad (7)$$

*Proof:* Let us focus on the parity-check matrix for $M = 3$ (see Figure 2), and consider the first two blocks of rows (i.e. the first $r_1 + r_2$ rows). A length-4 cycle exists between any two rows if they have two 1 symbols at the same columns. It can be easily verified that this cannot occur when $n_1 \leq \mathrm{lcm}(r_1, r_2)$, that is $n \leq \mathrm{lcm}(r_1, r_2) + r_2 + r_3$. If we consider the first and third blocks of rows, length-4 cycles are avoided among their rows when $n_1 \leq \mathrm{lcm}(r_1, r_3)$, that is $n \leq \mathrm{lcm}(r_1, r_3) + r_2 + r_3$. Finally, in the last two blocks of rows (i.e. the last $r_2 + r_3$ rows), length-4 cycles are absent for $n_2 \leq \mathrm{lcm}(r_2, r_3)$, that is $n \leq \mathrm{lcm}(r_2, r_3) + r_3$. Hence, in order to avoid length-4 cycles in the matrix of Figure 2, it must be $n \leq \min[\mathrm{lcm}(r_1, r_2) + r_2 + r_3, \mathrm{lcm}(r_1, r_3) + r_2 + r_3, \mathrm{lcm}(r_2, r_3) + r_3]$. This confirms the validity of (7) for the case $M = 3$. The same reasoning can be easily extended to the general case of $M$ component codes, thus proving the assertion. ■

*Corollary 1:* For a set of distinct, coprime and increasingly ordered $r_i$'s, $i = 1 \ldots M$, the Tanner graph of the M-SC-MPC code is free of length-4 cycles for code length $n \leq n'_{\max}$, with:

$$n'_{\max} = r_1 r_2 + \sum_{j=2}^{M} r_j. \qquad (8)$$

*Proof:* If we refer again to the case $M = 3$ (see Figure 2), by Lemma 1, length-4 cycles are avoided when $n \leq \mathrm{lcm}(r_1, r_2) + r_2 + r_3 = r_1 r_2 + r_2 + r_3 = r_2(r_1 + 1) + r_3$, $n \leq \mathrm{lcm}(r_1, r_3) + r_2 + r_3 = r_1 r_3 + r_2 + r_3$ and $n \leq \mathrm{lcm}(r_2, r_3) + r_3 = r_2 r_3 + r_3$. But $r_1 r_2 < r_1 r_3$ and $r_1 + 1 < r_3$, so the first condition is the most stringent one. This result can be extended to a generic value of $M$, in the sense that the condition set by the first two blocks of rows is always the most stringent one. So, under the hypotheses of the corollary, Eq. (8) results. ■

It is important to observe that the proposed design technique achieves very fine granularity in the code length. In fact, provided that $n \leq n_{\max}$, each value of $n$ is feasible and able to ensure a Tanner graph representation free of length-4 cycles.

By comparing (7) and (8), we can observe that the choice of $r_i$'s all distinct and coprime yields the highest values for the code length, i.e. the highest flexibility in the choice of $n$. For this reason, in the following we will consider distinct, coprime and increasingly ordered $r_i$'s, in such a way as to apply Eq. (8).

### C. Design of product codes

A particular form of serial concatenation can be realized by constructing a product code, that can be seen as an $N$-dimensional polytope in which each component code works along one dimension.

We focus on the simplest form of product codes, that are bi-dimensional codes. In this case, the overall code results from two component codes working on the two dimensions of a rectangular matrix. An example of such matrix is reported in Figure 3; we denote by $(n_a, k_a, r_a)$ and $(n_b, k_b, r_b)$ the length, dimension and redundancy of the two component codes. The information bits are written in the inner $k_b \times k_a$ matrix by following a fixed order (for example, in row-wise manner from top left to bottom right). When the inner matrix is filled, the first component code acts on its rows, producing a set of $k_b r_a$ checks, that fill the light grey rectangular region marked as "Checks a". Then, the second component code acts on all $n_a$ columns, so producing $k_a r_b$ checks on the information symbols and further $r_a r_b$ checks on checks. So, the encoding process of a product code can be seen as the serially concatenated application of two components codes. A special feature of this particular case of serial concatenation is that inverting the order of application of the two component codes does not yield any change in the encoded word.

A product code permits to increase the minimum distance in a multiplicative way. If the two component codes have minimum distances $d_a$ and $d_b$, respectively, the product code has minimum distance $d = d_a \cdot d_b$. Several types of component codes have been used in the design of product codes. SPC codes are often used because of their simplicity, but they can yield severe constraints on the overall code length and rate. Better results can be obtained with product codes based on Hamming codes, that can achieve very good

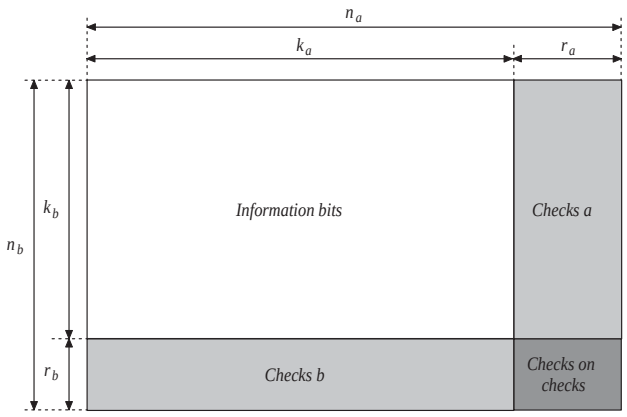



Figure 3. Encoding scheme of a bi-dimensional product code.

performance under soft-input/soft-output iterative decoding [18]. Furthermore, it has been demonstrated that product codes are potentially able to achieve error-free coding with a nonzero code rate (as the number of dimensions increases to infinity) [19], [20].

We are interested in designing bi-dimensional product codes that exploit, as component codes, two serially concatenated codes having the form described in Section II-B. In fact, when large block codes are needed, the form of serial concatenation in Section II-B can require rather large component codes, that can become unpractical for hardware implementation. On the other hand, bi-dimensional product codes could be designed that allow to obtain large blocks while still exploiting component codes with a rather small size. Next, we will show that the parity-check matrix of the bi-dimensional product code can be easily obtained starting from the parity-check matrices of the two component codes, and that the product code is still an LDPC code [1].

Let us suppose that the two component codes have the following parity-check matrices, where $\mathbf{h}_{i,j}$ represents the $j$-th column of matrix $\mathbf{H}_i$:

$$\begin{aligned} \mathbf{H}_a &= \left[ \begin{array}{cccc} \mathbf{h}_{a,1} & \mathbf{h}_{a,2} & \cdots & \mathbf{h}_{a,n_a} \end{array} \right], \\ \mathbf{H}_b &= \left[ \begin{array}{cccc} \mathbf{h}_{b,1} & \mathbf{h}_{b,2} & \cdots & \mathbf{h}_{b,n_b} \end{array} \right]. \end{aligned} \tag{9}$$

It follows that $\mathbf{H}_a$ has size $r_a \times n_a$, while $\mathbf{H}_b$ has size $r_b \times n_b$.

A valid parity-check matrix for the product code having such components can be expressed in the following form:

$$\mathbf{H}_p = \left[ \begin{array}{c} \mathbf{H}_{p1} \\ \mathbf{H}_{p2} \end{array} \right], \tag{10}$$

where $\mathbf{H}_{p1}$ has size $r_a n_b \times n_a n_b$, and $\mathbf{H}_{p2}$ has size $r_b n_a \times n_a n_b$. $\mathbf{H}_{p1}$ can be obtained as a block-diagonal matrix formed by $n_b$ repetitions of $\mathbf{H}_a$:

$$\mathbf{H}_{p1} = \left[ \begin{array}{cccc} \mathbf{H}_a & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_a & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}_a \end{array} \right], \tag{11}$$

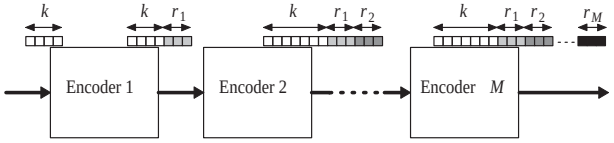



Figure 4. Scheme of the concatenated systematic encoder.

where $\mathbf{0}$ represents an $r_a \times n_a$ null matrix.

$\mathbf{H}_{p2}$ is given by (12): it consists of a row of blocks, in which the $i$-th block contains, along the main diagonal, $n_a$ copies of $\mathbf{h}_{b,i}$ (that is, the columns of $\mathbf{H}_b$), while all the remaining symbols are null. $\mathbf{H}_p$ is redundant, since it includes two sets of parity-check constraints representing checks on checks through both the component codes. For this reason, $\mathbf{H}_p$ cannot have full rank. When both components are in systematic form, a full rank parity-check matrix for the product code can be obtained by eliminating the last $r_a \cdot r_b$ rows from $\mathbf{H}_{p1}$ or $\mathbf{H}_{p2}$, in such a way to avoid doubled representation of checks on checks.

If we suppose that the densities of 1 symbols in $\mathbf{H}_a$ and $\mathbf{H}_b$ are $\delta_a$ and $\delta_b$, respectively, it is easy to prove that the density of $\mathbf{H}_{p1}$ is $\delta_a / n_b$, while that of $\mathbf{H}_{p2}$ is $\delta_b / n_a$. So, even starting from two component codes that are not characterized by very sparse parity-check matrices, the resulting product code can still be an LDPC code. Alternative representations of the parity-check matrix can be found, that can achieve even lower density [21]. For our purposes, however, the density of the parity-check matrix in the form (10), with $\mathbf{H}_{p1}$ and $\mathbf{H}_{p2}$ as expressed by (11) and (12), is low enough.

Furthermore, it is easy to verify that matrix (10) is free of length-4 cycles, provided that the same holds for the component matrices $\mathbf{H}_a$ and $\mathbf{H}_b$. So, the codes obtained as bi-dimensional product codes can be effectively decoded by means of LDPC decoding algorithms. We will give some examples in this sense in the next section.

## III. Code Characteristics

### A. Encoding and Decoding

The serially concatenated codes we consider, described in Section II-B, can be encoded by using a very simple concatenated encoder structure, like that shown in Figure 4. Each component code, described in Section II-A, is in systematic form; so, the $i$-th component encoder simply appends $r_i$ redundancy bits to the input vector. The overall codeword results in the concatenation of the input vector and the redundancy vectors added by the cascade of encoders.

Alternatively, the serially concatenated code can be encoded by using the standard "back substitution" technique. For the proposed concatenated code, the low-density parity-check matrix is in lower triangular form; so, the standard encoding algorithm has very low complexity due to the fact that it works on a sparse matrix. For generic LDPC

$$\mathbf{H}_{p2} = \begin{bmatrix} \mathbf{h}_{b,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{h}_{b,2} & \mathbf{0} & \cdots & \mathbf{0} & & \mathbf{h}_{b,n_b} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{h}_{b,1} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{h}_{b,2} & \cdots & \mathbf{0} & & \mathbf{0} & \mathbf{h}_{b,n_b} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}_{b,1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}_{b,2} & & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{h}_{b,n_b} \end{bmatrix}. \quad (12)$$

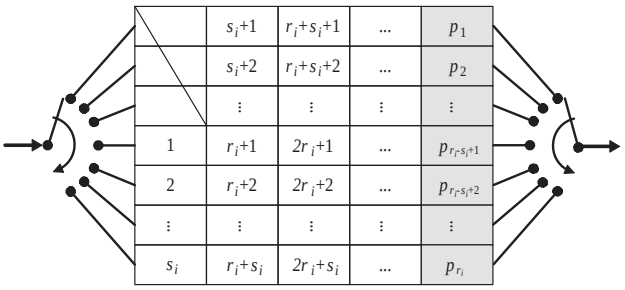



Figure 5. Parallel implementation of the encoder for the $i$-th component code.

codes, instead, a matrix pre-elaboration through Gaussian elimination may be needed in order to put the parity-check matrix in lower triangular form, and this usually does not preserve its sparse character, thus yielding increased complexity.

On the other hand, when adopting the encoder circuit shown in Figure 4, each component code can be encoded by means of a very simple circuit, based on a linear feedback shift register (LFSR) that implements the polynomial multiplication expressed by Eq. (5). Encoding of each component code can be also implemented by using a parallel encoder architecture and serial to parallel and parallel to serial converters [22]. For the proposed component codes, the parallel encoder coincides with a bank of SPC encoders, as shown in Figure 5. The parallel encoder for the $i$-th component code can be represented as a binary matrix with $r_i$ rows and $\left\lceil \frac{k_i}{r_i} \right\rceil + 1$ columns. For encoding, its cells are filled in column-wise order, from top left to bottom right. The first $r_i - s_i$ cells, with $s_i = k_i \mathrm{mod} r_i$, are unused, while the others are filled in the order reported in Figure 5, until the first $\left\lceil \frac{k_i}{r_i} \right\rceil$ columns are completed (white cells in the figure). When the $j$-th row is filled, $j = 1 \ldots r_i$, the parity bit $p_j$ is calculated, by XORing the elements of the row, and its value is stored in the last column, at the same row. When all the parity bits have been calculated, the encoder outputs the codeword by reading the matrix content in the same order used for the input, but including the parity bits.

Due to the LDPC nature of the serially concatenated codes we consider, and to the absence of short cycles in their associated Tanner graph, their decoding can be accomplished through the standard Sum-Product Algorithm with Log-Likelihood Ratios (LLR-SPA) [23], or through its low-complexity versions, like the normalized min-sum (NMS) algorithm [24]. These are BP-based decoding algorithms and, hence, they are able to exploit the length-4 cycle free Tanner graph representation of the code to achieve capacity approaching error-correction performance.

### B. Minimum Distance

An upper bound on the minimum distance of the serially concatenated codes can be obtained by exploiting their concatenated nature and the structure of the component codes.

*Lemma 2:* Serially concatenated codes in the proposed family have minimum distance:

$$d_{\min} \leq 2^M. \quad (13)$$

*Proof:* Let us consider the concatenated encoder shown in Figure 4 with systematic component encoders as shown in Figure 5, and let us focus on the first code, that has redundancy $r_1$. Its minimum weight codewords have Hamming weight 2, and correspond to input vectors having weight 1. Due to the encoder structure, the two 1 symbols in each minimum weight codeword are spaced by an integer multiple of $r_1$, say $a_1 r_1$, with $1 \leq a_1 \leq \left\lfloor \frac{k_1}{r_1} \right\rfloor$, where function $\lfloor \cdot \rfloor$ returns the greatest integer smaller than or equal to its argument.

When such codeword is given as input to the subsequent encoder, the two 1 symbols can be in the same row of the second encoder or not. In the first case, that occurs for $a_1 r_1 = a_2 r_2$, $1 \leq a_2 \leq \left\lfloor \frac{k_2}{r_2} \right\rfloor$, the output codeword has weight 2; otherwise, it has weight 4. The latter case occurs for $k_2 = n_1 < \mathrm{lcm}(r_1, r_2)$, and produces a weight-4 codeword whose 1 symbols can be spaced of integer multiples of $r_1$, $r_2$ and linear combinations of them. When such weight-4 codeword is given as input to the third component encoder, its four 1 symbols can be in four different rows or not. In the first case, the Hamming weight is doubled again, thus reaching 8. The same procedure can be generalized by induction, thus obtaining that, for $M$ component codes, the minimum Hamming weight cannot be greater than $2^M$, that proves the assertion. ■

The proof of Lemma 2 gives an implicit rule for approaching the upper bound on the minimum distance: the number of coincidences among linear combinations of the $r_i$ values, $i = 1 \ldots M$, must be reduced as much as possible in the range $[1, n]$. The choice of coprime $r_i$'s is also favorable from this viewpoint.

When serially concatenated codes are used as components in bi-dimensional product codes, the minimum distance of the overall code can be easily upper bounded starting from Lemma 2 and considering that the product code structure increases the minimum distance in a multiplicative way. So, the following corollary immediately follows.

*Corollary 2:* A bi-dimensional product code having, as components, two serially concatenated codes of the considered family, has minimum distance:

$$d_{\min} \leq 2^{M_a+M_b}, \tag{14}$$

where $M_a$ and $M_b$ represent the number of serially concatenated codes in the two components of the product code.

### C. Rate Compatibility

Rate compatibility consists in designing a set of "compatible" codes with different rates, in such a way as to allow the implementation of schemes with variable error correction capability.

For the serially concatenated scheme we consider, rate compatibility is ensured by systematic encoding, since redundancy is incrementally appended to the information vector. By considering each component code progressively, a set of rate compatible codes is simply obtained, with code rates

$$\frac{k}{k+r_1} > \frac{k}{k+r_1+r_2} > \ldots > \frac{k}{k+\sum_{j=1}^{M} r_j}. \tag{15}$$

An example of rate compatible serially concatenated codes is reported in the following section. Rate compatibility can also be ensured when the serially concatenated codes we consider are used as components in bi-dimensional product codes. In this case, a simple way to obtain a family of rate compatible codes is to fix one of the two components of the product code, whereas the other one can employ a variable number of its sub-components in order to change the overall code rate.

## IV. Examples of Serially Concatenated Codes

In this section, we give some examples of design of serially concatenated codes through the considered technique. We show that such technique is able to produce small, medium and large codes with very fine granularity in the code length and rate, and very good error correction performance. Together with their low complexity encoding, these peculiarities justify possible interest on such codes for practical applications.

All simulations have been performed with Binary Phase Shift Keying (BPSK) modulation over the Additive White Gaussian Noise (AWGN) channel. Coded transmission has been simulated through a suitable software, written in C++ language, that performs a Montecarlo evaluation of bit error rate (BER) and frame error rate (FER) for each value of energy per bit to noise power spectral density ratio ($E_b/N_0$). In order to provide a sufficient level of confidence for Montecarlo simulations, each BER and FER point has been estimated after waiting the occurrence of $100$ erred frames.

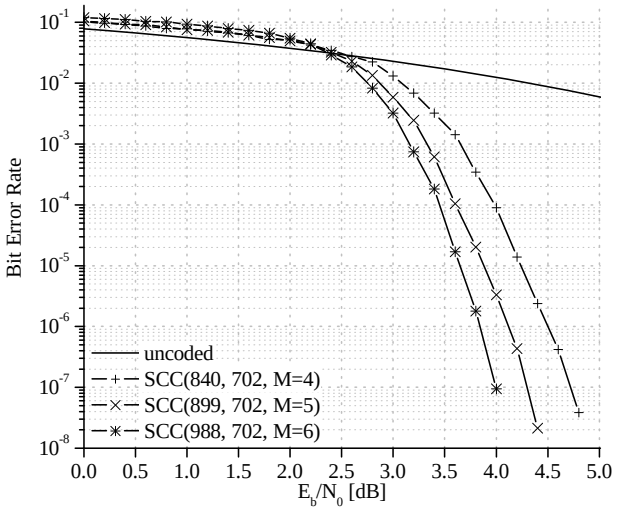



Figure 6. Simulated BER for small codes.

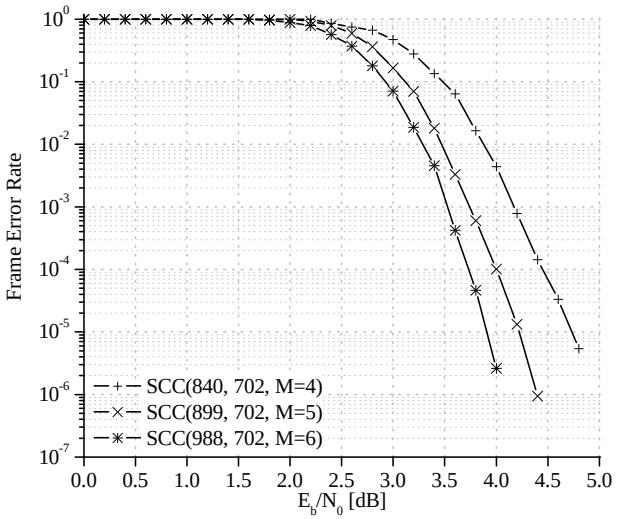



Figure 7. Simulated FER for small codes.

### A. Small Size Codes

We consider a first set of rate compatible serially concatenated codes obtained from the following choice of $r_i$ values: $[29, 31, 35, 43, 59, 89]$. All codes in the family have dimension $k = 702$, but different length and rate, depending on the number of component codes. The first code adopts $M = 4$ components, corresponding to the first four values of $r_i$; thus it has redundancy $r = 138$, length $n = 840$ and rate $R = 0.84$. The second code is obtained by including the fifth component code; so it has redundancy $r = 197$, length

$n = 899$ and rate $R = 0.78$. The last code is obtained by considering all the $M = 6$ values of $r_i$; thus, it has redundancy $r = 286$, length $n = 988$ and rate $R = 0.71$. Their simulated performance in terms of BER and FER, as a function of the signal-to-noise ratio $E_b/N_0$, is reported in Figures 6 and 7, respectively.

Rate compatibility of these codes can be exploited in a T-II HARQ scheme, by transmitting, initially, each packet encoded through the first code; then, if requested, by sending $r_5$ and, finally, $r_6$ bits of further redundancy.

### B. Medium Size Codes

In order to provide a design example of codes with moderate length, we have considered code parameters very similar to those proposed for application in near-Earth space missions by the Consultative Committee for Space Data Systems (CCSDS) [25]. The designed code has length $n = 8208$, dimension $k = 7182$ and, hence, rate $R = 0.875$. Its redundancy ($r = 1026$) corresponds to the following choice of $r_i$ values for the $M = 5$ component codes: $[177, 181, 214, 221, 233]$. Due to the very fine length granularity achievable through the proposed design approach, the code could be arbitrarily shortened in order to have dimension coincident with an integer multiple of 32, as suggested in [25].

The error correction performance of the proposed code, reported in Figure 8, looks very good: its curves are almost overlaid with those of an optimized code with very similar parameters (it has length $n = 8176$ and dimension $k = 7156$) recommended by the CCSDS. The performance of the latter code in terms of BER and FER, shown in Figure 8 (and derived from [25]), refers to an FPGA implementation adopting a maximum number of decoding iterations equal to 50.

It should be observed that the proposed code, besides achieving almost the same performance as the CCSDS code, allows the implementation of very simple encoder circuits, due to its concatenated nature. Therefore, it provides a valid alternative to the CCSDS code that, in turn, is characterized by low encoding complexity thanks to its quasi-cyclic nature.

### C. Large Size Codes

As a further example, we have considered the design of large codes with high rate. We have adopted the following choice of $r_i$ values for the $M = 5$ component codes: $[313, 569, 577, 641, 643]$, that yields redundancy $r = 2743$. With this choice of the parameters, we have designed a first code with length $n = 27430$, i.e. dimension $k = 24687$ and rate $R = 0.9$. Subsequently, this code has been shortened to length $n = 10972$, thus producing a code with dimension $k = 8299$ and rate $R = 0.75$.

In order to assess the effect of a different value of $M$, we have also considered an alternative code design that gives the same value of redundancy ($r = 2743$), but exploiting

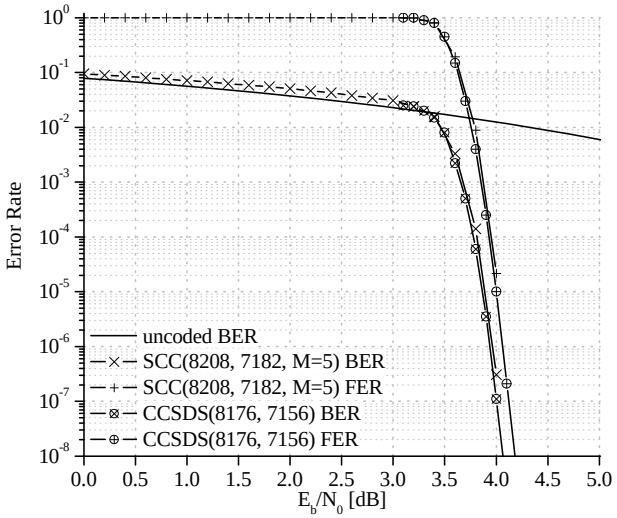



Figure 8. Performance of medium codes.

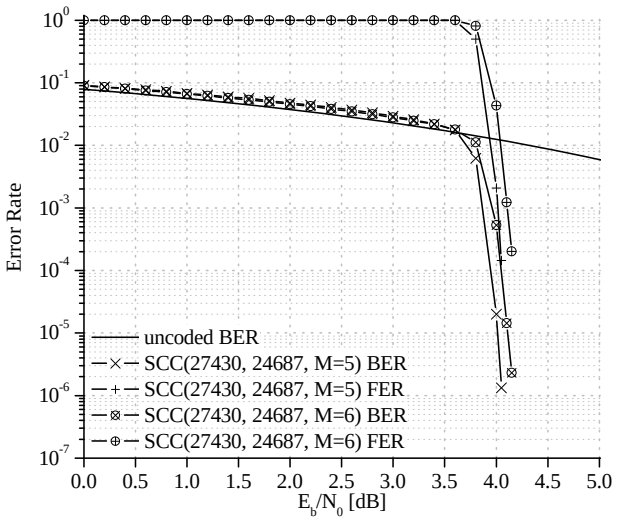



Figure 9. Performance of large codes with rate 0.9.

$M = 6$ serially concatenated codes. In this case, the chosen $r_i$ values are: $[313, 349, 401, 479, 563, 638]$.

Figure 9 shows the simulated performance of the two codes with $n = 27430$ and rate $0.9$. As we notice from the figure, the adoption of a higher number of component codes, in this case, does not give any advantage. In particular, both codes do not show any error floor in the explored region. The serially concatenated code with $M = 5$ components, however, has better performance in the waterfall region, and its BER curve intersects that of an uncoded transmission at a lower signal-to-noise ratio. So, it could be concluded that there is no need to increase the number of components over 5 for large codes with such a high rate.

However, when the code is shortened in such a way as to achieve lower rate, such conclusion is no more valid.

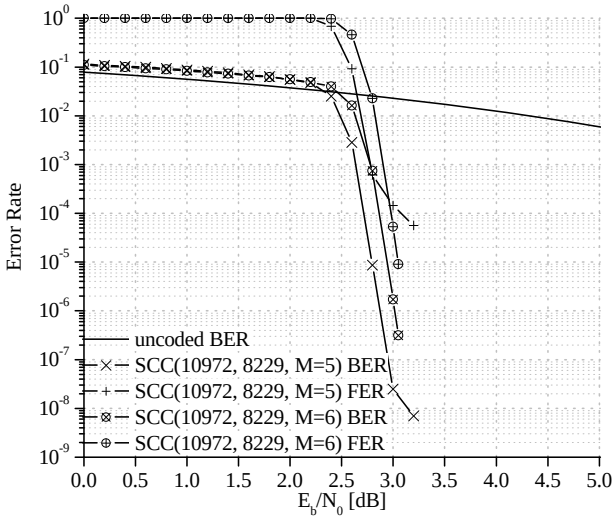



Figure 10. Performance of large codes with rate 0.75.

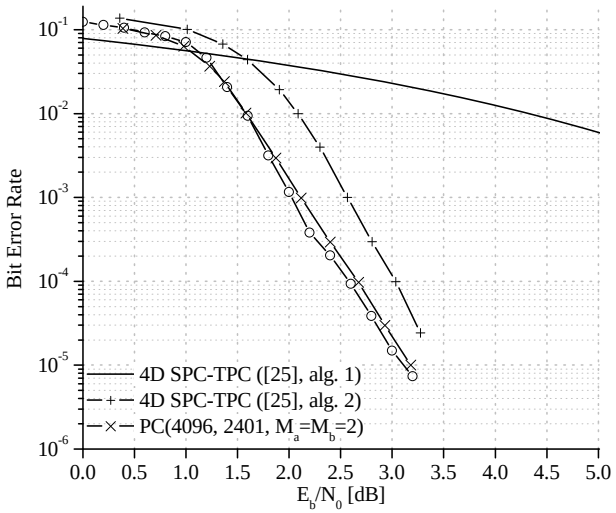



Figure 11. Comparison between $(4096, 2401)$ SPC-TPC and a product code with the same parameters.

The simulated performance of the two codes with length $n = 10972$ and rate $0.75$ is reported in Figure 10. As we notice from the figure, an error floor appears in the curves of the code with $M = 5$ components, though at $\mathrm{BER} < 10^{-7}$; on the other hand, the waterfall behavior of the code is very good. The presence of an error floor can be avoided, at the cost of a worse waterfall performance, by increasing the number of serially concatenated components up to $M = 6$. In this case, the error rate curves do not exhibit any change in their slope; so, they can intersect those of the code with $M = 5$ components. The FER curve for $M = 6$ intersects that for $M = 5$ at $\mathrm{FER} \simeq 10^{-4}$; an intersection is also expected for the BER curve.

## V. Examples of Product Codes

In this section, we provide some examples of bi-dimensional product codes that exploit, as components, two serially concatenated codes of the type described in Section II-B. For conciseness, we denote them as "product M-SC-MPC codes". The first example we consider is a product code with equal components. More precisely, each component code is a serially concatenated code having length $n_a = n_b = 64$, dimension $k_a = k_b = 49$ and $r_1 = 7$, $r_2 = 8$. Hence, the product code has length $n = 4096$, dimension $2401$ and rate $0.586$. Figure 11 shows its simulated performance by using the log-likelihood version of the SPA decoding algorithm. For the sake of comparison, the figure shows the performance of a 4D-TPC based on $(8, 7)$ SPC components [26], that has exactly the same parameters.

We observe that the bi-dimensional product code outperforms the 4D-SPC-TPC, when the latter is decoded through algorithm 1 in [26]. Instead, when adopting the decoding algorithm 2 in the same reference, performance of the two codes is almost the same. However, an important difference between the two codes is the fact that our product code is bi-dimensional, while the SPC-TPC is quadri-dimensional and, therefore, has larger decoding latency and complexity.

Two further examples are given in Figures 12 (for the BER) and 13 (for the FER), where two larger product M-SC-MPC codes are considered. The first code has $(n, k) = (10000, 5670)$ and has been obtained as the product of two different serially concatenated codes. Their parameters are as follows: $n_a = 100$, $k_a = 81$ and $r_j^a = [9, 10]$; $n_b = 100$, $k_b = 70$ and $r_j^b = [7, 11, 12]$.

The second code has $(n, k) = (12544, 6400)$; so, its rate is slightly reduced with respect to the former one. It adopts, as components, twice the same M-SC-MPC code. The latter has length $n_a = n_b = 112$, dimension $k_a = k_b = 80$ and $r_j^a = r_j^b = [8, 11, 13]$.

From the simulation results we see that, as expected, by using larger component codes, product M-SC-MPC codes can achieve better performance. The higher number of serially concatenated codes in each component allows to increase the minimum distance and improve performance in the error floor region. This is evident for the $(12544, 6400)$ code, whose components are both based on an order-3 serial concatenation. Together with its slightly lower rate, this makes the performance of such code better than that of the $(10000, 5670)$ code. It should also be noted that, differently from SPC-TPCs, in our codes the increase in minimum distance is achieved though preserving the bi-dimensional nature of the product code.

In order to better assess the effect on performance of the choice of $M_a$ and $M_b$, that is, the number of component codes in each one of the two serially concatenated codes forming the product code, we have designed three further

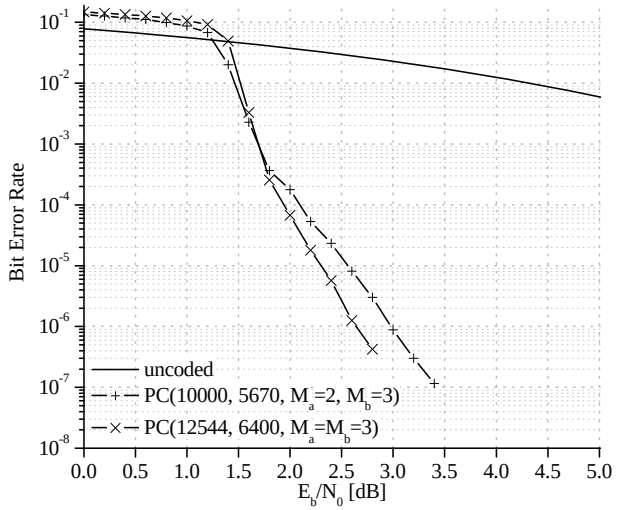



Figure 12. BER performance for $(10000, 5670)$ and $(12544, 6400)$ product codes.

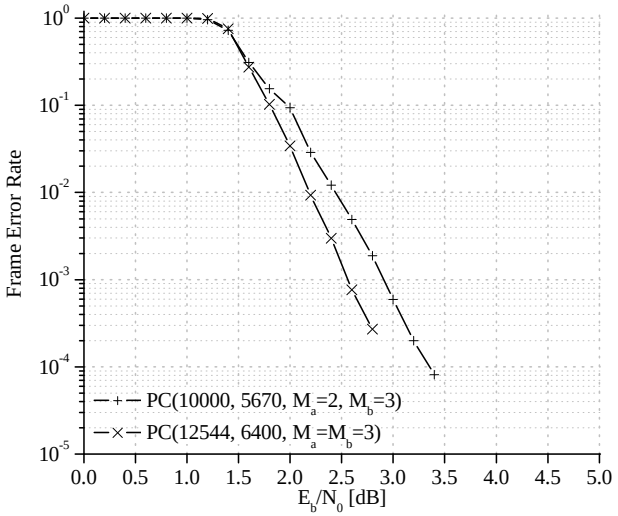



Figure 13. FER performance for $(10000, 5670)$ and $(12544, 6400)$ product codes.

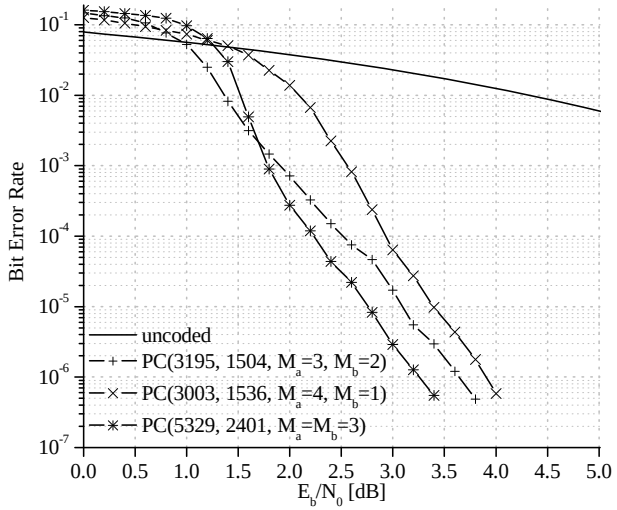



Figure 14. BER performance for $(3195, 1504)$, $(3003, 1467)$ and $(5329, 2401)$ product codes.

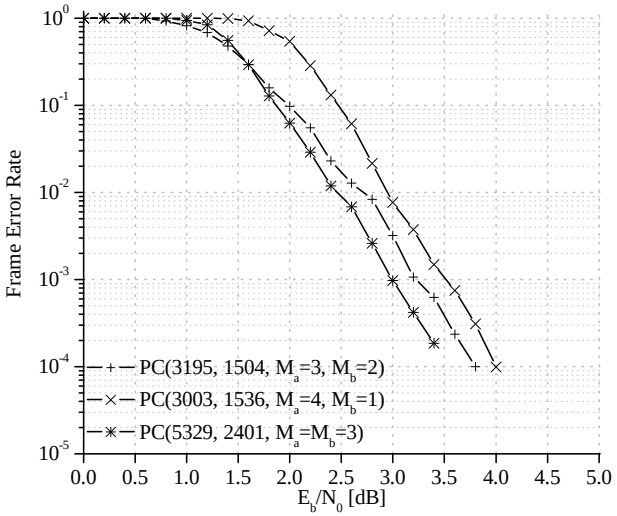



Figure 15. FER performance for $(3195, 1504)$, $(3003, 1467)$ and $(5329, 2401)$ product codes.

product codes with rate around $1/2$ and comparable size. The first code has length $3195$ and dimension $1504$; its two components have $n_a = 71$, $k_a = 47$, $r_j^a = [7, 8, 9]$ and $n_b = 45$, $k_b = 32$, $r_j^b = [6, 7]$, respectively. So, in this case, we have fixed $M_a = 3$ and $M_b = 2$. In the design of a second product code, we have adopted a different distribution of the serially concatenated components, that is, $M_a = 4$ and $M_b = 1$. The product code has length $n = 3003$ and dimension $k = 1467$. The first one of its two component codes has $n_a = 91$, $k_a = 48$ and $r_j^a = [9, 10, 11, 13]$. The second component, instead, is a single parity-check code with $n_b = 33$ and $k_b = 32$. As a further example, we have designed a third product code with $M_a = M_b = 3$. It uses twice the same serially concatenated code, that has length $n_a = n_b = 73$, dimension $k_a = k_b = 49$ and $r_j^a = r_j^b = [7, 8, 9]$. Thus, the product code has length $n = 5329$ and dimension $k = 2401$.

The simulated performance is reported in Figure 14, for the BER, and in Figure 15, for the FER. It results from numerical simulations that the code with $M_a = 3$ and $M_b = 2$ has better performance with respect to the code having $M_a = 4$ and $M_b = 1$, especially in the waterfall region. This suggests that a balanced choice of $M_a$ and $M_b$ can yield a performance improvement with respect to an unbalanced choice.

If both $M_a$ and $M_b$ are increased up to 3, performance can be further improved, at the cost of a larger code block, due to the constraints of the product structure. However, the

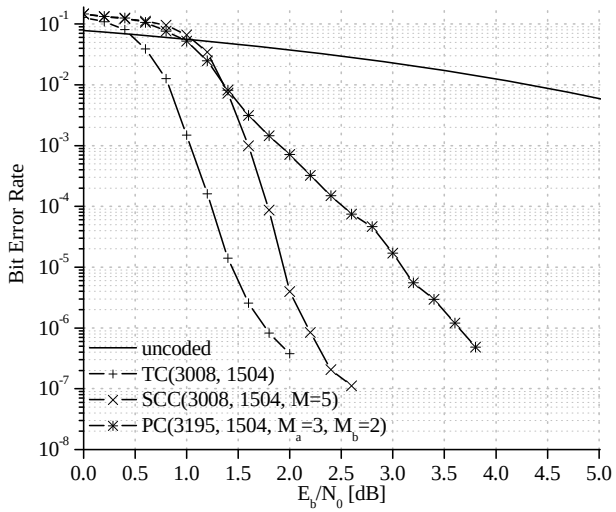



Figure 16. BER performance for codes with DVB-RCS compliant parameters.

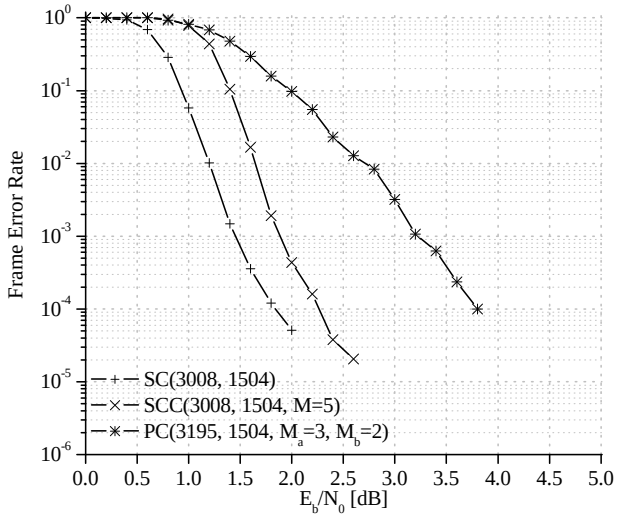



Figure 17. FER performance for codes with DVB-RCS compliant parameters.

$(5329, 2401)$ code is based on two very small component codes, so its complexity could still be reasonably low for many practical applications.

In comparison with single serially concatenated codes, it is reasonable to expect that the adoption of smaller component codes in a product code is paid in terms of a worse error rate performance. In order to verify such conclusion, we have considered a set of codes having parameters compliant with the DVB-RCS standard [27]. We have focused on the option of MPEG 2 frames, that are $188$ bytes long, and code rate $R = 1/2$. For this choice of the parameters, the standard recommends the usage of a $(3008, 1504)$ double binary turbo code with an optimized interleaver.

We have designed a serially concatenated code with exactly the same parameters. It has length $3008$ and adopts the following choice of the $r_i$'s for its components: $[277, 281, 293, 313, 340]$. For the sake of comparison, we have also considered one of the product codes introduced in the previous section. It has dimension $1504$, like in the standard, but length $3195$, because of the constraints due to the product code design. Its components have $n_a = 71$, $k_a = 47$ and $r_j^a = [7, 8, 9]$; $n_b = 45$, $k_b = 32$ and $r_j^b = [6, 7]$.

The simulated performance of the serially concatenated and the product code is reported in Figure 16, for the BER, and in Figure 17, for the FER. The simulated performance of the standard turbo code is also reported as a benchmark.

From the figures we see that the product code requires higher SNR per bit than the single serially concatenated code with $M = 5$; the loss at $\mathrm{BER} \simeq 10^{-6}$ and $\mathrm{FER} \simeq 10^{-4}$ is in the order of $1.5$ dB. This is the price to pay for reducing complexity by adopting a product code structure. In this example, where we have focused on low rate codes, the standard turbo code achieves the best performance, also in comparison with the M-SC-MPC code. On the contrary, at higher code rates, M-SC-MPC codes and, more generally, structured LDPC codes can be able to outperform turbo codes [28].

## VI. Conclusion

We have shown that the concatenation of very simple component codes can be effectively exploited in the design of structured LDPC codes. This allows to obtain a family of LDPC codes characterized by fine length and rate granularity, low complexity and rate compatibility. They ensure good error correction performance, and compete with codes optimized for specific applications. When large block codes are needed, a solution for continuing to exploit small components is to use serially concatenated codes in bi-dimensional product code structures. We have considered bi-dimensional product codes obtained as the direct product of two serially concatenated codes. We have shown that such product codes are still LDPC codes, and that their associated Tanner graph is suitable for performing LDPC decoding. Our simulations show that LDPC product codes in the considered class are able to achieve rather good performance in spite of their very low complexity. However, the advantage of using very small components in product code structures is paid in terms of coding gain: simulation results show a loss on the order of $1.5$ dB for LDPC product codes, compared with single LDPC codes with the same parameters.

As a further work, the introduction of an interleaver within the product code structure could be considered, in order to optimize the decoder performance in the waterfall region while preserving the multiplicative effect on the minimum distance due to the product code structure.

## REFERENCES

[1] M. Baldi, G. Cancellieri, and F. Chiaraluce, "A class of low-density parity-check product codes," in *Proc. SPACOMM 2009*, Colmar, France, Jul. 2009, pp. 107–112.

[2] G. D. Forney, Jr., "Concatenated codes," Ph.D. dissertation, MIT Press, Cambridge, MA, 1966.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE International Conference on Communications (ICC '93)*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[5] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[6] C. Sae-Young, G. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[7] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Cryptography and Coding. 5th IMA Conference*, ser. Lecture Notes in Computer Science, C. Boyd, Ed. Berlin: Springer, 1995, no. 1025, pp. 100–111.

[8] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[9] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[10] M. Baldi, G. Cancellieri, and F. Chiaraluce, "New LDPC codes based on serial concatenation," in *Proc. AICT '09*, Venice/Mestre, Italy, May 2009, pp. 31–315.

[11] ——. (2009) Good LDPC codes based on very simple component codes. [Online]. Available: http://www.gtti.it/GTTI09/Programma.php

[12] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[13] F. Babich, G. Montorsi, and F. Vatta, "Rate-compatible punctured serial concatenated convolutional codes," in *Proc. IEEE Globecom'03*, vol. 4, San Francisco, CA, Dec. 2003, pp. 2062–2066.

[14] ——, "Performance enhancement of partially systematic rate-compatible SCCCs through puncturing design," in *Proc. IEEE Globecom'04*, vol. 1, Dallas, TX, Nov. 2004, pp. 167–171.

[15] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1994.

[16] M. Baldi, G. Cancellieri, A. Carassai, and F. Chiaraluce, "LDPC codes based on serially concatenated multiple parity-check codes," *IEEE Commun. Lett.*, vol. 13, no. 2, pp. 142–144, Feb. 2009.

[17] J. S. K. Tee, D. P. Taylor, and P. A. Martin, "Multiple serial and parallel concatenated single parity-check codes," *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1666–1675, Oct. 2003.

[18] F. Chiaraluce and R. Garello, "Extended Hamming product codes analytical performance evaluation for low error rate applications," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2353–2361, Nov. 2004.

[19] P. Elias, "Error free coding," *IRE Trans. Inf. Theory*, vol. 4, no. 4, pp. 29–37, Sep. 1954.

[20] D. M. Rankin, T. A. Gulliver, and D. P. Taylor, "Asymptotic performance of single parity-check product codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 9, pp. 2230–2235, Sep. 2003.

[21] M. Esmaeili, "Construction of binary minimal product parity-check matrices," *Applicable Algebra in Engineering, Communication and Computing*, vol. 19, no. 4, pp. 339–348, Aug. 2008.

[22] O. Gazi and A. O. Yilmaz, "On parallelized serially concatenated codes," in *Proc. WCNC 2007*, Hong Kong, Mar. 2007, pp. 714–718.

[23] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[24] J. Chen and M. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, pp. 406–414, Mar. 2002.

[25] CCSDS, "Low Density Parity Check Codes for use in Near-Earth and Deep Space Applications," Consultative Committee for Space Data Systems (CCSDS), Washington, DC, USA, Tech. Rep. Orange Book, Issue 2, Sep. 2007, CCSDS 131.1-O-2.

[26] D. M. Rankin and T. A. Gulliver, "Single parity check product codes," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1354–1362, Aug. 2001.

[27] *Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems*, ETSI EN Std. 301 790 (v1.4.1), Sep. 2005.

[28] M. Baldi, F. Chiaraluce, and G. Cancellieri, "Finite-precision analysis of demappers and decoders for LDPC-coded M-QAM systems," *IEEE Trans. Broadcast.*, vol. 55, no. 2, pp. 239–250, Jun. 2009.

# An Iterative Algorithm for Compression of Correlated Sources at Rates Approaching the Slepian-Wolf Bound: Theory and Analysis

F. Daneshgaran, M. Laddomada, and M. Mondin

***Abstract*—This paper proposes a novel iterative algorithm based on Low Density Parity Check codes for compression of correlated sources at rates approaching the Slepian-Wolf bound. The setup considered in the paper looks at the problem of compressing one source without employing the source correlation, and employing the other correlated source as side information at the decoder which decompresses the first source. We demonstrate that depending on the extent of the source correlation estimated through an iterative paradigm, significant compression can be obtained relative to the case the decoder does not use the implicit knowledge of the existence of correlation. Two stages of iterative decoding are employed. During *global iterations* updated estimates of the source correlation is obtained and passed on to the belief-propagation decoder that performs *local iterations* with a pre-defined stopping criterion and/or a maximum number of local decoding iterations. Detailed description of the iterative decoding algorithm with embedded cross-correlation estimation are provided in the paper, in addition to simulation results confirming the potential gains of the approach.**

***Keywords*-Correlated sources; compression; iterative decoding; low density parity check codes; Slepian-Wolf.**

## I. INTRODUCTION

Consider two independent identically distributed (i.i.d.) discrete binary memoryless sequences of length $k$, $X = [x_1, x_2, \ldots, x_k]$ and $Y = [y_1, y_2, \ldots, y_k]$, where pairs of components $(x_i, y_i)$ have joint probability mass function $p(x, y)$. Assume that the two sequences are generated by two transmitters which do not communicate with each other, and that both sequences have to be jointly decoded at a common receiver. Slepian and Wolf demonstrated that the achievable rate region for this problem (i.e., for perfect recovery of both sequences at a joint decoder), is the one identified by the following set of equations imposing constraints on the rates $R_X$ and $R_Y$ by which both correlated sequences are transmitted:

$$\begin{cases} R_X \geq H(X|Y), \\ R_Y \geq H(Y|X), \\ R_X + R_Y \geq H(X,Y) \end{cases} \quad (1)$$

whereby, $H(X|Y)$ is the conditional entropy of source $X$ given source $Y$, $H(Y|X)$ is the conditional entropy of source $Y$ given source $X$, and $H(X,Y)$ is the joint entropy of the sources. A pictorial representation of this achievable region is given in Fig. 1.

Fred Daneshgaran is with the ECE Dept., Calif. State Univ., Los Angeles, USA. E-mail: fdanesh@calstatela.edu

Massimiliano Laddomada is with the Electrical Engineering Department of Texas A&M University-Texarkana, USA. E-mail: mladdomada@tamut.edu.

Marina Mondin is with the Dipartimento di Elettronica, Politecnico di Torino, Italy. E-mail: mondin@polito.it

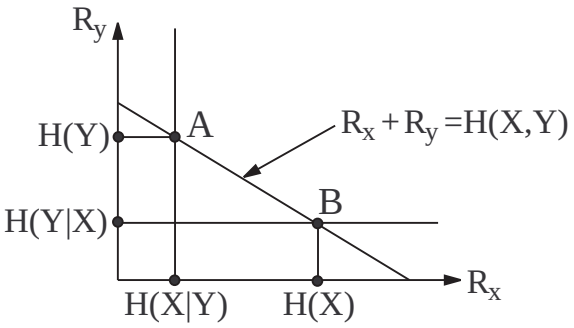



Fig. 1. Rate region for Slepian-Wolf encoding.

In this paper, which is an extended and thorough version of [1], we focus on trying to achieve the corner points $A$ and $B$ in Fig. 1, since any other point between these can be achieved with a time-sharing approach [2]. In particular, we focus on the architecture shown in Fig. 2 in which we assume that one of the two sequences, namely $X$ in our framework, is independently encoded with a source encoder that does not employ the correlation between the sources $X$ and $Y$. We assume that sequence $Y$ is compressed up to its source entropy $H(Y)$ and is known at the joint decoder as side information, and our aim is at compressing sequence $X$ with a rate $R_1 \leq R_X$ as close as possible to its conditional entropy $R_1 \geq H(X|Y)$ in order to achieve the corner point $A$ in Fig. 1. The decoder tries to decompress the sequence $X$, in order to obtain an estimate $\widehat{X}$, by employing $Y$ as side information and without the prior knowledge of the amount of correlation between the sequences. Indeed, it estimates this correlation through an iterative algorithm which improves the decoding reliability of $X$.

Obviously, our solution to joint source coding at point $A$ is directly applicable to point $B$ by symmetry. The overall rate of transmission of both sequences is greater than $H(Y) + H(X|Y) = H(X,Y)$.

With this background, let us provide a quick survey of the recent literature related to the problem addressed in this paper. This survey is by no means exhaustive and is meant to simply provide a sampling of the literature in this area.

In [3], [4], [5], the authors deal with applications of sensor networks for tracking and processing of information to be sent to a common destination. In particular, in [4], the authors inspired by information theory concepts and in particular, the

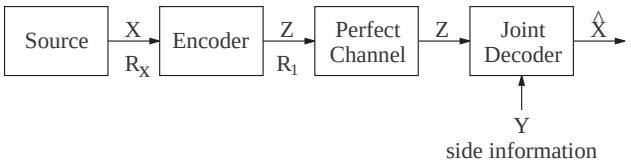



Fig. 2. Architecture of the encoder and joint decoder for the Slepian-Wolf problem.

Slepian-Wolf theorem [2], present the problem of distributed source coding using syndromes. The approach is based on the use of coset codes and significant compression gains are achieved using the proposed technique.

In [6], the authors show that turbo codes can allow to come close to the Slepian-Wolf bound in lossless distributed source coding. In [7], [8], [9], the authors propose a practical coding scheme for separate encoding of the correlated sources for the Slepian-Wolf problem. In [10], the authors propose the use of punctured turbo codes for compression of correlated binary sources whereby compression has been achieved via puncturing. The proposed source decoder utilizes an iterative scheme to estimate the correlation between two different sources. In [11], punctured turbo codes have been applied to the compression of non-binary sources.

The article [12] focuses on the problem of reducing the transmission rate in a distributed environment of correlated sources and the authors propose a source coding scheme for correlated images exploiting modulo encoding of pixel values and compression of the resulting symbols with binary and non-binary turbo codes.

Paper [13] deals with the use of irregular repeat accumulate codes as source-channel codes for the transmission of a memoryless binary sequence with side information at the decoder, while in [14] parallel and serial concatenated convolutional codes are considered for the same problem. In [15], [16], the authors propose a practical coding scheme based on Low Density Parity Check (LDPC) codes for separate encoding of the correlated sources for the Slepian-Wolf problem.

The dual problem of Slepian-Wolf correlated source coding over noisy channels has been dealt with in [17]-[21].

In [22] the authors consider the problem of encoding distributed correlated sources, demonstrating that separate source-channel coding is optimal in any network in which correlated sources do not interfere with each other, and that the information on a network behaves as a flow. Work [23] proposes a distributed joint source-channel coding scheme for multiple correlated sources.

Finally, in [24] the authors consider the design for joint distributed source and network coding.

Almost all proposed works in the literature use soft-metrics for improving the decoding performance. An excellent work describing the "mathematics of soft-decoding" is the paper by Hagenauer *et al.* [25].

Relative to the cited articles, the main novelty of the present work may be summarized as follows: 1) in [10] and [16] the encoder and decoder must both know the correlation between the two sources. We assume knowledge of mean correlation at the encoder. The decoder has implicit knowledge of this via observation of the length of the encoded message. It iteratively estimates the *actual* correlation observed and uses it during decoding; 2) our algorithm can be used with any pair of systematic encoder/decoder without modifying the encoding and decoding algorithm; 3) the proposed algorithm is very efficient in terms of the required number of LDPC decoding iterations. We use quantized integer LLR values (LLRQ) and the loss of our algorithm for using integer LLRQ metrics is quite negligible in light of the fact that it is able to guarantee performance better than that reported in [10] and [16] (where, to the best of our knowledge, authors use floating point metrics) as exemplified by the results shown in table II below; 4) we utilize post detection correlation estimates to generate extrinsic information, which can be applied to any already employed decoder without any modification; and 5) we do not use any interleaver between the sources at the transmitter. Using the approach of [10] in a network, information about interleavers used by different nodes must be communicated and managed. This is not trivial in a distributed network such as the internet. Furthermore, there is a penalty in terms of delay that is incurred.

This paper is an extended version of [1] and is organized as follows. Section II deals with the definition of the encoding algorithm for correlated sources, and presents the class of LDPC codes which is the focus of our current work. In section III, we present modification of the belief-propagation algorithm for decoding of LDPC codes, and follow up with details of our algorithm for iterative joint source decoding of correlated sources with side information and iterative correlation estimation. Section IV deals with the issue of sensitivity of the cross-correlation function between two sequences to residual errors after channel decoding, demonstrating the relative robustness of the empirical cross-correlation measure to channel induced errors. In section V, we present simulation results and comparisons confirming the potential gains that can be obtained from the proposed iterative algorithm. Finally, we present the conclusion in section VI.

## II. Architecture of the LDPC-based Source Encoder

This section focuses on the source encoder used for source compression. LDPC coding is essential to achieve performance close to the theoretical limit in [2].

The LDPC matrix [26] for encoding each source is considered as a systematic $(n, k)$ code. The codes used need to be systematic for the decoder to exploit the estimated correlation between $X$ and $Y$ directly.

Each codeword $C$ is composed of a systematic part $X$, and a parity part $Z$ which together form $C = [X, Z]$. With this setup and given the parity check matrix $H^{n-k,n}$ of the LDPC code, it is possible to decompose $H^{n-k,n}$ as follows:

$$H^{n-k,n} = (H^X, H^Z) \tag{2}$$

whereby, $H^X$ is a $(n-k) \times (k)$ matrix specifying the source bits participating in check equations, and $H^Z$ is a $(n-k) \times$

$(n-k)$ matrix of the form:

$$H^Z = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 \end{pmatrix}. \quad (3)$$

The choice of this structure for $H$, also called staircase LDPC (for the double diagonal of ones in $H^Z$), has been motivated by the fact that aside from being systematic, we obtain a LDPC code which is encodable in linear time in the codeword length $n$. In particular, with this structure, the encoding operation is as follows:

$$z_i = \begin{cases} \left[\sum_{j=1}^{k} x_j \cdot H^X_{i,j}\right] \ (mod\ 2), & i = 1 \\ \left[z_{i-1} + \sum_{j=1}^{k} x_j \cdot H^X_{i,j}\right] \ (mod\ 2), & i = 2,.,n-k \end{cases} \quad (4)$$

where, $H^X_{i,j}$ represents the element $(i,j)$ of the matrix $H^X$, and $x_j$ is the $j$-th bit of the source sequence $X$.

Source compression takes place as follows: considering the scheme shown in Fig. 2, we encode the length $k$ sequence belonging to the source $X$ and transmit on a perfect channel only the parity sequence $Z$, whose bits are evaluated as in (4). The rate guaranteed by such an encoder is

$$R_1 = \frac{n-k}{k}.$$

In relation to the setup shown in Fig. 2, the Slepian-Wolf problem reduces to that of encoding the source $X$ with a rate $R_1$ as close to $H(X|Y)$ as possible (i.e., $R_1 \geq H(X|Y)$). Note that source correlation is not employed at the encoder.

The objective of the joint decoder as explained in the next section, is to recover sequence $X$ by employing the correlated source $Y$ (considered as a side information at the decoder), and the estimates of the correlation between the sources $X$ and $Y$ obtained in an iterative fashion.

Before explaining the iterative decoding algorithm, let us briefly discuss the correlation model adopted in this framework. We consider the following model in order to follow the same framework pursued in the literature [10], [15]:

$$P(x_j \neq y_j) = p, \ \forall j = 1, \dots, k \quad (5)$$

In light of the considered correlation model, and noting that the sequence $Y$ is available losslessly at the joint decoder, the theoretical limit for lossless compression of $X$ is

$$R_1 \geq H(X|Y) = H(p),$$

whereby $H(p)$ is the binary entropy function defined as

$$H(p) = -p\log_2(p) - (1-p)\log_2(1-p).$$

With this setup, the following holds:

$$R_1 + R_Y \geq H(X,Y) \rightarrow R_1 + 1 \geq H(p) + 1. \quad (6)$$

Note that the encoder needs to know the mean correlation so as to choose a rate close to $H(p)$. It does so, by keeping $k$ constant while choosing $n$ appropriately. We use the term mean correlation, because in any actual setting, the exact correlation between the sequences may be varying about the mean value. Hence, it is beneficial if the decoder estimates the actual correlation value from observations itself. While no side information about the rate is communicated to the decoder, the decoder knows the mean correlation implicitly from the knowledge of block length $n$.

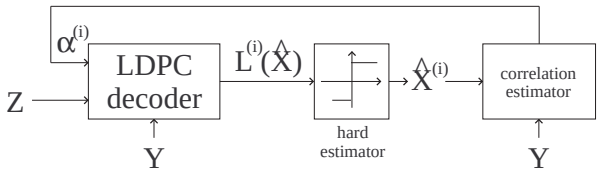



Fig. 3. Architecture of the Iterative Joint decoder of correlated sources.

## III. Joint Iterative LDPC-Decoding of Correlated Sources

The architecture of the iterative joint decoder for the Slepian-Wolf problem is depicted in Fig. 3. Its goal is to determine the best estimate $\widehat{X}$ of the source $k$-sequence $X$, by starting from the received parity bit $(n-k)$-sequence $Z$. We note that aside from the fact that the receiver may a-priori presume some correlation between both sequences exist, no side information on this correlation is communicated to the joint decoder. Indeed, this correlation is estimated in an iterative fashion.

With reference to the architecture of the joint decoder depicted in Fig. 3, we note that there are two stages of iterative decoding. Index $i$ denotes a *global iteration* whereby during each global iteration, the updated estimate of the source correlation obtained during the previous global iteration is passed on to the belief-propagation decoder that performs *local iterations* with a pre-defined stopping criterion and/or a maximum number of local decoding iterations.

Based on the notation above, we can now develop the algorithm for exploiting the source correlation in the LDPC decoder. Consider a $(n,k)$-LDPC identified by the matrix $H^{(n-k,n)}$ as expressed in (2). Note that we only make reference to maximum rank matrix $H$ since the particular structure assumed for $H$ ensures this. In particular, the double diagonal on the parity side of the $H$ matrix always guarantees that the rank of $H$ is equal to the number of its rows, i.e., $n-k$.

It is well known that the parity check matrix $H$ can be described by a bipartite graph with two types of nodes as shown in Fig. 4; $n$ bit-nodes corresponding to the LDPC code bits, and $n-k$ check-nodes corresponding to the parity checks as expressed by the rows of the matrix $H$. Following the notation employed in [29], let $B(m)$ denote the set of bit-nodes connected to the $m$-th check-node, and $C(n)$ denote the set of check-nodes adjacent to the $n$-th bit-node. With this setup, $B(m)$ corresponds to the set of positions of the 1's in the $m$-th row of $H$, while $C(n)$ is the set of positions of the 1's in the $n$-th column of $H$. In addition, let us use the notation $C(n) \setminus m$ and $B(m) \setminus n$ to mean the sets $C(n)$ and $B(m)$ in which the $m$-th check-node and the $n$-th bit-node respectively, are excluded. Furthermore, let us identify with $\lambda_{n,m}(u_n)$ the log-likelihood of the message that the $n$-th bit-node sends to

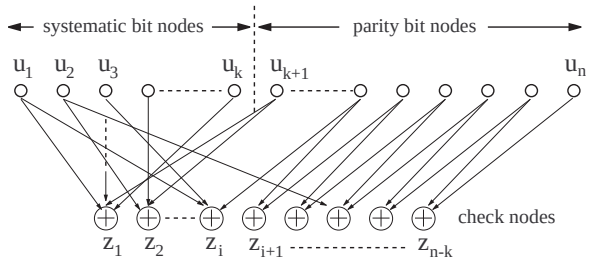



Fig. 4. Pictorial representation of the Tanner graph of a staircase LDPC code.

the $m$-th check-node, that is, the LLR of the probability that $n$-th bit-node is 1 or 0 based on all checks involving the $n$-th bit except the $m$-th check, and with $\Lambda_{m,n}(u_n)$ the log-likelihood of the message that the $m$-th check-node sends to the $n$-th bit-node, that is, the LLR of the probability that the $n$-th bit-node is 1 or 0 based on all the bit-nodes checked by the $m$-th check except the information coming from the $n$-th bit-node. With this setup, we have the following steps of the belief-propagation algorithm:

*Initialization Step:* each bit-node is assigned an a-posteriori LLR $L(u_j)$ as follows:

$$\begin{cases} \log\left(\frac{P(x_j=1|y_j)}{P(x_j=0|y_j)}\right) = (2y_j-1)\,\alpha^{(i)}, & j=1,\ldots,k \\ (2z_j-1), & j=k+1,\ldots,n \end{cases} \tag{7}$$

whereby,

$$\alpha^{(i)} = \log\left(\frac{p^{(i)}}{1-p^{(i)}}\right)$$

is the correction factor taking into account the estimated correlation between sequences $X$ and $Y$ at global iteration $i$. Note that this term derives from the correlation model adopted in this paper as expressed in (5), in which the correlation between any bit in the same position in the two sequences $X$ and $Y$ is seen as produced by an equivalent binary symmetric channel with transition probability $p$.

In our setup, before the first global iteration, $\alpha^{(0)} = 1$ since the degree of correlation is not known a priori.

In summary, for any position $(m,n)$ such that $H_{m,n} = 1$, set:

$$\lambda_{n,m}(u_n) = L(u_n), \tag{8}$$

and

$$\Lambda_{m,n}(u_n) = 0. \tag{9}$$

*(1) Check-node update:* for each $m = 1,\ldots,n-k$, and for each $n \in B(m)$, compute:

$$\Lambda_{m,n}(u_n) = 2\tanh^{-1}\left(\prod_{p\in B(m)\setminus n}\tanh\left(\frac{\lambda_{p,m}(u_p)}{2}\right)\right) \tag{10}$$

*(2) Bit-node update:* for each $t = 1,\ldots,n$, and for each $m \in C(t)$, compute:

$$\lambda_{t,m}(u_t) = L(u_t) + \sum_{p\in C(t)\setminus m}\Lambda_{p,n}(u_t). \tag{11}$$

*(3) Decision:* for each bit node $u_t$ with $t = 1,\ldots,n$, compute:

$$\lambda_t(u_t) = L(u_t) + \sum_{p\in C(t)}\Lambda_{p,n}(u_t), \tag{12}$$

and quantize the results such that $u_t = 0$ if $\lambda_t(u_t) < 0$, and $u_t = 1$ otherwise.

If $H \cdot U^T = \mathbf{0}$ then halt the algorithm and output $\widehat{x}_t = u_t,\ t = 1,\ldots,k$ as the estimate of the decompressed source bits, $X$, corresponding to the first source. Otherwise, if the number of iterations is less than a predefined maximum number, iterate the process starting from step $(1)$.

The architecture of the iterative joint decoder is depicted in Fig. 3. Let us elaborate on the signal processing involved. In particular, as before let $x$ and $y$ be two correlated binary random variables which can take on the values $\{0,1\}$ and let $r = x \oplus y$. Let us assume that random variable $r$ takes on the values $\{0,1\}$ with probabilities $P(r=1) = p_r$ and $P(r=0) = 1-p_r$.

The correction factor $\alpha^{(i)}$ at global iteration $(i)$ is evaluated as follows,

$$\alpha^{(i)} = \log\left(\frac{p_{\hat{r}}}{1-p_{\hat{r}}}\right), \tag{13}$$

by counting the number of places in which $\widehat{X}^{(i)}$ and $Y$ differ, or equivalently by evaluating the Hamming weight $w_H(.)$ of the sequence $\widehat{R}^{(i)} = \widehat{X}^{(i)} \oplus Y$ whereby, in the previous equation,

$$p_{\hat{r}} = \frac{w_H(\widehat{R}^{(i)})}{k}.$$

In the latter case, by assuming that the sequence $\widehat{R} = \widehat{X} \oplus Y$ is i.i.d., we have:

$$\alpha^{(i)} = \log\left(\frac{w_H(\widehat{R}^{(i)})}{k - w_H(\widehat{R}^{(i)})}\right) \tag{14}$$

where $k$ is the source block size. Above, letters highlighted with $\widehat{.}$ are used to mean that the respective parameters have been estimated.

Formally, the iterative decoding algorithm can be stated as follows:

1) Set the log-likelihood ratios $\alpha^{(0)}$ to one (see Fig. 3). Compute the log-likelihood ratios for any bit node using (7).
2) For each global iteration $i = 1,\ldots,q$, do the following:
   a) perform a belief-propagation decoding on the parity bit sequence $Z$ by using a predefined maximum number of local iterations, and the side information represented by the correlated sequence $Y$ along with the correction factor $\alpha^{(i-1)}$;
   b) Evaluate $\alpha^{(i)}$ using (14);
   c) If $\left|\alpha^{(i)} - \alpha^{(i-1)}\right| \geq 10^{-4}$ go back to (a) and continue iterating, else exit.

Point c) in the previous code fragment is used in order to speed-up the overall iterative algorithm. Extensive tests we conducted suggested that the threshold value of $10^{-4}$ may be used for this purpose. Obviously, one can keep iterating until the last global iteration as well.

### A. Overview of Integer-Metrics Belief-Propagation Decoder

In this section, we briefly describe the LDPC decoder working with integer LLRs. This approach leads to efficient belief-propagation decoding. Following the rationales described in [27], we begin by quantizing any real LLR (denoted LLRQ after quantization) employed in the initialization phase of the belief-propagation decoder in (7), using the following transformation:

$$LLRQ = \begin{cases} \lfloor 2^q L(u_j) + 0.5 \rfloor, & j = 1, \ldots, k \\ \lfloor 2z_j - 1 \rfloor \cdot S, & j = k+1, \ldots, n \end{cases} \quad (15)$$

whereby $\lfloor \cdot \rfloor$ stands for rounding to the smaller integer in the unit interval in which the real number falls, $L(u_j)$ is the real LLR, $S$ is a suitable scaling factor, and $q$ is the precision chosen to represent the LLR with integer metrics. In our belief-propagation decoder, we use $q = 3$, which guarantees a good trade-off between BER performance and complexity of the decoder implementation [27]. The scaling factor $S$ is the greatest integer metric processed by the iterative decoder. In our set-up, we use $S = 10000$. Note that such a scaling factor depends on the practical implementations of the belief-propagation decoder. Suffice it to say that in our setup, $S$ gives high likelihood to the parity bits $z_j,\ \forall j = k+1, \ldots, n$, since they are transmitted through a perfect channel to the decoder.

All the summations shown in equations (11) and (12) are performed on integer LLRs. Finally, (10) has to be modified for working with integer metrics. To this end, we follow the schedule proposed in [29] in order to avoid real calculations and only use look-up tables and integer additions. This way, the basic operation $L(x \oplus y)$ in the check node update is approximated as follows:

$$L(x \oplus y) \approx \text{sign}(L(x)) \cdot \text{sign}(L(y)) \cdot \min(|L(x)|, |L(y)|) + \log\left(1 + e^{L(x)+L(y)}\right) - \log\left(e^{L(x)} + e^{L(y)}\right) \quad (16)$$

whereby, sign$(\cdot)$ is the function which evaluates the sign of the enclosed LLR. The latter two log functions are evaluated through look-up tables containing samples of the log functions approximated with piecewise linear approximation as suggested in [29].

## IV. Sensitivity of the Cross-Correlation to Residual Errors After Channel Decoding

In this section, we wish to demonstrate the relative robustness of the empirical cross-correlation between sequences $\widehat{X}$ and $Y$ to residual errors after channel decoding. To this end, let $X$ and $Y$ be two binary vectors of length $k$. Correlation between two binary streams can be measured in two ways. One technique is to map a logic-0 to the integer $+1$, logic-1 to $-1$ and to take the inner product of the two vectors defining our data packets and divide by the vector length (assumed to be the same for both packets). It is well known that the resulting empirical correlation estimate lies in the range $-1 \leq \rho \leq +1$. Positive value of $\rho$ signifies a similarity between $X$ and $Y$, while a negative $\rho$ signifies a similarity between $X$ and $\bar{Y}$, where $\bar{Y}$ is the complement of vector $Y$. This traditional measure of correlation is not of direct interest to us since we wish to relate a correlation measure to an empirical probability which is strictly positive.

To this end, let us define $r_n = x_n \oplus y_n$ as the XOR of the $n$-th component of the vectors $X$ and $Y$. Similarly, we define $R = X \oplus Y$ whereby $R$ is obtained via componentwise XOR of the components of the vectors $X$ and $Y$. Let the number of places in which $X$ and $Y$ agree be $\gamma$ and define the empirical cross-correlation between these two vectors as $\rho = \gamma/k$. Note that with this definition, we may take $\rho$ as the empirical probability that $r_n = 0$ and a value of for instance $\rho = 0.7$ and $\rho = 0.3$ would have the same *information content* as measured by the value of the binary entropy function. Notice that, since intrinsic soft information is generated from probability estimates, this measure of correlation is more useful to us than the traditional definition.

Let us consider the Slepian-Wolf problem discussed above, and suppose that what is available at the receiver is a noisy version of $X$ denoted $\widehat{X}$. For instance, $\widehat{X}$ could be an erroneous version of $X$ obtained after transmission through a noisy channel modelled as a Binary Symmetric Channel (BSC) with transition probability $p_e$. We assume that the error events inflicting the sequence $X$ are independent identically distributed (i.i.d.). The joint decoder generates an empirical estimate of the cross-correlation based on the use of the sequences $\widehat{X}$ and $Y$ by forming the vector $\widehat{R} = \widehat{X} \oplus Y$ and counting the number of places where $\widehat{R}$ is zero. Let us denote this count as $\hat{\gamma}$. Clearly, $\hat{\gamma}$ is a random variable. The question is, what is the Probability Mass Function (PMF) of $\hat{\gamma}$? Knowledge of this PMF allows us to assess the sensitivity of our estimate of the cross-correlation to errors in the original sequence $X$.

It is relatively straightforward to express the probability that $(\hat{r}_n = r_n)$ and $(\hat{r}_n \neq r_n)$ as $P(\hat{r}_n = r_n) = P(\hat{x}_n = x_n) = 1 - p_e,\ P(\hat{r}_n \neq r_n) = P(\hat{x}_n \neq x_n) = p_e$. Consider applying a permutation to the sequences $X$ and $Y$ so that the permuted sequences agree in the *first* $\gamma$ locations, and disagree in the remaining $(k - \gamma)$ locations. The permutation is applied to simplify the explanation of how we may go about obtaining the PMF of $\hat{\gamma}$ and by no means impacts the results. It is evident that the permuted sequence $\pi(R)$ contains $\gamma$ zeros in the first $\gamma$ locations and $(k-\gamma)$ ones in the remaining locations. Now consider evaluation of the probability $P(\hat{\gamma} = \gamma + \nu)$ for $\nu = 0, 1, .., (k - \gamma)$. We define $\pi(R)_\gamma$ to represent the first $\gamma$ bits of $\pi(R)$ and $\pi(R)_{k-\gamma}$ the remaining $(k - \gamma)$ bits. Similarly we define $\pi(\widehat{R})_\gamma$ and $\pi(\widehat{R})_{k-\gamma}$. For a fixed $\nu$, the event $\{\hat{\gamma} = \gamma + \nu\}$ corresponds to the union of the events of the type: $\pi(\widehat{R})_{k-\gamma}$ differs from $\pi(R)_{k-\gamma}$ in $(\nu + l)$ positions for some $l \in \{0, 1, .., \gamma\}$, $\pi(\widehat{R})_\gamma$ differs from $\pi(R)_\gamma$ in $l$ positions, and the remaining bits of $\pi(\widehat{R})$ and $\pi(R)$ are identical. The probability of such elementary events are given by:

$$\binom{\gamma}{l} \binom{k-\gamma}{\nu+l} (1 - p_e)^{k-\nu-2l} (p_e)^{\nu+2l}. \quad (17)$$

It can be shown that the probability of the event $\{\hat{\gamma} = \gamma + \nu\}$ for $\nu = 0, 1, .., (k - \gamma)$ is given by:

$$P(\hat{\gamma} = \gamma + \nu) = \sum_{l=0}^{\gamma} \left[ \binom{\gamma}{l} \binom{k-\gamma}{\nu+l} \cdot \right.$$

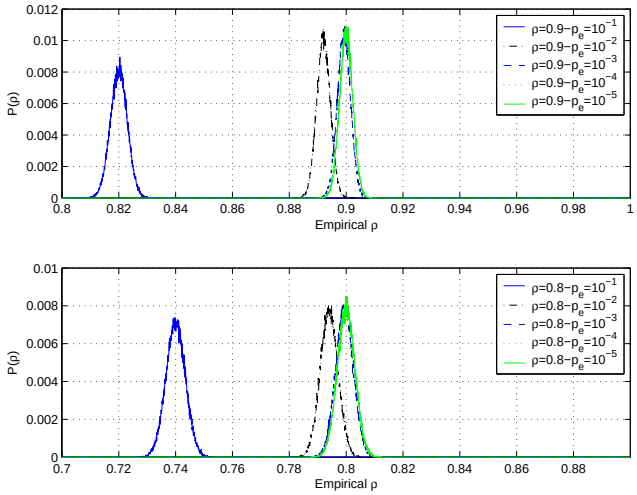



Fig. 5. Probability mass function (PMF) of $\hat{\rho}$ for three different values of raw error rate $p$ when the true cross-correlations between the data packets are $\rho = 0.8$ and $0.9$ at data block length of $k = 16400$.

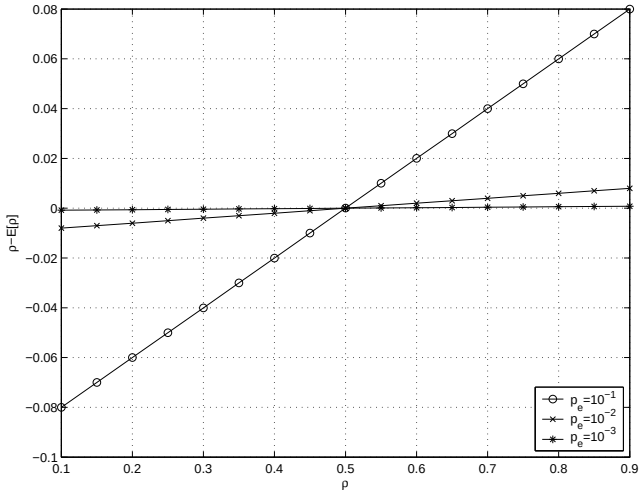



Fig. 6. The difference $(\rho - M(\hat{\rho}))$ versus $\rho$ whereby $M(\hat{\rho})$ denotes the most probable value of $\hat{\rho}$ obtained from empirical evaluation of the cross-correlation from noisy received vectors (block length $k = 16400$).

$$\cdot(1-p_e)^{k-\nu-2l}(p_e)^{\nu+2l}\Big]\,, \tag{18}$$

while, for $m = 1, 2, .., \gamma$ we have:

$$P(\hat{\gamma} = \gamma - m) = \sum_{l=m}^{\gamma}\left[\begin{pmatrix}\gamma\\ l\end{pmatrix}\begin{pmatrix}k-\gamma\\ l-m\end{pmatrix}\cdot \right.$$
$$\left.\cdot(1-p_e)^{k-2l+m}(p_e)^{2l-m}\right]. \tag{19}$$

In the next section, we shall present simulation results for iterative joint decoding of correlated sources for data packet size of $k = 16400$. Hence, let us look at representative results associated with the estimation of $\rho$ for this block length:

1) Fig. 5 depicts the PMF of $\hat{\rho}$ for five different values of raw error rate $p_e$ when the true cross-correlations between the data packets are $\rho = 0.8$ and $0.9$ at data block length of $k = 16400$. The following observation is in order: there is a bias in the empirical estimate of $\rho$ as measured by the most probable value of $\hat{\rho}$ and the true value of $\rho$. This bias is a strong function of $p_e$.
2) As noted above, the most likely value of $\hat{\rho}$, denoted $M(\hat{\rho})$ (i.e., the Mode), obtained from evaluation of the empirical cross-correlation from noisy received vectors is not necessarily the true value $\rho$. This is particularly so at larger values of $p_e$ and for small and large values of $\rho$. Fig. 6 captures this behavior for three values of $p_e = 10^{-1}$, $10^{-2}$ and $p_e = 10^{-3}$ as a function of $\rho$ for block length $k = 16400$. In particular, this figure shows the difference $(\rho - M(\hat{\rho}))$ versus $\rho$ obtained from empirical evaluation of the cross-correlation from noisy received vectors.
3) The standard deviation of $\hat{\rho}$ is a strong function of $p_e$ itself. Fig. 7 depicts the standard deviation of $\hat{\rho}$ as a function of $p_e$ for $k = 16400$ for three different values of $\rho$. This figure shows that the standard deviation increases slowly with increasing $p_e$. Note that even at values of $p_e$ as large as $p = 0.1$ this standard deviation is still relatively small for $\rho$ in the range $\rho = 0.1$ to $\rho = 0.9$.

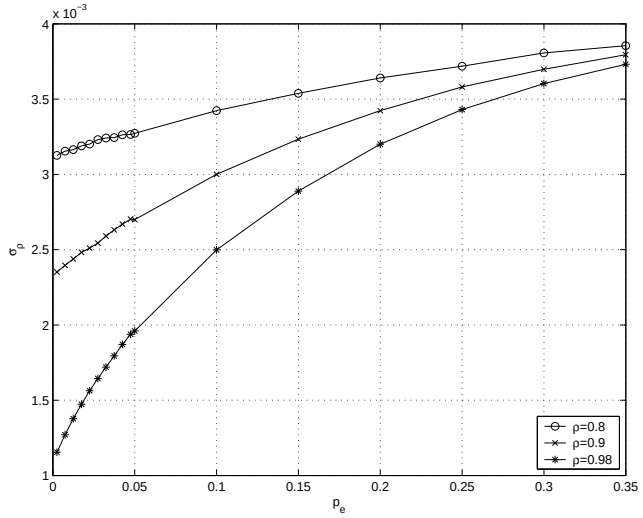



Fig. 7. Standard deviation of $\hat{\rho}$ as a function of $p$ (block length $k = 16400$).

## V. Simulation Results and Comparisons

We have simulated the performance of our proposed iterative joint source decoder. In order to compare our results with others proposed in the literature, we follow the same framework as in [10], [15], [16].

In the following, we provide sample simulation results associated with various $(n, k)$ LDPC codes designed with the technique proposed in [28]. In particular, for a fair comparison with the results provided in [16], we designed various LDCP codes with $k = 16400$ and $k = 16000$. The details and the parameters of the designed LDPCs are given in Tables I-II.

Parameters given in Table I are the source block length $k$, the codeword length $n$, the rate $R_1$ of the source, expressed as $\frac{n-k}{k}$ (i.e., inverse of the compression ratio), the average degree $d_v$ of the bit nodes, and the average degree $d_c$ of the check nodes of the designed LDPCs. Note that, the encoding procedure adopted in our approach is different from the one proposed in [16] in that we source encode $k$ bits at a time

TABLE I
PARAMETERS OF THE DESIGNED LDPCS.

| LDPC | $k$ | $n$ | $R_1$ | $d_v$ | $d_c$ |
|---|---|---|---|---|---|
| $L_1$ | 16400 | 30750 | 0.875 | 2.8 | 6 |
| $L_2$ | 16400 | 26200 | 0.597 | 3 | 8 |
| $L_3$ | 16400 | 22400 | 0.365 | 3.21 | 12 |
| $L_4$ | 16000 | 20000 | 0.25 | 3.59 | 18 |
| $L_5$ | 16400 | 20300 | 0.237 | 3.45 | 18 |
| $L_6$ | 16400 | 20000 | 0.219 | 3.23 | 18 |
| $L_7$ | 16400 | 19300 | 0.176 | 3.0 | 20 |
| $L_8$ | 16400 | 19500 | 0.189 | 3.0 | 19 |

and transmit only $n-k$ bits. In [16], the authors proposed a source compression which encodes $n$ source bits at a time, and transmits $n-k$ syndrome bits.

The degree distributions of the designed LDPC codes, labelled $L_1$ to $L_8$ in Table I, are shown in Table II whereby, the integer number multiplying any power of the indeterminate $x^w$ identifies the number of bit nodes in $\lambda_{L_r}(x)$, and check nodes in $\rho_{L_r}(x)$ having degree $w+1$ in the $r$-th designed LDPC with $r = 1, \ldots, 8$.

For local decoding of the LDPC codes, the maximum number of local iterations has been set to 50, while the maximum number of global iterations is 5, even though the stopping criterion discussed in the previous section has been adopted.

The results on the compression achieved with the proposed algorithm are shown in Table III. The first row shows the correlation parameter assumed, that is, the probability $p = P(x_j \neq y_j),\ \forall j = 1, \ldots, k$. The second row shows the joint entropy limit as expressed in (6) for various values of the correlation parameter $p$. The third and fourth rows show the results on source compression presented in papers [10], [16], while the last row presents the results on compression achieved with the proposed algorithm employing a maximum of 5 global iterations in conjunction with using the stopping criterion noted in the previous section. As in [16], we assume error free compression for a target Bit Error Rate (BER) $10^{-6}$. Note that statistic of the results shown has been obtained by counting 30 erroneous frames.

From Table III it is evident that significant compression gains with respect to the theoretical limits can be achieved as the correlation between sequences $X$ and $Y$ increases. Notice that as opposed to the algorithm proposed in [16], we do not assume $p$ is known at the decoder. Indeed, this parameter is iteratively estimated as discussed in the previous section. A similar approach has been pursued in [10], whereby an iterative approach is employed for the estimation of the correlation between the two correlated sequences, but employing turbo codes.

Finally, note that we employ integer soft-metrics as explained in the previous section, while in [10], [16], to the best our knowledge, the authors employ real metrics. The algorithm working on integer metrics is very fast and reduces considerably the complexity burden required by the two-stage iterative algorithm (i.e., the local-global combination).

Fig. 8 shows the BER performance of the proposed iterative decoding algorithm for various global iterations and as a function of the joint entropy between sources $X$ and $Y$, when the stopping criterion for global iterations is applied and the maximum number of global iterations specified in the legend are used. The LDPC used for encoding is the one labelled $L_8$ in Tables I-II which guarantees a compression rate of $R_1 = 0.189$. At BER$= 10^{-6}$, the overall rate $R = R_1 + R_Y = 1.189$ is within 0.075 of the theoretical limit of 1.1175.

In order to assess the effects of the global iterations on BER performances, we simulated the same LDPC $L_8$ without stopping criterion on the global iterations. The resulting BER performance are shown in Fig. 9. Notice that 3- or 4- global iterations suffice to get almost all that can be gained from the knowledge of the cross-correlation.

In Table IV, we show the average number of local iterations

TABLE II
DEGREE DISTRIBUTIONS OF THE DESIGNED LDPCS.

| | |
|---|---|
| $\lambda_{L_1}(x)$ | $1 + 14349x + 15400x^2 + 200x^3 + 800x^{12}$ |
| $\rho_{L_1}(x)$ | $x^4 + 14349x^5$ |
| $\lambda_{L_2}(x)$ | $1 + 9799x + 11000x^2 + 4700x^3 + 700x^9$ |
| $\rho_{L_2}(x)$ | $x^6 + 9799x^7$ |
| $\lambda_{L_3}(x)$ | $1 + 5999x + 13700x^2 + 1800x^3 + 900x^{12}$ |
| $\rho_{L_3}(x)$ | $x^{10} + 5999x^{11}$ |
| $\lambda_{L_4}(x)$ | $1 + 3999x + 12600x^2 + 1300x^3 + 2100x^9$ |
| $\rho_{L_4}(x)$ | $x^{16} + 3999x^{17}$ |
| $\lambda_{L_5}(x)$ | $1 + 3899x + 12800x^2 + 2400x^3 + 1200x^{11}$ |
| $\rho_{L_5}(x)$ | $x^{16} + 3899x^{17}$ |
| $\lambda_{L_6}(x)$ | $1 + 3599x + 14400x^2 + 1200x^3 + 800x^{11}$ |
| $\rho_{L_6}(x)$ | $x^{16} + 3599x^{17}$ |
| $\lambda_{L_7}(x)$ | $1 + 2899x + 15400x^2 + 750x^3 + 250x^{11}$ |
| $\rho_{L_7}(x)$ | $x^{18} + 2899x^{19}$ |
| $\lambda_{L_8}(x)$ | $1 + 3099x + 15900x^2 + 500x^9$ |
| $\rho_{L_8}(x)$ | $x^{17} + 3099x^{18}$ |

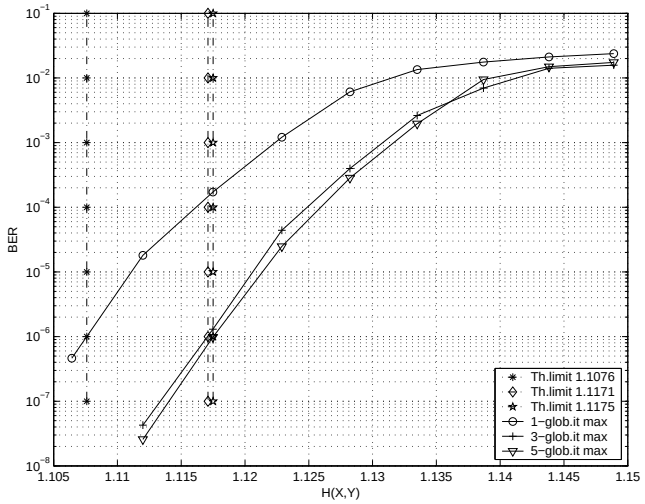



Fig. 8. BER performance of the proposed iterative decoding algorithm for various global iterations as a function of the joint entropy between sources $X$ and $Y$, when the stopping criterion for global iterations is applied. Results refer to the LDPC labelled $L_8$ in Tables I-II, guaranteeing a compression rate of $R_1 = 0.189$. For a target BER$= 10^{-6}$, we also show the theoretical limit as expressed by the Slepian-Wolf bound achieved by the iterative algorithm applied to the LDPC $L_8$ for increasing values of the maximum number of global iterations.

TABLE III
COMPRESSION RATE PERFORMANCE OF THE ITERATIVE ALGORITHM FOR VARIOUS JOINT ENTROPIES.

| $p$ | 0.01 | 0.02 | 0.025 | 0.02875 | 0.05 | 0.1 | 0.2 |
|---|---|---|---|---|---|---|---|
| $H(p)+1$ | 1.08 | 1.141 | 1.169 | 1.188 | 1.286 | 1.469 | 1.722 |
| $R$ [10] | - | - | 1.31 | - | 1.435 | 1.63 | 1.89 |
| $R$ [16] | - | - | 1.276 | - | 1.402 | 1.60 | 1.875 |
| $R = R_1 + R_Y$ | 1.176-$L_7$ | 1.219-$L_6$ | 1.237 -$L_5$ | 1.25-$L_4$ | 1.365-$L_3$ | 1.597-$L_2$ | 1.875-$L_1$ |

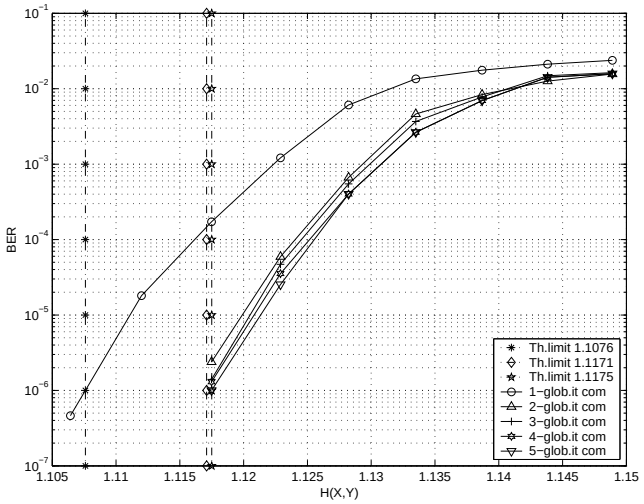



Fig. 9. BER performance of the proposed iterative decoding algorithm for various global iterations as a function of the joint entropy between sources $X$ and $Y$, without stopping criterion on global iterations. Results refer to the LDPC labelled $L_8$ in Tables I-II, guaranteeing a compression rate of $R_1 = 0.189$. For a target BER$= 10^{-6}$, we also show the theoretical limit as expressed by the Slepian-Wolf bound achieved by the iterative algorithm applied to the LDPC $L_8$ for increasing values of the maximum number of global iterations.

TABLE IV
AVERAGE NUMBER OF LOCAL ITERATIONS FOR LDPC $L_8$ DECODING WITH MAXIMUM LOCAL ITERATIONS PARAMETER SET TO 50, FOR DIFFERENT GLOBAL ITERATION INDICES.

| $p$ | $H(X,Y)$ | 1-st it. | 2-nd it. | 3-rd it. | 4-th it. |
|---|---|---|---|---|---|
| 0.0213 | 1.1487 | 50.00 | 50.00 | 50.00 | 50.00 |
| 0.0204 | 1.1437 | 50.00 | 50.00 | 50.00 | 50.00 |
| 0.0195 | 1.1386 | 50.00 | 46.16 | 46.12 | 46.11 |
| 0.0186 | 1.1335 | 50.00 | 39.51 | 38.45 | 34.58 |
| 0.0177 | 1.1283 | 42.47 | 23.65 | 23.35 | 22.31 |
| 0.0168 | 1.1231 | 34.73 | 16.16 | 16.09 | 15.17 |
| 0.0159 | 1.1178 | 22.97 | 12.59 | 12.58 | 12.57 |

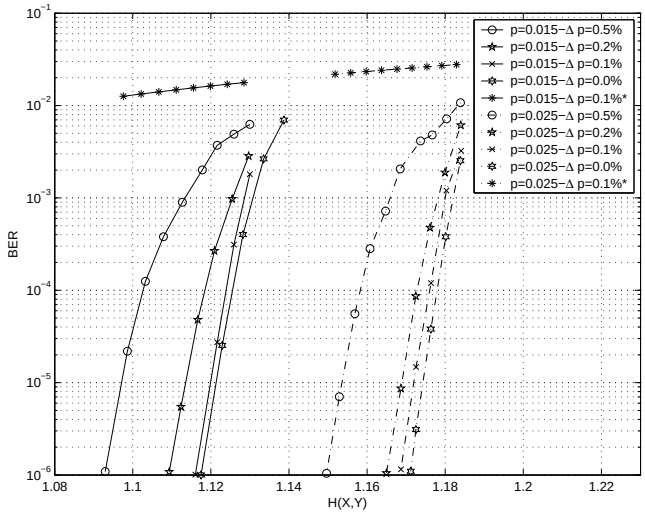



Fig. 10. BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations as a function of the joint entropy between sources $X$ and $Y$, when the stopping criterion for global iterations is applied. Results refer to the LDPCs labelled $L_8$ and $L_5$ in Table I. The legend shows the mean correlation value $p$ and the maximum value of the correlation variation with respect to the mean value. Curves labelled with $*$ refer to the ones obtained without the iterative paradigm, using the mean correlation value.

performed by the joint decoder at the end of a given global iteration, for various values of correlation between the sources and when the stopping criterion on the global iterations is not applied. Finally, we evaluated the average number of global iterations performed by the iterative algorithm when the stopping criterion on global iterations is employed during decoding. Simulation results show that when the LDPC decoder works at BER levels less than or equal to $10^{-5}$, the average number of global iterations equals 1.2, thus guaranteeing a very efficient iterative approach to the co-decompression problem. In other words, an overall average number of 80 LDPC decoding iterations suffices to obtain good BER performance.

In the following set of simulation results, we adopted another approach in order to test the proposed algorithm for varying actual correlation levels. In particular, for any given value of mean correlation $p$, we generate a uniform random variable having mean value equal to the mean correlation itself and with a maximum variation of $\Delta p$ around this mean value. We used the following maximum variations: $\Delta p = 0.5, 0.2, 0.1\%$, and $\Delta p = 0.0\%$ which refers to the case in which the correlation value is not variable, but fixed.

For each data block, we set the actual correlation equal to the mean correlation plus this perturbation. The decoder iterates to estimate the actual correlation value which varies around its mean value from one block to the next. In effect, the parameter $p$ is iteratively estimated as discussed in the previous section. A similar approach has been pursued in [10] for fixed correlation level, whereby an iterative approach is used for the estimation of the correlation between the two correlated sequences, but employing turbo codes.

Fig. 10 shows the BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations and as a function of the joint entropy between sources $X$ and $Y$, when the stopping criterion for global iterations is applied. LDPCs used for encoding are the one labelled $L_5$ and $L_8$ in

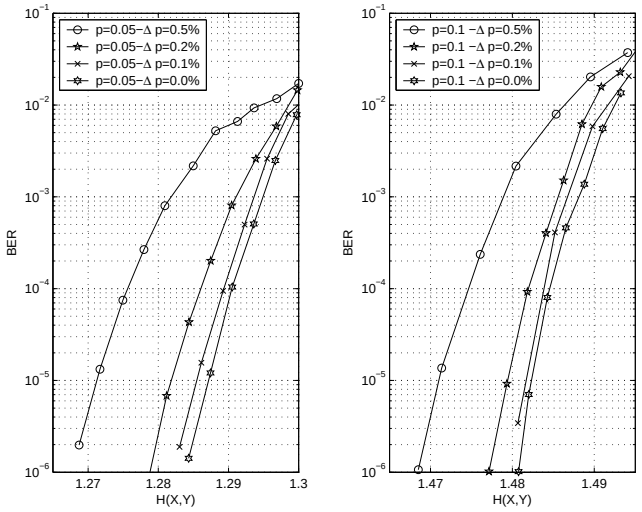



Fig. 11. BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations as a function of the joint entropy between sources $X$ and $Y$, when the stopping criterion for global iterations is applied. Results refer to the LDPCs labelled $L_3$ (left subplot) and $L_2$ (right subplot) in Table I. The legend shows the mean correlation value $p$ and the maximum value of the correlation variation with respect to the mean value.

Table I which guarantee compression rates of $R_X = 0.237$ and $R_X = 0.189$, respectively. LDPC labelled $L_5$ is used at mean values of $p$ equal to $0.025$, while LDPC $L_8$ is adopted for a mean correlation of $0.015$. From Fig. 10 one clearly sees that LDPC decoding does not converge when the decoder does not iterate for estimating the actual value of $p$, but uses only its mean value for setting the extrinsic information. Notice also that the performances of the iterative decoder when the correlation value is fixed (curves labelled $\Delta p = 0.0$ in Fig. 10), are very close to the case in which the actual correlation value varies within $\Delta p = 0.1\%$ from the mean value.

Similar considerations can be deduced from Fig. 11 which shows the BER performance of the proposed iterative decoding algorithm when using LDPCs labelled $L_2$ and $L_3$ in Table I which guarantee compression rates of $R_X = 0.597$ and $R_X = 0.365$, respectively. LDPC labelled $L_2$ is used at mean correlation equal to $0.1$, while LDPC $L_3$ is used with a mean correlation of $0.05$. Note that the performance degrades as $\Delta p$ increases since the encoder works further away from its optimal operating point.

The results on the compression achieved with the proposed algorithm are shown in Table V for the case in which the correlation value is fixed. The first row shows the fixed correlation parameter assumed, namely, $p = P(x_j \neq y_j),\ \forall j = 1, \ldots, k$ in our model. The second row shows the joint entropy limit for various values of the fixed correlation parameter $p$. The third row presents the results on compression achieved with the proposed algorithm employing a maximum of 5 global iterations in conjunction with using the stopping criterion noted in the previous section. As in [16], we assume error free compression for a target Bit Error Rate (BER) $10^{-6}$. Note that statistic of the results shown has been obtained by counting 30 erroneous frames.

From Table V it is evident that significant compression gains

TABLE V
COMPRESSION RATE PERFORMANCE OF THE ITERATIVE ALGORITHM FOR $H(p) = 0.112$.

| $p$ | 0.015 |
|---|---|
| $H(p)+1$ | 1.112 |
| $R = R_X + R_Y$ | 1.189 -$L_8$ |

with respect to the theoretical limits can be achieved as the correlation between sequences $X$ and $Y$ increases.

## VI. CONCLUSION

In this paper we have presented a novel iterative joint decoding algorithm based on LDPC codes for the Slepian-Wolf problem of compression of correlated information sources.

In the considered scenario, two correlated sources transmit toward a common receiver. The first source is compressed by transmitting the parity check bits of a systematic LDPC encoded codeword, without the knowledge of the correlation among the sources. The correlated information of the second source is employed as side information at the receiver and used for decompressing and decoding of the first source. The correlation is not known a-priori, but is estimated through an iterative paradigm. Both the iterative decoding algorithm and the cross-correlation estimation procedure have been described in detail.

Simulation results suggest that relatively large compression gains are achievable at relatively small number of global iterations specially when the sources are highly correlated.

## VII. ACKNOWLEDGEMENTS

This work was partially supported by the PRIN 2007 project *Feasibility study of an optical Earth-satellite quantum communication channel* and by the CAPANINA project (FP6-IST-2003-506745) as part of the EU VI Framework Programme.

## REFERENCES

[1] F. Daneshgaran, M. Laddomada, and M. Mondin, "LDPC-based iterative algorithm for compression of correlated sources at rates approaching the Slepian-Wolf bound," *In Proc. of the Int. Conf. SPACOMM 2009,* pp.74-79, Colmar, France, July 2009.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory,* vol.IT-19, no. 4, pp.471-480, July 1973.

[3] L. J. Guibas, "Sensing, tracking, and reasoning with relations," *IEEE Signal Processing Magazine,* no.3, pp.73-85, March 2002.

[4] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine,* no.3, pp.51-60, March 2002.

[5] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine,* no.3, pp.61-72, March 2002.

[6] A. Aaron and B. Girod, "Compression with side information using turbo-codes," *In Proc. of Data Compress. Conf., DCC'02,* pp.252-261, April 2002.

[7] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," *In Proc. of Global Telecomm. Conf., GLOBECOM'01,* vol.2, pp.1400-1404, Nov. 2001.

[8] J. Bajcsy and P. Mitran, "Coding for the Wyner-Ziv problem with turbo-like codes", *In Proc. of Inter. Symposium on Infor. Theory, ISIT'02,* pp.91, 2002.

[9] J. Bajcsy and I. Deslauriers, "Serial turbo coding for data compression and the Slepian-Wolf problem",*In Proc. of Inform. Theory Workshop,* pp.296-299, March 31-April 4 2003.

[10] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo-codes," *IEEE Communications Letters,* vol.5, no.10, pp.417-419, October 2001.

[11] Y. Zhao and J. Garcia-Frias, "Joint estimation and compression of correlated nonbinary sources using punctured turbo codes," *IEEE Transactions on Communications,* vol.53, no.3, pp.385-390, March 2005.

[12] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "A distributed source coding technique for correlated images using turbo-codes," *IEEE Communications Letters,* vol.6, no.9, pp.379-381, September 2002.

[13] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Joint source-channel coding of binary sources with side information at the decoder using IRA codes," *In Proc. of IEEE 2002 Multimedia Signal Processing Workshop,* December 2002.

[14] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," *In Proc. of IEEE Data Compression Conference,* 2003.

[15] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters,* vol.6, no.10, pp.440-442, October 2002.

[16] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information using low-density parity-check codes," *In Proc. of IEEE GLOBECOM 2002,* vol. 2, pp.1300-1304, 17-21 Nov. 2002.

[17] J. Garcia-Frias, "Joint source-channel decoding of correlated sources over noisy channels," *In Proc. of DCC 2001,* pp.283-292, March 2001.

[18] J. Garcia-Frias, W. Zhong, and Y. Zhao, "Iterative decoding schemes for source and joint source-channel coding of correlated sources," *In Proc. of Asilomar Conference on Signals, Systems, and Computers,* November 2002.

[19] F. Daneshgaran, M. Laddomada, and M. Mondin, "Iterative joint channel decoding of correlated sources employing serially concatenated convolutional codes," *IEEE Transactions on Information Theory,* vol.51, no.7, pp.2721-2731, July 2005.

[20] F. Daneshgaran, M. Laddomada, and M. Mondin, "Iterative joint channel decoding of correlated sources," *IEEE Transactions on Wireless Communications,* Vol. 5, No. 10, pp.2659-2663, October 2006.

[21] F. Daneshgaran, M. Laddomada, and M. Mondin, "LDPC-based channel coding of correlated sources with iterative joint decoding," *IEEE Transactions on Communications,* Vol. 54, No. 4, pp.577-582, April 2006.

[22] J. Barros and S.D. Servetto, "The sensor reachback problem," *Submitted to IEEE Transactions on Information Theory*, available online at http://cn.ece.cornell.edu/publications/papers/20031112, November 2003.

[23] X. Zhu, L. Zhang, and Y. Liu, " A distributed joint source-channel coding scheme for multiple correlated sources," *Fourth International Conference on Digital Communications and Networking in China,* 2009, pp. 1 - 6.

[24] W. Yunnan, V. Stankovic, Z. Xiong, and S. Kung, " On practical design for joint distributed source and network coding," *IEEE Transactions on Information Theory,* Vol. 55, No. 4, pp. 1709 - 1720, April 2009.

[25] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. on Inform. Theory,* vol.42, no.2, pp.429-445, March 1996.

[26] T.J. Richardson and R.L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol.47, no.2, pp.638 - 656, Feb. 2001.

[27] G. Montorsi and S. Benedetto, "Design of fixed-point iterative decoders for concatenated codes with interleavers," *IEEE Journal on Selected Areas in Communications,* vol.19, No. 5, pp.871-882, May 2001.

[28] Xiao-Yu Hu, E. Eleftheriou, and D.M. Arnold, "Progressive edge-growth Tanner graphs," *IEEE Global Telecommunications Conference*, vol.2, pp.995-1001, 25-29 Nov. 2001.

[29] Xiao-Yu Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," *IEEE Global Telecommunications Conference*, vol.2, pp.25-29, 25-29 Nov. 2001.

# Scalable and Robust Wireless JPEG 2000 Images and Video Transmission with Adaptive Bandwidth Estimation

Max Agueh, Stefan Ataman
and Cristina Mairal
LACSC – ECE
37, Quai de Grenelle
75015 Paris, France
{agueh, ataman, mairal}@ece.fr

Henoc Soude
LIASD – Université Paris 8
2, Rue de la liberté
93526 Saint Denis, France
soude@ai.univ-paris8.fr

***Abstract*— This paper presents a scalable and dynamic scheme for robust JPEG 2000 images/video transmission over wireless channels. The proposed system relies on an adaptive bandwidth estimation tool to select the suitable JPEG 2000 streams, layers and resolution. A Wireless JPEG 2000 (JPWL) compliant Forward Error Correction (FEC) scheme is introduced. An optimal layer oriented Unequal Error Protection Forward Error Correction rate allocation scheme is proposed to ensure codestreams protection against transmission errors. The main advantages of the proposed scheme are its optimality, its compliance to Wireless JPEG 2000 (the 11th part of JPEG 2000 standard) and its low time consumption. We demonstrate that our proposed scheme outperforms the layer oriented FEC scheme proposed by Guo *et al.* and other existing layer based FEC schemes. We also show that, due to its low run time, our layer based scheme is a good candidate for highly time constrained motion JPEG 2000 video streaming applications, thus, in this sense, its overcomes the limitation of optimal packet oriented FEC rate allocation scheme. We then validate the effectiveness of our proposed layer oriented scheme with a Wireless Motion JPEG 2000 client/server application.**

***Keywords: layer-oriented FEC; unequal error protection; layer scalability; wireless JPEG 2000; video streaming***

## I. INTRODUCTION

In high error rate environments such as wireless channels, data protection is mandatory for efficient transmission of images and video. In this context, JPEG 2000, the newest image representation standard [1] proposes in its 11th part (Wireless JPEG 2000 – JPWL) [2] different techniques such as data interleaving, FEC with Reed-Solomon (RS) codes etc. in order to enhance the protection of JPEG 2000 codestreams against transmission errors.

Since wireless channels' characteristics depend on the transmission environment, the packet loss rate in the system also changes dynamically. Thus a priori FEC rate allocation schemes such as the one proposed in [3] are less efficient.

Moreover, in wireless multimedia systems such as the one considered in this paper (see Figure 1), a straightforward FEC methodology is used, by applying FEC uniformly over the entire stream (Equal Error Correction - EEP). However in [4], Gupa et al. suggest that for hierarchical codes, such as JPEG 2000, Unequal Error Protection (UEP) which assigns different FEC to different portion of codestream has been considered as a suitable protection scheme.

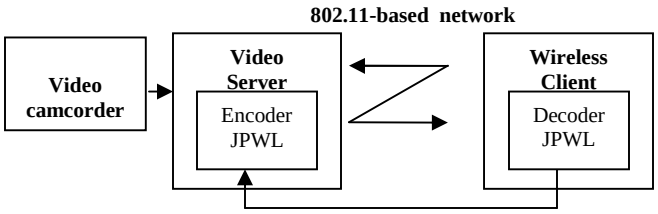



Figure 1. Wireless video streaming system

Two families of data protection schemes address this issue by taking the wireless channel characteristics into account in order to dynamically assign the FEC rate for JPEG 2000 based images/video. The first family is based on a dynamic layer-oriented unequal error protection methodology whereas the second relies on a dynamic packet-oriented unequal error protection methodology. Hence, in the first case, powerful RS codes are assigned to the most important layers and less robust codes are used for the protection of less important layers. It is worth noting that in this case, all the JPEG 2000 packets belonging to the same layer are protected with the same selected RS code. Examples of layer-oriented FEC rate allocation schemes are available in [4] and [5]. On the other side, in packet-oriented FEC rate allocation schemes such as the one presented in [6], RS codes are assigned in descending order of packets importance. In [6], we demonstrate that the proposed optimal packet-oriented FEC rate allocation is more efficient than the layer-oriented FEC rate allocation scheme presented in [4] and [5]. However, layer-based FEC rate allocation schemes have low complexity while packet-oriented FEC allocation methodologies are complex especially when the number of packets in the codestream is high. In this case, packet oriented FEC schemes are unpractical for highly time-constrained images/video streaming applications. Therefore, switching to a layer oriented FEC rate allocation scheme is more interesting. The smart FEC rate allocation scheme proposed in [7] addresses this issue by allowing switching from a packet oriented FEC scheme to a layer oriented scheme such as the ones proposed in [4] and [5]. However, to our knowledge, existing layer oriented FEC rate allocation schemes are based on heuristics and thus are suboptimal because they rely on parameters which are empirically fixed. For example, the layer oriented FEC scheme proposed in [4] relies on the permissible error rate ($\theta$) whereas the layer oriented FEC scheme proposed in [5] relies on a Quality of Service (QoS) metric ($\varepsilon_0$).

In this paper, based on the optimal packet oriented FEC methodology presented in [6], we propose an optimal layer oriented FEC rate allocation scheme for robust JPEG 2000 codestreams transmission over wireless channels.

The paper is organized as follows. In Section II, an overview of the video streaming system and a presentation of layer based FEC rate allocation schemes are provided. In Section III and IV, we describe the optimal layer oriented FEC scheme and the available bandwidth estimation scheme. We show the performances achieved by our proposed layer based FEC scheme in Section V. Finally, some conclusions are drawn in Section VI.

## II. Overview Of The Streaming System and of The JPEG 2000 Based FEC Rate Allocation Schemes

This section is dedicated to the description of the considered wireless JPEG 2000 video streaming system and to the presentation of existing FEC rate allocation schemes.

### A. *The Wireless JPEG 2000 video streaming system and the wireless channel*

Unlike the system described in [6], where the FEC rate allocation scheme is packet oriented, in the current paper we consider a layer oriented FEC rate allocation scheme. In other words the difference between both systems is the FEC rate allocation module. In the packet oriented scheme the redundancy is added by taking the packets importance into account (see Figure 2) while in our layer oriented scheme, we rely on layer importance in order to allocate the adequate RS codes (see Figure 3).

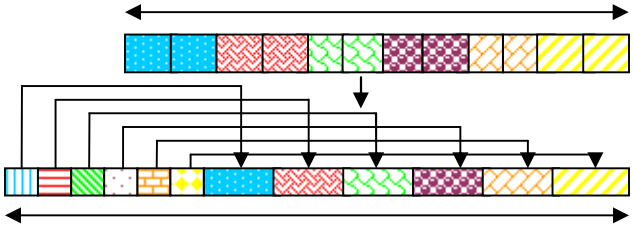
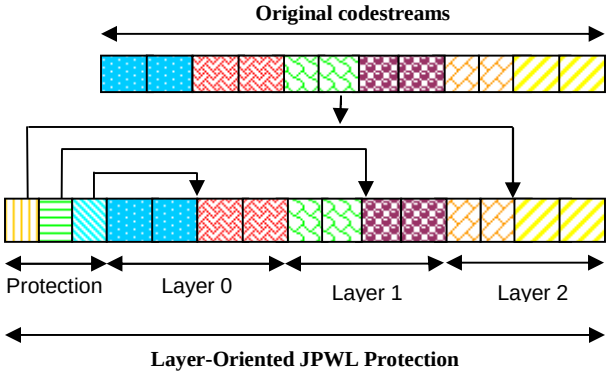


Figure 2. A JPEG 2000 codestreams transmission through the JPWL packet-oriented FEC rate system




Figue 3. A JPEG 2000 codestreams transmission through the JPWL layer-oriented FEC rate system

The wireless channel is emulated by real wireless channel traces available in [13]. At the encoder side, the Gilbert model [14] is used to derive application level models of error occurrences in the considered traces. A detailed description of the platform used to generate the loss patterns along with an analysis on wireless channel modeling with Gilbert model is provided in [6].

### B. *Overview of JPEG 2000 based FEC rate allocation schemes*

In this section we present an overview of JPEG 2000 based FEC schemes.

#### 1) *Packet-oriented FEC rate allocation scheme*

In this FEC rate allocation scheme the correct decoding of packet $i$ at the receiver yields a reduction of the distortion on the transmitted image. Since $RD_{i,\gamma}$ is considered as the reduction of distortion achieved when packet $i$ is protected at the level $\gamma$, the gain is the ratio between the image quality improvement $RD_{i,\gamma}$ and the associated cost in terms of bandwidth consumption.

For each JPEG 2000 image, the optimal packet-oriented algorithm models the channel with a Gilbert model and for each possible protection level $\gamma$, its evaluates the probability of incorrect word decoding. Then, for each packet $i$, based on the estimated probability of decoding word, the reduction of distortion associated with the decoding of packet $i$, protected from level $\gamma$ to $\gamma_{\max}$ is estimated. The gain yield by the associated increment of quality is also computed. After ordering those gain values by decreasing order of importance, the optimal protection rate for each packet is selected up to meet the rate constraint.

#### 2) *Layer-oriented FEC rate allocation scheme*

In layer-oriented FEC rate allocation schemes ([4] and [5]), unequal error protection is applied by taking the importance of each layer into account. Those families of FEC rate allocation schemes can be divided in two groups: non JPWL compliant schemes and JPWL compliant schemes.

*a) Non JPWL compliant Layer-oriented FEC rate allocation scheme*

In non JPWL compliant layer-oriented FEC rate allocation schemes such as the one presented in [4], the slope of the Rate-Distortion curve is used to select a code for each quality layer. The slope $S_j$ corresponding to layer $j$ is expressed as: $S_j = \Delta D_j / \Delta R_j$ where, $S_j$ represents the contribution of layer $j$ to the improvement of image quality, $\Delta R_j$ is its length (in byte) and $\Delta D_j$ corresponds to the distortion decrement measured by MSE (Mean Square Error).

Hence powerful non JPWL compliant RS codes are set to the most important layers such as the base layer and the other layers are protected by decreasing order of importance. However, this algorithm is not JPWL compliant and was designed based on the assumption that the channel is a memoryless Binary Symmetric Channel (uncorrelated error occurrence) which is not realistic because wireless channels have correlated errors sequences. Moreover, the algorithm proposed by Z. Guo *et al* was not adaptive. In this work, we overcome this limitation by adapting the Z. Guo algorithm for a dynamic selection of RS codes thanks to a dynamic Bit Error Rate (BER) estimation scheme.

*b) JPWL compliant dynamic Layer-oriented FEC rate allocation scheme*

JPWL compliant FEC rate allocation schemes, such as the one presented in [5], are based on the assumption that transmitted JPEG 2000 image quality is linked to the amount of correctly decoded packets at the receiver. Hence, goal of this scheme is to maximize the overall throughput in the system under a Quality of Service (QoS) fixed parameter ( $\varepsilon_0$ ). The dynamic layer-oriented FEC rate allocation scheme improves the performance by about 10% compared to a priori selection of channel coding.

The layer oriented FEC rate allocation schemes presented in this section are characterized by heuristics whose parameters are set in order to achieve a desired Quality of Service. Hence, high QoS level means more data protection and thus more bandwidth consumption. Since wireless client/server systems are bandwidth constrained, the desired high QoS level may not be achieved because of the risk of exceeding the available bandwidth. In this sense, traditional layer based FEC rate schemes may be viewed as suboptimal as they achieve a less effective codestreams protection. The proposed optimal layer oriented FEC rate allocation scheme overcomes these limitations by selecting the best RS codes yielding by this way the maximum achievable images/video quality regarding the available resources in the system.

## III. OPTIMAL LAYER ORIENTED FORWARD ERROR CORRECTION RATE ALLOCATION

### A. Layer Based FEC rate allocation problem formalization

Considering that JPEG 2000 codestreams are constituted of a set of $L$ layers, the optimal FEC allocation problem can be resumed by answering the question: how to optimally protect each layer in order to minimize the transmitted image distortion under a rate constraint determined by the available bandwidth in the system?

Let $B_{av}$ be the budget constraint in bytes corresponding to the available bandwidth in the system. Let $lay_i$ be the length in bytes of the $i^{th}$ layer of the $L$ layers available in the codestream and $RS(n,k)$ the Reed-Solomon code used for its protection. The corresponding protection level is $\gamma$ and the FEC coding rate is $R = k/n$ . We define $fec = 1/R = n/k$ as the inverse of the channel coding rate, so $(lay_i) \times fec$ represents, in bytes, the increase of the $i^{th}$ layer length when protected at a level $\gamma$ .

Unlike the packet oriented FEC scheme, where all the 16 default RS codes, defined by JPWL standard, are considered in the FEC rate allocation process, here we restrict the considered RS codes to those with $fec \le 2$ . In other words we consider only the first 10 default codes. This assumption makes sense in the context of layer oriented FEC rate allocation, because adding redundant data which in ratio is more than twice superior to the original layers may overload the networks and drastically increase the losses instead of reducing them.

Let $\gamma$ be a layer protection level selected in the range $0 \le \gamma \le \gamma_{\max}^{lay}$ , each protection level corresponding to a specific RS code selected between the 10 JPWL default RS codes ( $\gamma = 0$ means that the layer is not transmitted, $\gamma = 1$ means transmission with protection level 1, higher values imply increasing channel code capacity with $\gamma$ and $\gamma_{\max}^{lay} = 10$ ).

Let $\beta_i$ be the number of data packet constituting the $i^{th}$ quality layer of a JPEG 2000 codestream, $RD_{lay_i}^0$ and $RD_{lay_i}^{\gamma}$ be respectively the reduction of distortion associated to the correct decoding of layer $i$ and the reduction of distortion associated to the correct decoding of layer $i$ protected to level $\gamma$ . The reduction of distortion metric associated to the correct decoding of the packets of a JPEG 2000 codestream is extracted from a codestream index file. The reduction of distortion metric is further presented in [6]. We rely on this codestream index file to derive $RD_{lay_i}^0$ and we associated the decoding error probability estimation process presented in [14] in order to derive $RD_{lay_i}^{\gamma}$ . Hence, the layer oriented FEC rate allocation problem is formalised by:

$$\text{Maximize} \quad \sum_i^L \frac{RD_{lay_i}^{\gamma}}{(lay_i) \times fec_i} \tag{3-1}$$

$$\text{Subject to} \quad \sum_i^L (lay_i) \times fec_i \le B_{av} \tag{3-2}$$

We addressed this problem in [18] by proposing an optimal layer oriented FEC rate allocation scheme. In the following we present the proposed algorithm.

*B. Optimization*

We define $G_{lay_i}^{\gamma}$ as the gain in quality of the transmitted image obtained at the receiver side when layer $i$ is decoded.

We derive $RD_{lay_i}^{1}$ and $RD_{lay_i}^{\gamma}$ the reduction of distortion obtained when layer $i$ is transmitted respectively with protection level 1 and with protection level $\gamma$ .We have:

$$RD_{lay_i}^{1} = (1 - P_{lay_i}^{1}) \times RD_{lay_i}^{0}$$
$$RD_{lay_i}^{\gamma} = (1 - P_{lay_i}^{\gamma}) \times RD_{lay_i}^{0} \quad (3\text{-}3)$$

where $P_{lay_i}^{1}$ and $P_{lay_i}^{\gamma}$ are the decoding error probabilities obtained when layer $i$ is protected respectively to level 1 and to level $\gamma$ . The final gain is:

$$G_{lay_i}^{1} = \frac{RD_{lay_i}^{1}}{lay_i} = \frac{(1 - P_{lay_i}^{1}) \times RD_{lay_i}^{0}}{lay_i} \quad (3\text{-}4)$$

Similarly, any transmission between two consecutive protection levels ( $\gamma - 1$ and $\gamma$ ) yields an improvement in terms of reduction of distortion but has a budget cost equal to $(fec_{\gamma} - fec_{\gamma-1}) \times lay_i$ , hence we have:

$$G_{lay_i}^{1} = \frac{RD_{lay_i}^{\gamma} - RD_{lay_i}^{\gamma-1}}{(fec_{\gamma} - fec_{\gamma-1}) lay_i}$$
$$G_{lay_i}^{\gamma} = \frac{(P_{lay_i}^{\gamma-1} - P_{lay_i}^{\gamma}) RD_{lay_i}^{0}}{(fec_{\gamma} - fec_{\gamma-1}) \times lay_i} \quad (3\text{-}5)$$

For each layer, a set of gain values is computed, ordered in decreasing order of importance and stored in a vector. Then, the FEC rate associated to the first gain value of each vector is applied for the corresponding layer's protection without exceeding the available bandwidth. It is worth noting that all the packets belonging to the same layer are protected at the same FEC rate.

*C. Contribution of the proposed optimal layer oriented FEC rate allocation scheme*

Even if the gain metrics presented in the previous section seem close to the ones used in [6], they hold a fundamental difference because they rely on the contribution of each layer to the reduction of distortion instead of just taking into account the contribution of a specific packet. Actually, during the source coding process, the incremental contribution from the set of image codeblocks is collected in quality layers. Due to the fact that the rate-distortion compromises derived during JPEG 2000 truncation process are the same for all the codeblocks, for any quality layer index $i$ the contributions of quality layer 1 through quality layer $i$ constitute a rate-distortion optimal representation of the entire image. Hence, at layer level the reduction of distortion values are strictly decreasing. In contrast, the selection of a specific JPEG 2000 packet does not guarantee that the contributions of packet 1 to the selected index packet are monolithically decreasing. In this case, as confirmed by A. Descampe *et al* [15], some additional restrictions have to be added to the considered convex-hull in order to ensure rate-distortion and cost-distortion optimality. This justifies the necessity to have a merging step in the packet oriented FEC scheme [6] (it ensures that the convex-hull is always convex). In the layer oriented FEC this step is skipped because the reduction of distortion curve is already monolithically decreasing, significantly reducing the complexity and thus the time-consumption of the FEC rate allocation algorithm. Moreover, in the proposed optimal layer oriented FEC scheme, we only consider the first 10 RS codes instead of considering all the 16 default RS codes defined by JPWL standard as it is the case in [6]. This also reduces considerably the FEC scheme time consumption as its leads to less gains values computation which make the proposed optimal layer FEC rate allocation scheme a good candidate for real time images/video streaming applications.

In addition, the number of layers available in the codestreams is another criterion which contributes to the reduction of the time consumption of our proposed FEC scheme. Actually, a JPEG 2000 image extracted from a Motion JPEG 2000 video sequence is defined by ( $L$ , $R$ , $C$ ) where $L$ is the number of quality layers of the considered image , $R$ is its resolution level corresponding to the decomposition levels of the Discrete Wavelet Transform and $C$ is the number of components. Assuming that the considered JPEG 2000 image is not spatially divided and thus is described by a unique tile, the number of data packets available in the considered JPEG 2000 codestreams is given by $S = L \times R \times C$ . In this context, the complexity of packet oriented FEC schemes [6] is based on the $S$ data packets while the complexity of the optimal layer based FEC is based on the $L$ layers available in the codestreams. In scalable JPEG 2000 images, since the number of layers is significantly lower in comparison to the number of data packets, the time consumption of our proposed layer oriented FEC scheme is significantly low in comparison to packet oriented scheme.

*Algorithm:*

For each JPEG 2000 image

- Model the channel with a Gilbert model and for each possible protection level $\gamma$ ( $0 \le \gamma \le 10$ ), evaluate the probability of incorrect word decoding $P_{lay_i}^{\gamma}$
- For $i = 1$ to $i = L$ (Number of JPEG 2000 layers)

  For $\gamma = 1$ to $\gamma = 10$

  Estimate $RD_{lay_i}^{\gamma} = (1 - P_{lay_i}^{i,\gamma}) \times RD_{lay_i}^{0}$

  $$G_{lay_i}^{1} == \frac{RD_{lay_i}^{\gamma} - RD_{lay_i}^{\gamma-1}}{(fec_{\gamma} - fec_{\gamma-1}) lay_i}$$

  End For

End For

- Order gain values in decreasing order of importance
- Select each gain value, corresponding to a specific protection level, up to meeting the rate constraint
- Optimally protect JPEG 2000 layers with the corresponding RS codes

End For

## IV. BANDWIDTH ESTIMATION AND IMAGES/VIDEO SCALABILITY SCHEME

### A. Available Bandwidth estimation

In the literature many authors investigate classifications of the bandwidth estimation tools. In [7], R. Prasad et al. define four types of bandwidth estimation tools. The first one, the Variable Packet Size technique (VPST) measures the capacity of individual hops. It uses ICMP (Internet Control Message Protocol) packets to measure the RTT (Roundtrip Time). It assumes that the minimum RTT means no queuing delays. Hence queuing delays are not taken into account to estimate capacity. The result is a straight line, whose slope equals 1/C. The second bandwidth estimation tool presented in [7] is the Probing Packet Pair Dispersion Technique (PPPDT) which measures the end-to-end capacity using a packet pair with the same length and rate. The capacity is found using the formula:

$$C = \frac{L}{\Delta_{out}}$$

where $\Delta_{out}$ is the dispersion of the packet pair at the receiver.

The third bandwidth estimation tool is the Self-loading Periodic Streams Technique (SLoPST) which measures the available bandwidth sending packet with the same length but at different rates and calculating its one way delay. The moment the delay starts to increase means the available bandwidth has been exceeded.

The last bandwidth estimation tool is the Probing Packet Pair Trains Dispersion Technique (PPPTDT) which measures the available bandwidth and the capacity of an end-to-end path. It sends a train of packet pairs at gradually increasing packet rate and estimates its average dispersion rate at the receiver. The result is a curve where the slope is 1/C and the inflexion point is the available bandwidth value.

In our work, we focus on a fast bandwidth estimation technique because it is more effective in tracking fast varying wireless channels like the ones considered in our scenarios.

### B. Available Bandwidth estimation tool

The available bandwidth estimation tool which is implemented in our system is called WBest (Bandwidth Estimation Tool for Wireless Networks) and is presented in [8] by M. Li and C. Chang. It is a two step fast converging and accurate bandwidth estimation tool. It uses the effective capacity ($C_e$) to estimate the available bandwidth. It takes also into account cross-traffic impact. First, the effective capacity is measured using an improved Probing Packet Pair Technique. Actually, in this technique, the median of the dispersion of the train of packet pairs sent is calculated. Then, another packet train is sent at the rate of the effective capacity in order to evaluate the real available bandwidth. This procedure avoids the additional delay which is yielded by the incremental packet rate generated while seeking for the rate which congests the path.

In this context, the available bandwidth ($AB$) is derived as follows:

$$AB = C_e(2 - \frac{C_e}{\Delta_{out}}) \text{ if there is no congestion}$$

$$\text{or } AB = 0 \text{ if congestion is detected}$$

It is worth noting that during the second step, congestion is detected by analyzing the dispersion of the packet train. If the dispersion is lower than $C_e / 2$ , it means that congestion occurred. In this case, the bandwidth estimation process is cancelled because packets are queuing.

The authors of this work pointed out that finding the optimal lengths of the trains used in both stages is a difficult issue, and in [8], they propose a methodology to address this issue. Relying in the methodology proposed in [8], we empirically derive that in our scenarios, 6 packets pairs for the first train and 30 packets for the second train is a good tradeoff which yields sufficiently accurate bandwidth estimation results.

Once the available bandwidth estimated in our system, the following step consists in adapting the images and video streams to the channel conditions.

In the present work we implement an adaptive bandwidth estimation tool and propose an additional scalability tool at the encoder which dynamically and efficiently selects the best resolution and layer for each JPEG 2000 frame before transmission through the wireless channel. Hence, according to the estimated bandwidth, refinement layers could be added or removed from JPEG 2000 codestreams. We present in the following the processes which are implemented at the encoder.

*Algorithm*:

Once connected, the server starts the WBest [8] process in order to obtain the initial available bandwidth. At this step the goal is to send images and video with maximum detail (highest resolution and all refinement layers) matching with the estimated bandwidth. The original resolution and number of layers for the considered video is found using an indexer like the one available in [9].

In our work, the default number of resolution is 6, the length and the width of the image must be a power of 2 (here 352x288) and the number of layers is 3.

Let $l$ be a layer of a JPEG 2000 image and $SE_{rate}^{l}$ is corresponding source encoding rate. Let $fec_{rate}^{l} = \frac{n}{k}$ be the inverse of the Reed-Solomon code RS(n,k) selected by the layer oriented FEC rate allocation scheme to protect layer $l$ against transmission errors. Let $frame\_length$ be the amount of data needed to transmit layer $l$ protected, we have:

$$frame\_length = H \times W \times SE_{rate}^{l} \times fec_{rate}^{l} \qquad (4\text{-}1)$$

Since quality is the most important parameter for a JPEG 2000 transmission system, in our algorithm, for a given resolution we try to transmit a maximum number of refinement layers.

The proposed scheme is able to adapt to channel variations thanks to the bandwidth estimation tool. Hence, when the channel experiences good conditions, our heuristic algorithm selects the highest resolution with the highest quality (all the refinement layers are transmitted). If the channel experiences harsh conditions, image layers and resolution are decreased up to defined thresholds. We empirically derive thresholds ($l_{\max}/2$) for layer removing and for ($resol_{\max}/l_{\max}$) resolution reduction because we notice that images quality starts to show a huge degradation when we remove more than half of the original image layers and image visualization becomes impossible under this resolution reduction threshold. Hence, when the channel experienced bad conditions, image layers are incrementally reduced while maintaining original resolution of the JPEG 2000 frame to highest level. However, if the corresponding frame length do not match the available bandwidth, image resolution downscaling is done. It is worth noting that our fixed thresholds are valid for our scenarios and may change in different environments.

Once the resolution and the number of layers are chosen, the server sends the video streaming to the client.

The available bandwidth estimation tool is launched every 10 frames but this frequency could be changed according to the application requirements. An interesting extension to this work could be to optimally adapt the frequency of the bandwidth estimation tool to the channel conditions.

The efficiency of the proposed heuristic algorithm is demonstrated using a wireless client/server video streaming application. In the following section, we present the results derived from different video streaming scenarios.

### *C. Results of Available Bandwidth estimation*

The video streaming scenarios considered in this work are derived from wireless transmission trials used in the literature for bandwidth estimation purpose.

The video sequence is speedway.mj2 [16] which is a 352x288 motion JPEG2000 sequence constituted of 200 JPEG 2000 frames with six resolutions and three layers each.

#### *1) Scenario 1*

In the first scenario, the wireless channel considered is derived from BART tool [10], which estimates the available bandwidth in an end-to-end path where the bottleneck is a wireless hop. It uses the Probing Packet Pair Trains Dispersion Technique and improves the system using Kalman filters to measure and track the changes.

In this scenario, we focus on the fast varying part of the estimated bandwidth. Moreover, we divide the bandwidth estimated by BART tool [10] by a parameter δ=2 in order to show that our scheme is efficient even when the wireless channel experienced harsh conditions.

---

Heuristic

1-Wait for client connection request

2-Estimate available bandwidth ($B_{av}$) using WBest tool

3-Derive original images/video maximum number of layers ($l_{\max}$) and maximum resolution ($resol_{\max}$) from the indexer

For each JPEG 2000 image:

4-Initialize current layer parameters ($cur\_lay = l_{\max}$) and current resolution ($cur\_resol = resol_{\max}$)

5-Calculate the needed bandwidth $B_{needed}$

$$B_{needed} = \sum_{i=1}^{cur\_lay} cur\_resol \times SE_{rate}^{i} \times fec_{rate}^{i}$$

6- While ($cur\_lay > l_{\max}/2$)

{

If ($B_{av} \geq B_{needed}$) Send Image

Else

{

$cur\_lay = cur\_lay - 1$

Calculate $B_{needed} = \sum_{i=1}^{cur\_lay} cur\_resol \times SE_{rate}^{i} \times fec_{rate}^{i}$

}

}

Else

{

While ($cur\_resol > \frac{resol_{\max}}{l_{\max}}$)

{

$cur\_resol = \frac{cur\_resol}{2}$

Calculate $B_{needed} = \sum_{i=1}^{cur\_lay} cur\_resol \times SE_{rate}^{i} \times fec_{rate}^{i}$

If ($B_{av} \geq B_{needed}$) Send Image

Else

{ $cur\_resol = \frac{cur\_resol}{2}$

$cur\_lay = cur\_lay + 1$

}

}

}

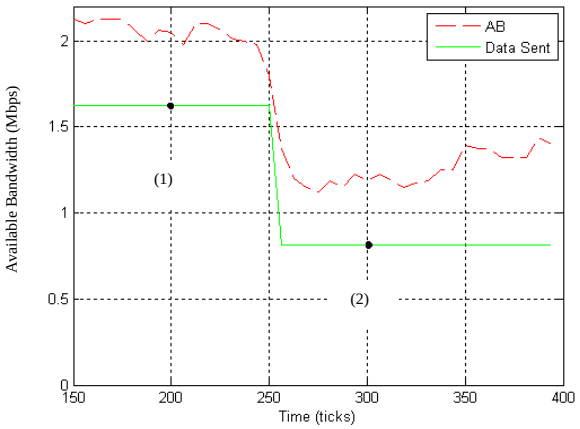



Figure 4.a Available bandwidth versus time – BART Tool scenario

Speedway10.j2k

Speedway25.j2k

Figure 4.b Images processed at point (1) and (2) - BART Tool scenario

| Scenario 1 | Image Length | Image Width | Layer of transmitted Images |
|---|---|---|---|
| (1) | 352 | 288 | 3 |
| (2) | 352 | 288 | 2 |

Table 1: Scalability parameters - BART Tool scenario

In figure 4.a, point (1) indicates that the estimated bandwidth is higher than the needed bandwidth, hence initial JPEG 2000 frames are transmitted. Point (2) shows that the estimated bandwidth is decreased and becomes insufficient to send original images. Hence, the algorithm maintains the resolution at the highest level (initial value) but one layer is removed from original frames as shown in Table 1.

In Figure 4.b we randomly select and present images speedway10.j2k and speedway25.j2k which have been processed respectively on points (1) and (2).

*2) Scenario 2*

The second scenario is derived from the work of Gupta et al. [11]. In their work, their compare a passive technique that uses packets containing useful data as probing packets and the pathchirp tool which is a general bandwidth estimation tool designed using Self-loading Periodic Streams (SLoPS). They demonstrated that the passive tool follows more accurately the changes of the available bandwidth. Moreover, they show that saturating the network to estimate the bandwidth is not the best choice.

In this scenario, the network is a single link between two computers with two additional cross-traffics. The estimated bandwidth has been divided by $\delta$=7 in our tests.

In figure 5.a we observe three points corresponding to different available bandwidth along with adapted data rate. Point (1) indicates that the available bandwidth is significantly low compared to the needed bandwidth. In this case resolution is downscaled without removing a quality layer. Then point (2) shows that when the available bandwidth is increased, resolution is up-scaled while removing a quality layer. Finally, original frames are transmitted at point (3).

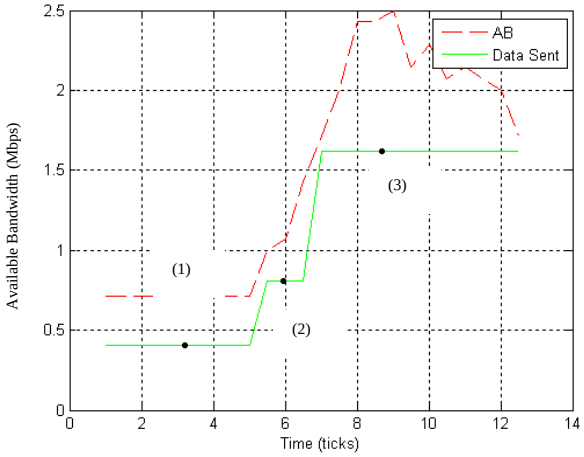



Figure 5.a Available bandwidth versus time – Pathchirp tool scenario

Speedway5.j2k

Speedway13.j2k

Speedway18.j2k

Figure 5.b Images processed at point (1), (2) and (3) - Pathchirp tool scenario

| Scenario Pathchirp tool | Image Length | Image Width | Layer of transmitted Images |
|---|---|---|---|
| (1) | 176 | 144 | 0 |
| (2) | 352 | 288 | 1 |
| (3) | 352 | 288 | 0 |

Table 2: Scalability parameters - Pathchirp tool scenario

In the following, instead of using pathchirp tool for bandwidth estimation, we use a passive tool which is a more accurate tool [11]. Results are presented in Figure 6.a. For the same scenario, we notice that one additional bandwidth state is detected in point (4) which demonstrates that the passive tool is more accurate than pathchirp tool

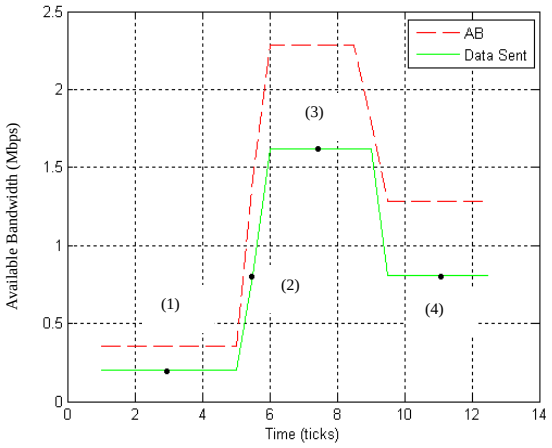



Figure 6.a Available bandwidth versus time – Passive tool scenario

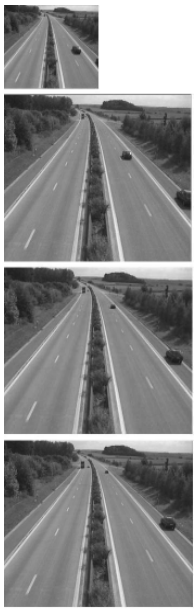



Figure 6.b. Images processed at point (1), (2), (3) and (4) - Passive tool scenario

In comparison to the pathchirp tool, using the passive tool yields image quality layer downscaling at point (4) instead of maintaining the transmission of unmodified original images (see Table 3). Hence, using pathchirp tool may lead to networks overloading.

| Scenario Passive tool | Image Length | Image Width | Layer of transmitted Images |
|---|---|---|---|
| (1) | 176 | 144 | 1 |
| (2) | 352 | 288 | 1 |
| (3) | 352 | 288 | 0 |
| (4) | 352 | 288 | 1 |

Table 3: Scalability parameters - Passive tool scenario

Figure 6.b presents speedway5.j2k, speedway10.j2k, speedway14.j2k and speedway20.j2k which have been processed respectively on points (1), (2), (3) and (4).

## V. PERFORMANCES OF THE OPTIMAL LAYER ORIENTED FEC RATE ALLOCATION SCHEME

### *A. Performance of layer based FEC scheme in terms of time consumption*

In Figure 7 the run time of the proposed layer based FEC rate allocation scheme is plotted versus the number of data packets available in the JPEG 2000 codestreams. This curve is compared to the one achieved using the optimal packet oriented FEC rate allocation scheme [6]. These results are achieved using an Intel core Duo CPU 2.9 GHz workstation.

As packet-oriented and layer oriented schemes are linked by the number of layers available in each image, we vary this parameter in order to derive some comparable results. In the considered scenario, the number of available resolution and component of JPEG 2000 frames are fixed (resolution = 10 and component = 1) because these parameters do not impact the time-performance of layer oriented FEC rate allocation schemes. In Figure 7 each packet (i) corresponds to a specific JPEG 2000 frame (with a specific quality layer).

In this scenario, the available bandwidth in the system is set to 18 Mbit/s ($B_{av} = 18\,Mbits/s$). It is worth noting that in practice few existing JPEG 2000 codecs allow high quality scalability and to our knowledge, none of them can handle more than 50 quality layers. Hence, the considered scenario allows generalization to future high quality layer scalable FEC rate allocation systems.

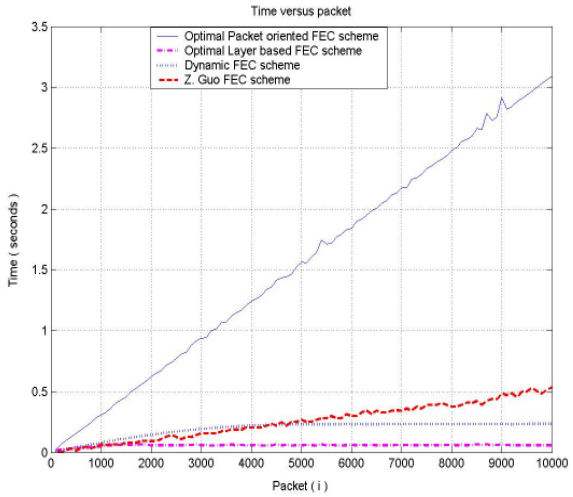



Figure 7. Time versus packets: Fixed image resolution (R=10) –Varying quality layers (0 to 100) – One component (C =1)

In Figure 7 we notice that both layer and packet oriented scheme have a run time linearly increasing with the number of packets available in the codestreams. However, the optimal layer based FEC scheme is significantly less time consuming than the packet based FEC scheme. For codestreams containing less than 1000 packets (quality layers $\leq$ 10) the packet oriented FEC scheme is 3 times more time consuming than our optimal layer based FEC scheme. For JPEG 2000 codestreams, whose number of packets is between 1000 and 5000 (quality layers between 10 and 50) the packet oriented scheme is up to 5 times the run time of the layer based FEC scheme. Since existing JPEG 2000 codecs handle less than 50 quality layers, our proposed optimal layer based scheme is a good candidate for real-time JPEG 2000 codestreams over wireless channel as its yields low time consumption. For codestreams with a number of packets between 5000 and 10000 (quality layers between 50 and 100 – high layer scalability) the packet oriented scheme has 6 times the run time of the layer based FEC scheme. Hence, our proposed layer based scheme, due to its low time consumption, could be viewed as a good candidate for future high quality layer scalable wireless JPEG 2000 based images and video streaming applications.

Although our proposed scheme achieves good performances in terms of time consumption in comparison to packet oriented FEC rate allocation schemes, the last ones present better performance in terms of visualization quality especially for highly noisy channels. It is worth noting that packet oriented and layer oriented FEC schemes advantages could be combined in a smart switching FEC rate allocation scheme such as the one proposed in [12].

In the following section we demonstrate the effectiveness of our proposed layer based FEC scheme thank to a client/server application of Motion JPEG 2000 video streaming over real ad-hoc network traces.

### *B. Layer oriented FEC rate allocation for Motion JPEG 2000 video streaming over real ad-hoc network traces*

In this section we present the results achieved while streaming Motion JPEG 2000 based video over real ad-hoc network channel traces [13] and we demonstrate that the proposed optimal layer based scheme outperforms existing layer oriented FEC schemes even if for highly noisy channels it is less efficient than packet oriented FEC scheme. The comparison is handled both in terms of Structural Similarity (SSIM) [17] and in terms of successful decoding rate. We derive the Mean SSIM metric of the Motion JPEG 2000 video sequence by averaging the SSIM metrics of the JPEG 2000 images contained on the considered video sequence. It is worth noting that each SSIM measure derived is associated to a successful decoding rate metric which corresponds to decoder crash avoidance on the basis of 1000 transmission trials.

The considered wireless channel traces are available in [13] and the video sequence used is *speedway.mj2* [16] containing 200 JPEG 2000 frames generated with an overall compression ratio of 20 for the base layer, 10 for the second layer and 5 for the third layer.

Figure 8 presents the successful decoding rate of the motion JPEG 2000 video sequence *speedway.mj2* [16] transmission over real ad-hoc network channel traces [13]. We observe that for highly noisy channels ($C/N \leq 15\ dB$), the proposed optimal layer outperforms other layer based FEC schemes but is less efficient than the packet oriented scheme. For noisy channel ($15\ dB \leq C/N < 18\ dB$), we notice that all layer based UEP schemes exhibit similar performances in terms of successful decoding rate. However, for low noisy channels ($C/N \geq 18\ dB$) all the FEC schemes yield the same improvement in terms of successful decoding rate.

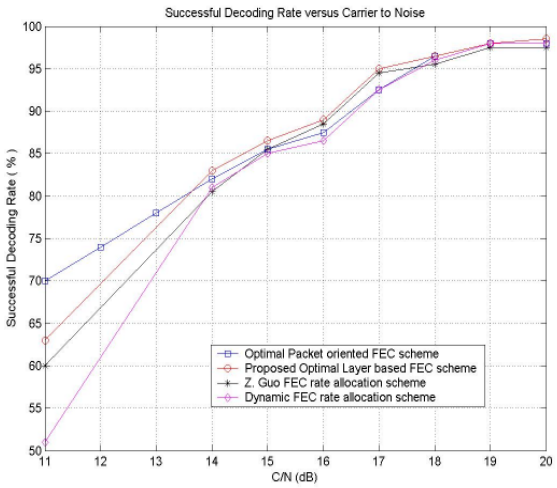



Figure 8. Successful decoding rate versus Carrier to Noise Ratio

In Figure 9 we show that our proposed optimal layer based FEC rate allocation scheme still outperforms other layer based schemes in terms of Mean SSIM. This is due to the fact that the base layer which is the most important part of the codestreams is highly protected in our proposed scheme, in comparison to other layer based schemes, guaranteeing this way a good quality for the visualization.

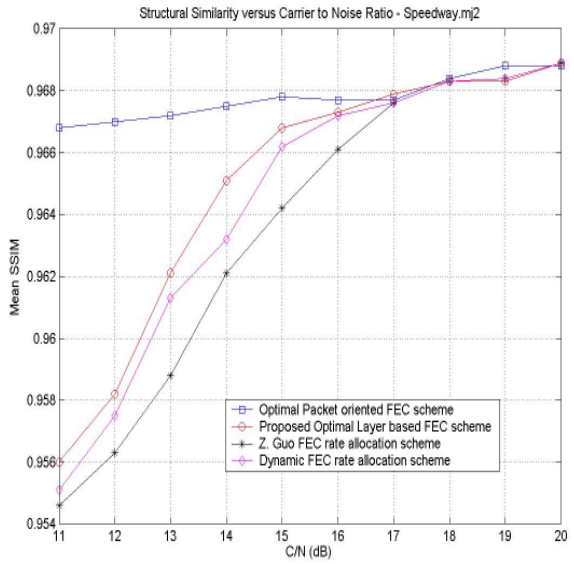



Figure 9. Mean Structural Similarity versus Carrier to Noise Ratio

It is worth noting that, for highly noisy channels, our optimal layer oriented FEC scheme is less efficient than optimal packet oriented FEC scheme presented in [6]. However the last one is unpractical for real time streaming applications when the number of packets in the codestreams is high. In contrast our proposed layer oriented efficiently overcomes this limitation. In this context, instead of being used to replace packet oriented FEC rate allocation schemes, our proposed optimal layer based FEC scheme should be used to complete it. Thus, an interesting extension to this work could be to consider the framework of unifying both families (packet oriented and layer oriented) and going straightforward to derive an optimal combined packet/layer oriented FEC rate allocation scheme for robust transmission of JPEG 2000 images/video over wireless channels.

## VI. CONCLUSION

In this paper we presented an optimal layer oriented FEC rate allocation scheme for robust JPEG 2000 images/video streaming over wireless channels. Compared to packet oriented FEC schemes, our proposed layer based scheme is significantly less time consuming while offering similar performances in terms of Structural Similarity for low noisy channels. However, for highly noisy channels, the packet oriented scheme is more efficient than the proposed scheme which leads us to the conclusion that both packet oriented and layer oriented schemes should not be viewed as antagonists FEC schemes, but should be combined in a new framework of combined packet/layer based FEC rate allocation scheme. In this context, our proposed optimal layer oriented scheme is a good candidate for both highly constrained video streaming applications and for future unified packet/layer based FEC rate allocation schemes.

## REFERENCES

[1] D. S. Taubman and M. W. Marcellin, “JPEG 2000 Image Compression Fundamentals, Standards and Practice,” Kluwer Academic Publishers, The Netherlands 2001

[2] JPEG 2000 part 11 Final Draft International Standard, ISO/IEC JTC 1/SC 29/WG 1 N3797

[3] M. Agueh, F. O. Devaux, and J. F. Diouris, “A Wireless Motion JPEG 2000 video streaming scheme with a priori channel coding,” 13th European Wireless 2007 (EW-2007), April 2007, Paris, France

[4] Z. Guo, Y. Nishikawa, R. Y. Omaki, T. Onoye, and I. Shirakawa, “A Low-Complexity FEC Assignment Scheme for Motion JPEG 2000 over Wireless Network,” IEEE Transactions on Consumer Electronics, Vol. 52, Issue 1, Pages: 81 – 86, Feb. 2006

[5] M. Agueh, J. F. Diouris, M. Diop, and F. O. Devaux, “Dynamic channel coding for efficient Motion JPEG 2000 streaming over MANET,” Proc. Mobimedia2007 conf., Aug. 2007, Greece

[6] M. Agueh, J. F. Diouris, M. Diop, F. O. Devaux, C. De Vleeschouwer, and B. Macq, “Optimal JPWL Forward Error Correction rate allocation for robust JPEG 2000 images and video streaming over Mobile Ad-hoc Networks,” EURASIP Journal on Ad. in Sig. Proc., Spec. Issue wireless video, Vol. 2008, ID 192984

[7] R. Prasad, C. Dovrolis, M. Murray and kc claffy, “Bandwidth estimation: metrics, measurement techniques and tools,” IEEE *Network*, vol. 17, pp. 27-35, 2003

[8] M. Li, M. Claypool and R. Kinichi, “ WBest: a bandwidth estimation tool for IEEE 802.11 wireless networks,” Proc. of 33rd IEEE Conf. on Local computer networks, 2008.

[9] Openjpeg, website: www.openjpeg.org

[10] A. Johnsson and M. Björkman, “On measuring available bandwidth in wireless networks,” Proc. of 33rd IEEE Conf. on Local Computer networks, 2008.

[11] D. Gupta, D. Wu, P. Mohapatra and C. Chuah, “Experimental comparison of bandwidth estimation tools for wireless mesh networks,” Proc. of IEEE Inocom mini-conference, April 2009

[12] M. Agueh, S. Ataman, and H. Soude, “A low time-consuming smart FEC rate allocation scheme for robust wireless JPEG 2000 images and video transmission,” 4th International conf. on Com. and Networking, 2009, Xi’an (China)

[13] Loss patterns acquired during the WCAM Annecy 2004 measurement campaigns IST-2003-507204, “Wireless Cameras and Audio-Visual Seamless Networking,” website: <http://wcam.epfl.ch/> 01-04-2009

[14] J. R. Yee and E. J. Weldon, “Evaluation of the performance of error-correcting codes on a Gilbert channel,” in IEEE Transactions on Communications. vol. 43, no. 8, Pages: 2316-2323, 1995

[15] A. Descampe, C. De Vleeschouwer, C. Iregui, B. Macq, and F.Marques, “Pre-fetching and caching strategies for remote and interactive browsing of JPEG 2000 images,” IEEE Transactions on Image Processing, vol 16, no 5, Pages: 1339-1354, 2006

[16] Speedway video sequences have been generated by UCL: <http://euterpe.tele.ucl.ac.be/WCAM/public/Speedway%20Sequence> 01-04-2009

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Tran. on Image Proc., vol. 13, no. 4, Pages: 600-612, Apr. 2004

[18] Max Agueh, Henoc Soude, "Optimal Layer-Based Unequal Error Protection for Robust JPEG 2000 Images and Video Transmission over Wireless Channels," mmedia, pp.104-109, 2009 First International Conference on Advances in Multimedia, 2009

# Modelling of Mobile Workflows with UML

Michael Decker
*Institute AIFB, University of Karlsruhe (TH)*
*Karlsruhe, Germany*
*Email: decker@aifb.uni-karlsruhe.de*

***Abstract*—Thanks to the advances on the field of mobile computing nowadays it is possible to realize workflow systems, which provide support for mobile activities, i.e., activities which are usually performed in situations where no stationary computer is available. It is obvious that the integration of mobile computers into workflow systems has a great potential especially for companies with a high portion of mobile workers. However, so far there is a lack of modelling techniques that allow to express mobile-specific aspects of workflows. In the following article we therefore introduce an extension to activity diagrams from the *Unified Modelling Language*, which allows to model different types of so called *location constraints*. Such a location constraint is a statement that defines where a particular activity of a workflow has to be performed or is not allowed to be performed. The proposed technique in particular supports *dynamic location constraints*, i.e., constraints that are defined during the runtime of a workflow instance. Further, different classes of anomalies that may occur in diagrams according to the proposed extension are also discussed.**

***Keywords*-Mobile Workflow Systems, Location-based Services, Business Processes, Activity Diagrams**

## I. Introduction

A workflow is a set of partial-ordered activities that is performed with the support of a special computer system (so called Workflow Management System) to reach a particular business goal [2]. This goal could be the fulfillment of an order to perform some kind of on-site repair (e.g., fix machine in factory or in a private apartment). The individual activities required to reach this goal include *receive customer's call*, *dispatch service technician*, *perform on-site-repair* and *post-processing* (e.g., writing bill, evaluation). There is also a partial order defined for these activities that states which activities have to be finished before a particular activity can be started or if there are optional activities. Such a description of a workflow is also called a *workflow schema*. The individual invocations of that schema (e.g., order no. 123 by customer A., also termed a *case*) is a *workflow instance* of that schema. A workflow schema is a *mobile workflow* if there are instances that have activities that are typically performed using mobile devices like Personal Digital Assistants (PDA), smartphones or notebooks. The workflow sketched above as example is such a mobile workflow, because during on-site repair activities the concerned actors (mobile workers) do not have access to stationary desktop computers, so they have to use mobile computers to query technical data about components to be repaired, to order replacement parts or to write on-site reports about which workings had to be performed.

It is unquestionable that the employment of mobile devices for the enactment of workflows provides a great potential [3][4]: even while staying in the field mobile workers have access to latest information from backend systems (e.g., technical documentation, customer data, state of orders, availability of items). Further, it is possible to avoid media disruptions because mobile workers don't have to print every information they might need on-site and don't have to write reports and gathered data (e.g., orders for spare parts, measurement readings) onto paper forms that have to be entered into a computer system. This helps to avoid costs (for paper as well as working time needed for data gathering), errors (e.g., typos, double entries, lost forms) and the time delay between printing and information consumption or writing down the information and entering it into the backend information system is reduced.

The contribution of the article at hand is the introduction of an UML profile for activity diagrams that allows to cover important mobile-specific aspects, namely the constantly changing location of the mobile actors. To enable this so called *location constraints* are introduced: such a constraint is a statement attached to an activity that says at which locations the respective activity has to be performed or is *not* allowed to be performed. Not only static constraints are possible, i.e., constraints that are defined at the design time of a workflow schema, but also dynamic constraints, which are derived automatically during the runtime of a workflow instance. To actually enforce location constraints it is necessary that the workflow system knows about the mobile user's current location. For this several locating technologies are available: the most prominent one is the satellite-based *Global Positioning System (GPS)*, but there are many other technologies, see [5] or [6] for a good introduction and overview on this topic.

Location constraints for workflows are a variant of *Location-aware Access Control (LAAC)*. Access control is concerned with the decision if a user's request to perform a particular operation on a resource of an information system has to be granted or denied. Examples for pairs of operation and resource are "read on report.doc" or "write on database table1". When LAAC is employed the access decision is not only based on the user's identity and properties of the

resource but also on the current location of the user as determined by a locating technology. The novel feature of our extension for activity diagrams from the perspective of LAAC is that it supports process-aware location constraints, i.e., that it is possible to formulate location constraints based on the order of activities within a workflow. Further, it is even supported that activities of a workflow instance create location constraints for subsequent activities of the same workflow instance.

The work at hand is the extended version of a paper that was presented at the IARIA UbiComm-conference 2009 [1]. In addition to the original paper we also introduce some shortcut notations (Section V-B) and a notation for derived constraints (Section V-C). Further, we discuss the problem of anomalies with location constraints (Section VIII) and sketch how our concept of location constraints can also be applied for UML usecase diagrams (Section IX).

The following sections are organized as follows: In Section II the basics necessary for the understanding of the paper are covered. The underlying location model for our modelling approach is introduced in Section III. Based on these preparations the concept of *location constraints* is introduced in Section IV. The next Section V is devoted to the visual representation of the different types of location constraints. A detailed example of an activity diagram annotated with several location constraints can be found in Section VI. In Section VII it is elicited how UML's metamodel has to be extended to obtain a UML profile for location constraints. There are different types of anomalies that may occur if location constraints are used; this is the subject of Section VIII. Location constraints can also be used to annotate usecase diagrams which is explained in Section IX. A short survey on related work can be found in Section X before we conclude with a summary and outlook to future work in the final Section XI.

## II. Preliminaries

In this section we describe the basics from the domain of *Location Aware Access Control* and the *Unified Modelling Language* that are required for the understanding of the discussion in this paper.

### A. Location-Aware Access Control

When *Location-Aware Access Control (LAAC)* is applied the user's current position is considered by the component of the system that makes the access control decision [7]. For example, it could be enforced that a user is not allowed to access confidential data with his mobile computer when he stays at places that are deemed as unsafe, e.g., public places, countries where espionage has to be feared or regions where no secured wireless data transmission is available. So LAAC is a mean to tackle specific security challenges that come along with the employment of mobile computers. These challenges arise from the fact that due to their portability and size mobile computers often get lost or stolen. There is also the danger that someone looks over the shoulder of the legitimate user to gather data that is classified ("shoulder sniffing" or "shoulder surfing"). Further, wireless data transmission could get eavesdropped (passive attack) or even manipulated (active attack).

Another serious challenge in mobile computing are usability problems, since mobile devices only have a small display of limited quality and rudimentary means for data input (e.g., no full keyboard). Because of this the mobile user will appreciate it if data items and elements of the graphical user interface are hidden, when they are not relevant for him at his current location. For example, if a mobile service technician stays in a particular city all files concerning customers in other cities could be hidden.

### B. Unified Modelling Language

The Unified Modelling Language (UML, [8]) is the result of the consolidation of several independently developed modelling languages (e.g., Object Modelling Technique (OMT) and Object Oriented Software Engineering (OOSE)) from the domain of software engineering. Nowadays UML is maintained by an industry consortium, namely the Object Modelling Group (OMG).

UML comprehends 13 different diagram types. On the uppermost level we can discern diagrams to describe *structural aspects* of software systems and *behavioural aspects*. A well-know example for the former type are *class diagrams* that are commonly used to describe data structures in object oriented programming languages (e.g., Figure 1). Activity diagrams belong to the latter type of diagrams. In the next section we show how these diagrams can be extended to enable expressing mobile-specific aspects.

The purpose of UML's activity diagrams is to describe workflows, i.e., to represent a set of activities and the relations between these activities. An individual activity is depicted as box with rounded edges that contains the name of the activity. The initial activity is a black circle and the final activity is represented as black circle within a bigger white circle. Arrows with solid lines connect these boxes to indicate which activities have to be finished before another one can be started; this way the control flow of the workflow is defined. To enable depicting conditional flows of activities there are control nodes (shown as diamonds) that are used to connect arrows. Further concepts in activity diagrams are swimlanes to group activities together (e.g., all activities that have to be performed by the same actor or organisation) or the forking/joining of the control flow for parallel activities.

Many other graphical languages for the modelling of workflows and business processes can be found in literature, e.g., Petri-nets, Event-driven Process Chains (EPC) or

Flowcharts. The distinguishing feature of UML activity diagrams for our purpose is that it provides an explicit defined meta-model and thus explicitly supports the definition of extensions.

## III. Location Model

Before the actual extension of UML activity diagrams can be described we have to introduce a simple location model. For the sake of simplicity we consider only two dimensions, but it is simple to extend the model to support three dimensions which might be necessary to cover indoor scenarios where for example offices at different floors of a building have to be distinguished.

As geometric primitives there are *Point* and *Polygon* which both are extensions of the abstract superclass *AbstractGeometry* (see Figure 1). For polygons it is demanded that the lines of a given polygon don't cross each other. If a class is abstract this means that it is not possible to directly create instances of that class; to obtain an instance one has to create a concrete (i.e., non-abstract) subclass. A *circle area* is associated to one point that represents the center point of the circle; the radius of that circle is defined as member variable of class circular area and holds a double value that defines the radius of that circle in meters. *LocationInstances* are associated with exactly one polygon and belong to exactly one *LocationClass*. Location classes are used to group location instances that conceptually belong together, e.g., there might be location classes that represent cities, countries, districts or rooms within buildings. The concept of location classes and instances can also be found in the *Geographic Markup Language (GML)* in the form of feature types and features [9][10]. It is demanded that two polygons that belong to location instances of the same location class do not overlap spatially. Examples of location instances for location class *city* are *Malta* or *Berlin*. To exemplify this model also some instances of location instances and classes are drawn in the figure (surrounded by the box with the dotted line). It is *not* demanded that all the instances of one particular location class cover the whole reference space, i.e., that a given class provides an exhaustive classification. This is the case for class *city*, because not every point in a country can be assigned to a city, there are also rural areas. But a location class *country* could provide an exhaustive classification if the reference space is the whole area of a given continent; each point on that continent can be assigned to one instance of that class which represents a particular country.

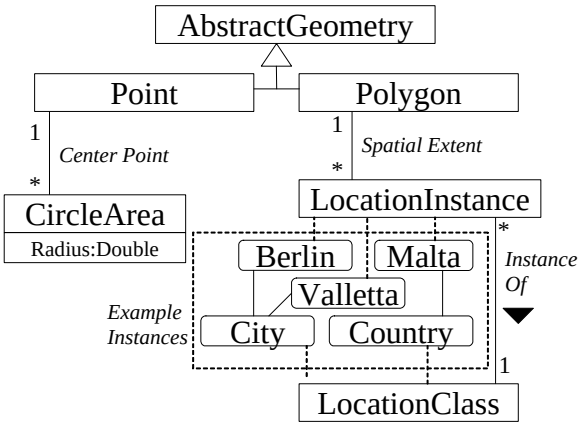



Figure 1. Location Model as UML-Class Model

## IV. Location Constraints

In this section the concepts of *Location Constraints* and *Location Rules* are introduced.

### A. Definition and Classification of Location Constraints

A *location constraint* is a statement about the location where one or more activities of a workflow schema or instance have to be performed or are not allowed to be performed. Constraints of the former type are called *positive constraints* while constraints of the latter type are *negative constraints*. These two types of constraints can be found in the classification of location constraints depicted in Figure 2.

A positive constraint could be demanded if a company wants to enforce that particular workflow activities dealing with confidential customer data are not performed outside the company's premises. Negative constraints will usually be employed if it is easier to express where something is not allowed than to enumerate the places where something is allowed; for example, if there are only a few countries in the world where certain software functions of a mobile workflow system should not be used (e.g., because of license or export restrictions) then a negative constraint will be defined that enumerates all these countries.

There is another dimension to classify location constraints which can also be found in Figure 2 and which is orthogonal to the distinction of positive and negative constraints:

**Static constraints:** These constraints are defined for the workflow schema *before* the runtime of the individual workflow instances. This implies that these constraints — which can also be called schema constraints — are enforced for all workflow instances that are created from that schema.

**Dynamic constraints:** They are defined *during the runtime* of a workflow instance are only valid for that instance. Further, we distinguish *external* and *internal* dynamic constraints: For external constraints the actual locations are not calculated by the mobile workflow system itself; rather, they are defined manually by a human operator during runtime (e.g., dispatcher, manager) or they are queried from an another information systems, e.g., a customer relationship management database that stores the addresses of all the customers of a company. When an internal constraint is defined then all the information needed by the mobile workflow system to calculate the actual location during the

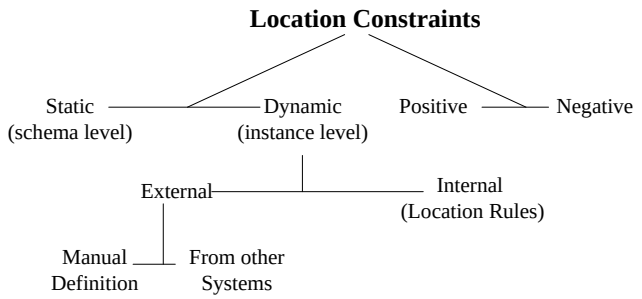



Figure 2. Different types of location constraints depicted as classification tree

runtime of the workflow instance has to be specified in the workflow graph. For this we propose so called *location rules* which are elicited in the subsequent subsection.

### B. Location Rules

The basic concept behind so location rules is that based on the user's current location where he performs a particular activity called *trigger activity* a location constraint for another activity called *target activity* can be derivated. If the location constraint created by a rule is a positive one this means that both the trigger and the target activity have to be performed at the same location, so this is called *binding of (same) location*. If a negative constraint is created then the target activity cannot be performed at the same location as the trigger activity; this case can also be called *prohibition of same location* or *separation of locations*.

These two types of location rules were greatly inspired by the work of Bertino at al. [11]: they applied the well know security principles *Separation of Duties (SoD)* and *Binding of Duties (BoD)* [12] to the workflow domain. SoD for workflow management means that if a given user performed a particular activity of a workflow instance he is not allowed to also perform a particular other activity of the same workflow instance. The standard example for SoD is an approval workflow where the submitter of the proposal is not allowed to decide over his own proposal. An example for BoD is the policy *one face to the customer* that says that the actor who had the initial contact with a customer has to perform all other activities for the same workflow instance that involve communication with the customer. Bertino et al. even consider the case of inter-instance constraints, e.g., if Alice had to perform the activity *make decision* in a proposal workflow instance started by Bob then Bob isn't allowed to perform that activity in a subsequent proposal instance initiated by Alice to prevent collusion.

To define what is *the same location* the location model has to be employed. A location rule can refer to a location class; the instance of that class that contains the user's location when performing the trigger activity is the location that is used for the target location. It is also possible to define a radius that is used to calculate a circle around the user's location during the activation of the trigger activity.

Table I
EXAMPLE FOR AN IMPLICATION LIST WHICH MAPS LOCATION INSTANCES OF CLASS *country* TO LOCATION INSTANCES OF CLASS *city*

| Trigger Location | Target Location |
|---|---|
| Germany | Munich |
| France | Munich |
| Malta | Valletta |
| ... | ... |

Another approach is to have rules that create constraints pointing to a location different from the location where the trigger-activity was performed; for these rules with *implied locations* it is necessary to have lookup tables that map *trigger locations* to *target locations*. As example such an implication list is sketched in Table I: this table maps different countries to the city where the headquarter of an international company is. It is allowed that different countries are mapped to the same location, e.g., all the orders from either France or Germany have to be handled in Munich, because the number of orders from France is so small that it wouldn't make sense to operate a headquarter in France.

## V. VISUAL REPRESENTATION

In this section first the basic elements of the visual representation are introduced. After this, shortcut notations are covered and the depiction of constraints derived based on location rules during the runtime of a workflow instance.

### A. Basic Elements

As visual representation for the different types of location constraints and rules in UML activity diagram we propose the one which is sketched in the table that can be found in Figure 3. All the constraints in the left column are positive constraints or produce positive constraints, while the right column is devoted to negative constraints. The uppermost row shows static location constraints while all the other three rows show different kind of dynamic location constraints.

- All location constraints and rules are attached as dotted arrows to the boxes that represent the activities. On the dotted arrow there is a circle that holds a symbol which indicates the *mode* of the constraint: *Positive Static Constraint* and *Binding of same Location* are symbolized by "$=$", *Negative Static Constraint* and *Prohibition of same Location* are symbolized as "$\neq$", rules for *Implied Binding of Location* are symbolized as "$\Rightarrow$" and *Implied Prohibition of Location* are symbolized as "$\not\Rightarrow$".
- Static constraints are shown as arrows pointing from a parallelogram to the constrained activity. The parallelogram stands for a location instance and holds the name of that location.

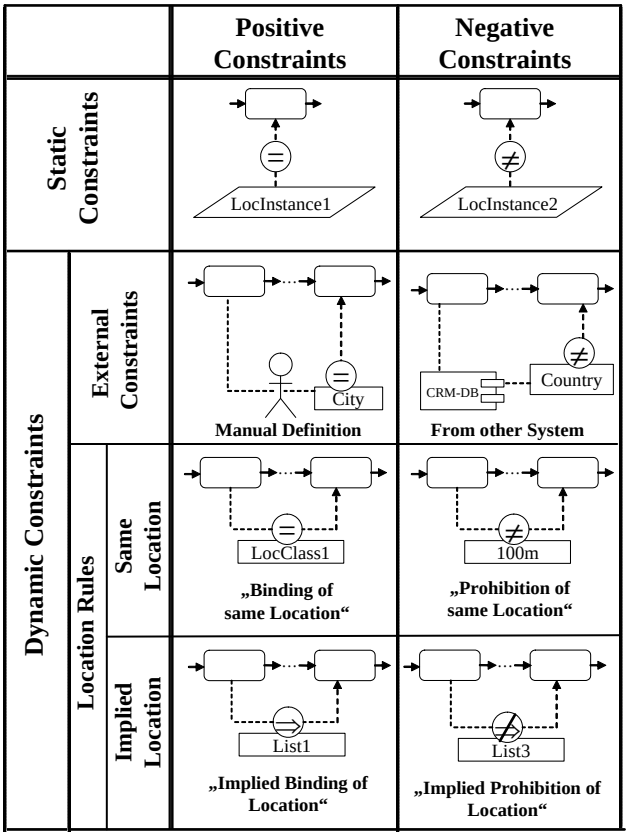



Figure 3. Different types of location constraints

- There are two types of external dynamic constraints, namely those where the actual constraint is entered manually by a human operator or is retrieved from another information system connected to the workflow system. Human operators are represented by a symbol that shows a little man. This symbol is also used to represent actors in UML usecase diagrams. An external information system as source for a dynamic location constraint is depicted by the symbol for a software component that was used in the deployment diagrams of UML 1.x. An external constraint is also depicited by a dotted arrow which connects two activities. The activity where that arrow starts is the activity which triggers the retrieval of the dynamic constraint. It is possible that the trigger activity is solely devoted to the retrieval of the location constraint; but the triggering could also be just a secondary effect of an activity. The target of the dotted arrow is the activity to which the retrieved location constraint will be assigned to. On the line of the arrow the symbol for the source of the constraint as well as a circle with hold the mode (positive or negative) and the granularity (location class or radius) is drawn.

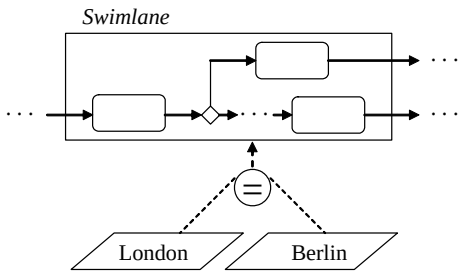



Figure 4. Location constraint for all activities in a swimlane

- A rule for the creation of dynamic constraints is shown as arrow pointing from the *trigger activity* to the *target activity* that will be bound to the derived location. For *same location rules* the granularity of what constitutes the *same location* is annotated by a box attached to the circle; the box can hold either a location class or a numeric value as radius; for *implied location rules* the box holds the name of the implication list. For example, if the location class *city* is used for the rule then the city that covers the user's current position is assigned as location constraint to the *target location*. If such a city cannot be found then no constraint is generated.

### B. Shortcut Notations

We also devised some shortcuts which help to reduce the number of required elements to depict location constraints in some scenarios.

In Figure 4 several activities are contained within a swimlane (rectangular box). In UML swimlanes are usually employed to group activities together that are performed by the same actor. It is also possible to assign a location constraint to a swimlane, which then has to be enforced for all activities within that swimlane. This example shows also that it is possible to assign more than just one location with a static location constraint. If two or more locations are assigned as positive location constraints this means that the constrained activities have to be performed in one of these locations; in the example in Figure 4 this would mean that all the activities within the swimlane have to be performed either in London or Berlin.

When several locations are assigned to a negative location constraint this means that the activity can be performed everywhere but not at any of the assigned locations.

The fragment in Figure 5 represents the case where two different activities share the same location constraint. For these fragments the meaning is that both activities have to be performed somewhere in Canada; however, it would be possible to perform the two activity in different Canadian cities. Such a constraint could occur in practice if the pre- and post-processing of a workflow should be performed within a particular country, but all the other activities can be performed somewhere else in the world.

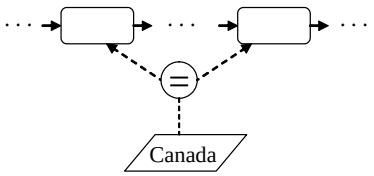



Figure 5. Single location constraint shared by two separate activities

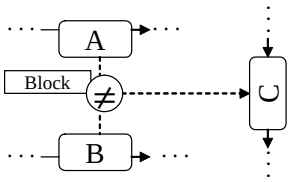



Figure 6. Location rule with two trigger activities

There are also shortcuts for location rules like shown in Figure 6: in this example one location rule has two trigger activities. The location instance of class *Block* (short for *Block of flats*) which contains the user's location when he performs activity A or B will define a negative constraint that forbids the execution of activity C at the same block. Analogous to this it is possible to have more than just one target activity for one location rule; this would mean that the location constraint created when that rule is triggered is attached to all the target activities at the same time.

### C. Visualisation of derived constraints

While not necessary for the actual workflow modelling for the purpose of demonstration it is useful to be able to depict the static location constraint generated by a dynamic constraint; the generated constraint can also be called *derived constraint*

A static constraint generated by a dynamic constraint is depicted as a static constraint (as parallelogram) and attached with a dotted line to the box that indicates the granularity of the contraint. In Figure 7 this is shown for the case of a location rule, but the notation can also be used for external constraints. The rule in the figure creates a positive constraint that restricts the execution of the target activity to the city where the trigger activity was performed. In the depicted example the trigger activity was performed somewhere in Italy, so that an accordant static constraint was created.

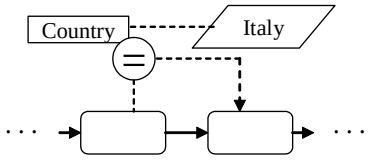



Figure 7. Depiction of derived constraint

## VI. Example Scenario

To exemplify the application of our UML profile the workflow already sketched in the introductory section is elaborated in more detail and shown as diagram with location constraints in Figure 8. This workflow could be found in a company that provides technical maintenance services and employs many technicians which are sent to the customers' premises to perform on-site repair works of technical components.

An instance of the depicted workflow schema is created when a call center agent receives a telephone call from a customer. This activity has a static location constraint that points to all the location instances that represent the premises of call centers operated by the company. The incoming calls are routed to the call center that is the nearest with regard to the origin of the call, so the local service center that has to send a mobile worker to the customer's premises (dispatching) is defined based on a dynamic location constraint: the district in which the dispatching activity has to be performed is determined by an external application that makes this decision by evaluating the caller's phone number. The service center first sends an inspector to the site (e.g., factory, private residence, places with components of public infrastructure like pipes, electric transformation station, junction box) where the allegedly defective component can be found. If the inspector cannot find a technical defect the mobile part of the workflow is aborted and *Follow-up Office Work* is performed as final activity. However, if there is indeed a defect the inspector will order a mobile repair team to the location of the component using his PDA. The location where that team can perform the actual repair work is restricted with a dynamic constraint in form of a location rule (binding of same location) which says that the repair activities have to be performed not farther away than 150 meters from the point that was determined by the inspector's GPS device when he placed the order for the mobile repair team.

Since some components cannot be repaired on-site a possible branch of the workflow is *Shop Floor Repair* at a special repair shop operated by the company. To minimize transportation ways the location of this activity is constrained with another dynamic constraint that is based on an implication list; this implication list maps each region where the company provides its service to the nearest repair shop. If the activity in the shop floor is finished then another invocation of the activity *On-Site Work* has to follow since the repaired component has to be brought back and reinstalled. It is possible that the sequence of on-site work and shop-floor work is performed several times, e.g., if the defect is not solved after the first component that was repaired in the shop-floor is brought back.

The final activity is called *Follow-up Office Work* and deals with writing the bill, ordering new spare parts and

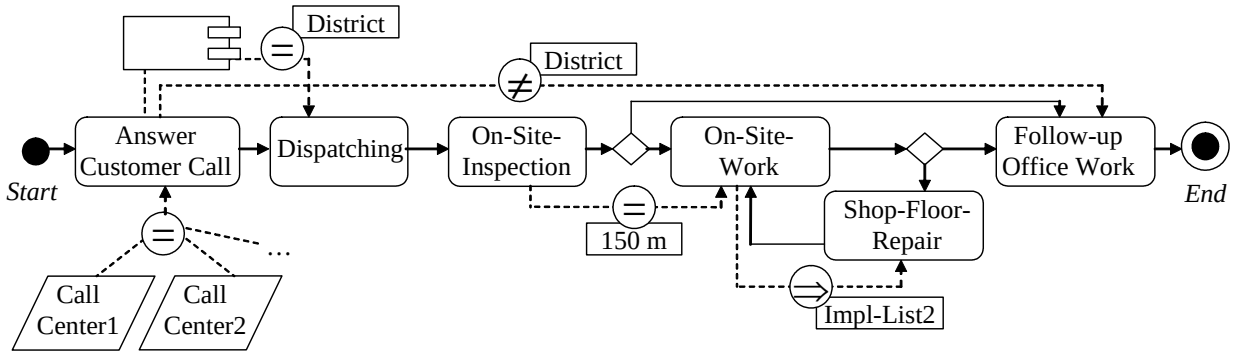



Figure 8. Example workflow with different kinds of location constraints

writing a report for the customer and the manufacturer of the component that had to be repaired. Since this activity also includes some kind of evaluation (e.g., how fast the customer's problem was solved, level of customer satisfaction) a dynamic constraint (prohibition of same location) is used to demand that the office center where this activity is performed is not in the same district as the one that initiated the workflow instance. The purpose of this rule is to ensure that employees knowing each other cannot collude to cover up mistakes that were made.

## VII. UML Profile

We are now prepared to show how the proposed extension to UML activity diagrams fits into the meta-model of UML. For the sake of brevity only the most important new constructs are covered. In Figure 9 the package *MobileWorkflowProfile* is shown, that represents the new UML profile. Outside of the package some important classes from the UML meta-model are shown.

The profile also contains the location model which was introduced in Section III. Due to space restrictions only the parts of the location model are drawn that are necessary for understanding the relation of the location model and the rest of the model. There are also two new classes to model implication lists for rules with implied locations: *ImplList* and *ImplPair*. Each instance of *ImplPair* stands in association with two location instances to represent the *trigger location* and the *target location*. These instances of *ImplPair* (Implication Pair) represent the individual entries of an implication *ImplList* (Implication List).

The main connection between the UML meta-model and our profile is the extension relationship between class *Activity* and *ConstrainedActivity*. Instances of *ConstrainedActivity* have to be used if any kind of location constraint has to be assigned to an activity. For this each *ConstrainedActivity* stands in association with at least one instance of class *AConstraint*, which is the short form for *Abstract Constraint*, i.e., it is not possible to obtain direct location instances of that class. A flag named *isPositive* indicates if the *AConstraint* represents a positive or a negative constraint. There are two direct subclasses of *AConstraint*, namely *ADynConstraint* (short for *AbstractDynamicConstraint*, another abstract class) and *StaticConstraint*. Each instance of class *StaticConstraint* stands in association with at least one instance of *LocationInstance*. *ADynConstraint* as the second subclass of *AConstraint* has two direct subclasses: *ImplLocDynConstraint* and *SameLocDynConstraint*. *ImplLocDynConstraint* is associated with exactly one *ImplList*. Class *SameLocDynConstraint* stands in association with class *LocClass*. However, this association is an optional one because what is considered as "the same location" cannot only be defined based on a location class but also based on a radius. For the latter case there is also a member variable *radius* in class *SameLocDynConstraint* that is set to a value greater than 0 if the modeller thinks it is more appropriate to derive the "same location" using a radius than looking a location instance. For a given instance of *SameLocDynConstraint* it is not allowed to use both methods to define the "same location", i.e., either the radius is greater than 0 or there is an associated location instance. Using UML's *Object Constraint Language (OCL)* it is possible to express this formally [13]:

```
context SameLocDynConstr
inv     (not targetClass->isEmpty()) XOR
        (radius >= 0.0)
```

The keyword *context* is followed by the name of that class from whose perspective the following statement has to be viewed. For the given statement this means that *targetClass* has to be the name of an end of an assocation assigned to class *SameLocDynConstr* and that *radius* has to be the name of a member variable of that class. The keyword *inv* stands for *invariant*, i.e., the following boolean expression has to evaluate to *true* for all instances of the model. *IsEmpty()* is a function that returns *true* if for the considered instance there is no assocation to an object of *LocClass*.

Class *ActivityEdge* from the UML meta-model couldn't

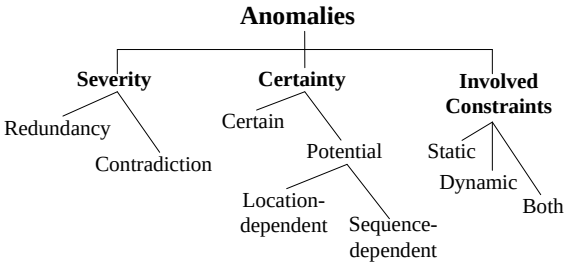



Figure 10. Different types on anomalies

be used to represent the arrows introduced in our profile to assign location constraints to activities because the edges in activity diagrams can carry tokens and we opted to have a different graphical representation (dotted arrows instead of solid arrows for the sake of better readability).

## VIII. Anomalies of Location Constraints

When an activity diagram is annotated with location constraints according to the UML profile introduced in this paper it is possible that anomalies occur. In this section we will describe different types of such anomalies in terms of pairs of location constraints. The different dimensions considered to classify anomalies of location constraints are depicted in Figure 10.

The first dimension to classify the different types of anomalies is the *severity level*:

- A location constraint could make a **redundant statement**, i.e., the constraint could be removed without altering the behaviour of the mobile workflow system for any thinkable workflow instance. There are two subtypes of redundancy: the constrained activity is never reached or another constraint makes the same restriction or an even stronger one. The first case would occur if a static constraint is made to an activity that is never reached; this means that there is an inconsistency in the underlying workflow schema. Another case is a location rule whose target activity cannot be performed again once the trigger activity is reached; this means the location constraint generated by the trigger activity will never have any effect.
- Two location constraints could make **contradictory statements** for a given activity, i.e., there is no place where that activity could be performed by an actor according to both the constraints. The suggestive approach to deal with such contradictions would be to assign priorities to the individual constraints, so that in case of a conflict only the constraint with the higher priority is considered.

Redundant statements do not cause problems during the executing of a workflow instance but unnecessarily bloat the model. Further, the existence of a redundant statement might be a hint that there was an error during the elicitation of the required constraints.

Most workflows schemas have decision points, i.e., there are different sets of activities that are actually performed for a given workflow instance. For example, there might be a branch of a workflow schema that isn't executed for every instance. Further, the order of the activities can also differ between two workflow instances of the same workflow schema. Some anomalies may occur depending on the location of the actor while performing a particular activity. This leads to the next dimension for the classification of anomalies called *certainty*:

- A **potential anomaly** will not appear in every workflow instance of a given workflow schema. We distinguish two subcases of potential anomalies: the anomaly appears depending on the order and/or the set of actually performed activities (sequence-dependent) or merely based on the location where one or more activities where performed (location-dependent) no matter of the sequence of performed activities.
- A **certain anomaly** will appear in every workflow instance of a given workflow schema. So to detect this type of anomaly we just have to look at one potential workflow instance of that schema.

It also can be considered between what types of location constraints the anomalies arise:

- If two static constraints produce an anomaly this is called **intra-static anomaly**. For example, a positive static constraint assigned directly to an activity could demand that that activity is performed in London. However, this activity is contained in a swimlane which has also a static constraint which says that the activities are *not* allowed to be performed in England. Since London lies within England there is no place that satisfies both contradictory constraints.
- If both constraints involved into are dynamic constraints it is named **intra-dynamic anomaly**. As example we consider an activity that is the target of two positive location rules. During runtime the first rule creates a constraint that says that this activity has to be performed in Berlin; subsequently, the second rule creates a constraint that says that the same activity has to be performed in Spain. Since Berlin isn't a City in Spain this is a contradiction.
- The last case are **inter-static-dynamic anomalies** where both types of constraints are involved. So the pair of the conflicting constraints has one dynamic and one static constraint.

In Figure 11 three activity diagrams of tiny workflows are depicted which cover the three cases of inra-static, inter-static-dynamic and intra-dynamic anomalies:

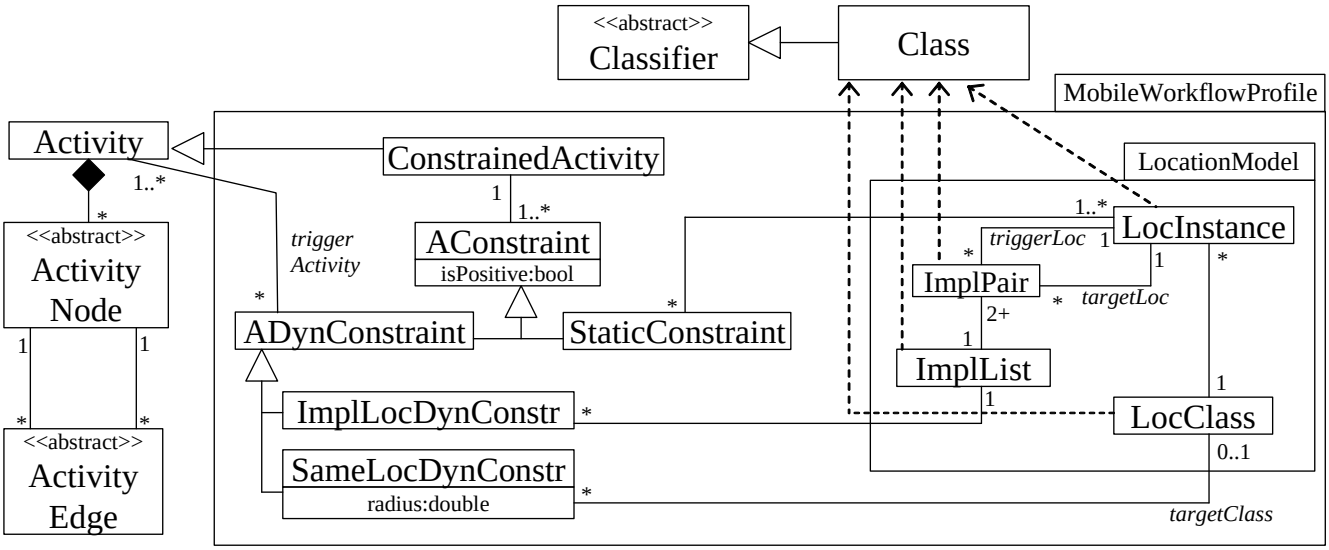



Figure 9. Extension of the meta-model of UML Activity Diagrams

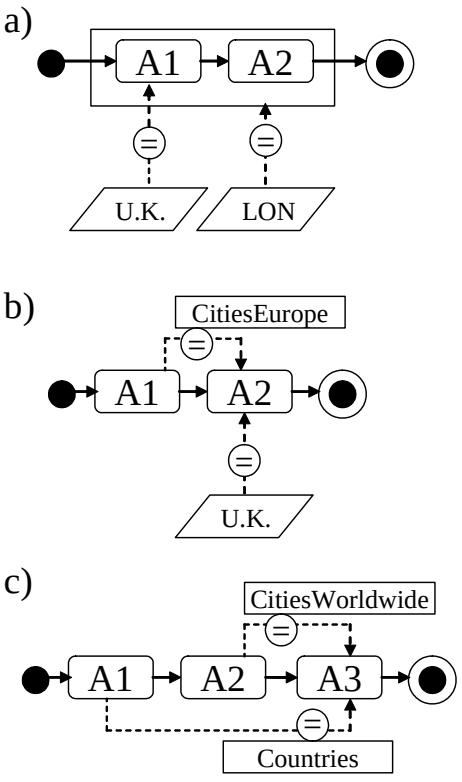



Figure 11. Examples for anomalies

- Diagram *a)* shows an intra-static anomaly. The swimlane that contains the two activities *A1* and *A2* has a constraint which confines the execution of these activities to the location *London*. Further, Activity *A1* has an individual constraint that restricts the execution of this activity to the location *United Kingdom (U.K.)*. This represents a redundancy, because *A1* is also confined by the constraint assigned to the swimlane, which is even stricter, since the location *London* is a subset of *U.K.* If the constraint assigned to activity *A1* would be omitted it wouldn't change semantics of the diagram. This case could occur if the constraint to *A1* was assigned long before the swimlane or the constraint for the swimlane was created and it was forgotten to remove the then redundant individual constraint for *A1*.
- Diagram *b)* shows an intra-dynamic anomaly: Activity *A2* is the target activity of a location rule and has also a static constraint. The location rule's trigger activity is *A1* and the derived location constraint will be an instance of the location class *Cities of Europe*. *A2's* location constraint points to the location *United Kingdom (U.K.)*, but not all European cities are located within the U.K., so this location rules represents are potential anomaly in the form of a contradiction.
- Diagram *c* shows an intra-dynamic anomaly: there are two location rules with the same target activity *A3*. The upper location rule (with trigger activity *A2*) assigns a location constraint that confines *A3* to a location instance of the class *Cities Worldwide*; the lower location rule with trigger activity *A1* assigns a location constraint that confines *A3* to a location instance of the class *Countries*. Since it is possible, that *A2* is

performed in a City that doesn't lie in the country where *A1* was performed this represents a potential anomaly in the form of a contraction. For example, it could happen, that *A3* has to be performed in *Germany* and *Lisbon* according to the derived location constraints, which is obviously impossible to satisfy, since Lisbon is not a German city.

In Figure 12 three more activity diagrams with potential anomalies are shown:

- In diagram *a)* the location constraint assigned to activity *A1* represents a redundancy (see also first diagram in Figure 11). However, the branch with the swimlane isn't executed in every run of this workflow this anomaly is a sequence-depend anomaly.
- Diagram *b)* shows a workflow schema with two concurrent branches. The potential anomaly is a contraction if *A2* is performed before *B2* in an European City outside the U.K. But if the execution of *B2* is started before *A2* is executed then the derived location constraint won't effect the execution of *B2* and so not contradiction occurs.
- The third diagram *c* in this figure has again two branches of which only one is executed. If the branch with activity *A* is executed in an European City this will lead to a contradiction, because the final activity *C* can only be executed in the *USA* according to the static constraint.

So all the anomalies in this figure are sequence-dependent. As example of a location-dependent anomaly we can consider diagram *b* in Figure 12: in this case a contradiction occurs if *A1* is executed outside the U.K.

## IX. Location Constraints for Usecase Diagrams

Another diagram type which can be found in UML is for *usecase diagrams* [8]. The purpose of a usecase diagram is to show which functions (usecases) a software system should provide. Such a usecase is depicted as ellipse that containts a short textual description. It further shows different user roles and how these roles are connected to individual usecases. A role is depicted as human operator (the same symbol that was used for manual location constraints) and is connected by a line to each usecase he is allowed to perform. If the system to be developed consists of several subsystems (e.g., distributed system) then each subsystem is represented by a rectangular box that contains the usecases that are operated by this subsystem. Usecase diagrams have also the ability to show how individual usecases stand in relation to each other, e.g., if one usecase always invokes another usecase. However, this features isn't relevant for our consideration.

Usecase diagrams are usually employed at an early stage of the software development cycle. Based on the results of the usecase analysis activity diagrams can be developed.

Location constraints as defined in Section IV can also be used to annotate usecase diagrams. An example for such a diagram can be found in Figure 13: The left subsystem contains two usecases, namely *Create new order* and *Finalize order*; the right subsystem has also two usecases which are named *Create new account* and *Edit master data*. There are three roles in the diagram: *Travelling Salesman*, *Manager* and *Accountant*. It is possible to assign static location constraints to roles, usecases, system boundaries and the association between a role and a usecase. Further, usecases can also be the trigger or the target of a location rule: the *part of town* where the usecase *Create new order* is performed defines where the usecase *Finalize order* has to be performed. The rationale behind this rule is that a travelling salesman should only be allowed to finalize the order where he started it. Further, the role *travelling salesman* is restricted to Spain by a location constraint. This says that users of that role can only invoke usecases when they are within Spain. However, users with role *manager* are allowed to invoke the usecases they are assigned to without any spatial restriction. The third role *Accountant* has also a location constraint that prevents users with that role of invoking usecases when they are outside the rooms of the company's *accounting department*.

The subsystem at the right side of the figure has a static location constraint that confines all the usecases provided by that subsystem to the location *company premises*.

## X. Related Work

In [14], an extension for UML activity diagrams is proposed that aims at modeling security requirements for business processes. The model introduces stereotypes to attach security requirements (e.g., privacy, access control, non-repudiation) to different elements in UML activity diagrams. For example, using this notation it is possible to assign a stereotype ≪privacy≫ to an activity partition (swimlane) to express that the disclosure of sensitive personal data has to be prevented during the activities contained by the within the swimlane.

Another profile for UML activity diagrams can be found in [15]. The purpose of this profile is to annotate workflows with concepts from the domain of business intelligence. Using this profile the modeller can express which activities require input from a data warehouse or data mart.

Baumeister et al. devised another extension of activity diagrams for modelling mobility [16]. But this approach is aimed at expressing how physical objects are moved by activities from one location to another and doesn't allow to formulate location constraints.

In literature some articles that propose location-aware access control models can be found. Notably most of them are extensions of Role-Based Access Control (RBAC, [17]) like *GEO-RBAC* [18], *LoT-RBAC* [19] or *S-RBAC* [20]. The most prominent aspects that distinguished these models is

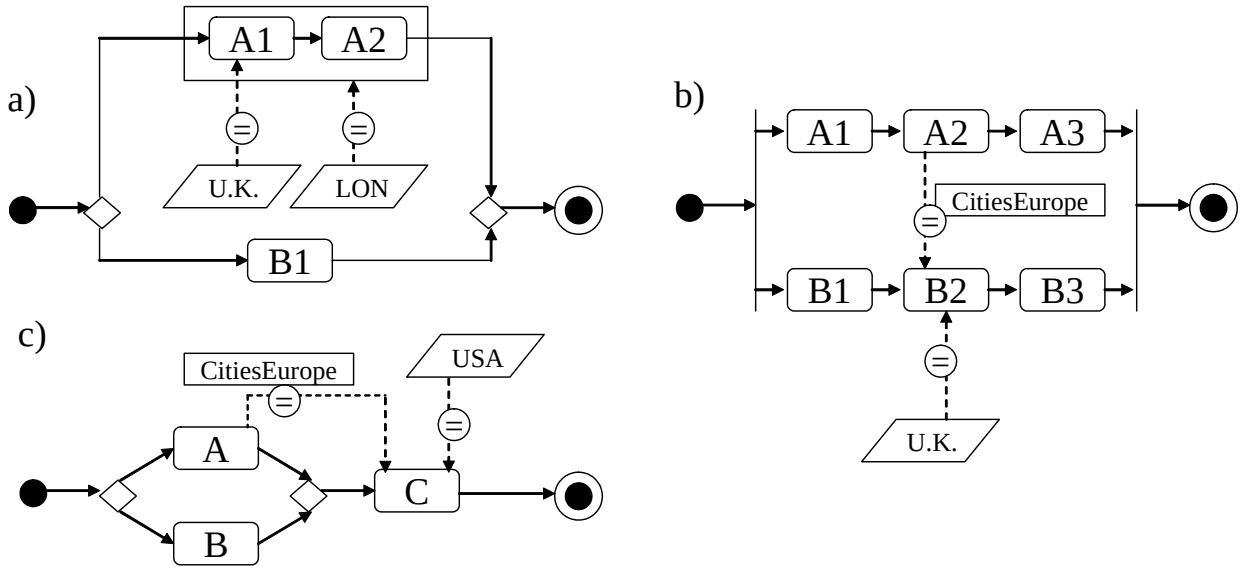



Figure 12. Examples for potential anomalies

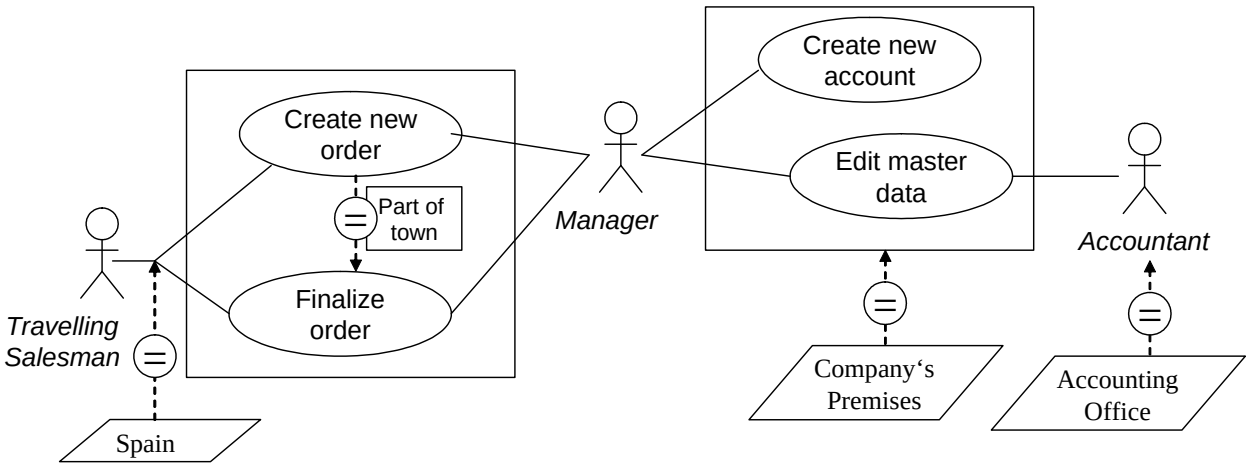



Figure 13. Usecase diagram with different types of location constraints

the subset of RBAC-components that can restricted with a location constraint. For example, in GEO-RBAC the location constraints are assigned to the roles itself, so depending on the mobile user's current location individual roles are switched on or off; in the S-RBAC-model location constraints can be assigned to the association between roles and permissions, so individual permissions of a role are switched on or off depending on the location. We surveyed these models in another article [7].

If access control decisions are based on the determination of a mobile user's location this leads to the question how trustworthy the employed locating system is. An attack with the intent to manipulate the location delivered by a locating system is called *location spoofing*. Such attacks can be performed by the possessor of the mobile computer (internal spoofing) or third party (external spoofing). In [21] we give an overview on different technical approaches for the prevention of location spoofing.

To the best of our knowledge we are aware of only one further paper by another group that deals with location constraints for workflows [22]. However, this model only allows to assign individual locations to activity nodes to state where

the respective activity is allowed to be performed. There are also location constraints assigned to the individual actors in the model. The focus of this work is the development of an algorithm to check if there are enough employees who can perform the individual activities of a workflow under the consideration of the respective location constraints.

In a recent paper we considered the problem of detecting inconsistencies in location aware access control policies [23]. However, this work focused solely on RBAC and didn't take any workflow-specific aspects into account. The basic idea is that it is possible to assign location constraints to several components of different type in a model at the same time. For example, a role *service technician* could have a location constraint so that this role can only be enabled in a certain region. Further, there could be also a location constraint assigned to a user that restricts his usage of the mobile information system to the city where he has to work. If the role *service technician* is assigned to that user it could occur that the intersection area of the two location constraints is empty, i.e., the user could never activate the newly acquired role. Such an "'empty assignment"' is a strong hint that the administrator of the system made a mistake and should be informed about this.

Further, the concept of *spatial coverage* as way to check the validity of location aware access control policies is introduced. For example, the coverage of a given role with respect to the entity *user* is the spatial extent of all the points where at least one user could activate that role according to the user's and the roles location constraints. If the coverage for the role *service technician* doesn't cover locations that should be served by the technicians then this again is a strong hint that something is wrong with the configuration of the access control model. In [23] we also describe several other types of spatial coverage for the purpose of model checking.

## XI. Conclusion

In the article, an UML profile was introduced that extends activity diagrams to express spatial constraints — so called *location constraints* — concerning where individual activities have to be performed or are not allowed to be performed. Using our profile location constraints can be defined at the design time of a workflow schema, but it is also possible to define rules to automatically generate constraints during the runtime of a workflow instance.

Ideas for future work include to introduce further stereotypes to express mobile-specific constraints, e.g., to state minimum capabilities for the mobile computers to perform the activity, for example a minimum screen size or the availability of tamper-proof memory. Further, we envision the development of a graphical editor that supports drawing UML diagrams according to our profile; this editor should integrate functionalities to work with geographic data to define the spatial extents of location constraints.

Further, it would also be worthwhile to extend the location model with security labels in the sense of mandatory access control (MAC). To the best of our knowledge there are only two publication by other authors which deal with MAC-based location-aware access control [24][25]. However, these publications do not cover workflow-specific aspects. We also published an article on MAC-based location constraints for database management system [26], but again this work isn't workflow-specific. The basic idea of introducting security labels into our modelling approach would be to assign clearance levels to location, e.g., a particular country or building could have the clearance *Top Secret*, while other locations have only the clearance for the level *Secret*. Based on this classification location constraints could be assigned to a UML activity diagram. Such a constraint could state that a particular activity can only be performed when the mobile actors stays at a location that has a clearance of *Top Secret*, but not below; or a location rule could be used to assign a derived constraint so that the target activity's location has at a clearance level not lower than the location of the trigger activity.

In Section VIII several types of anomalies where discussed that can be found in activity diagrams when location constraints are applied. Since models should be free of such anomalies we are working towards an algorithmic approach to automatically detect such anomalies.

UML activity diagrams are by far not the only graphical language for the modelling of workflows. Another popular language for workflow modelling is the *Business Process Modelling Notation (BPMN)*, which is also maintained by the OMG [27]. We are also working on a BPMN profile for the expression of location constraints similar to the profile presented in this article [28].

Further, we are currently working on the application of the concept of location constraints for business processes from the domain of agriculture [29].

## References

[1] M. Decker, "Modelling Location-Aware Access Control Constraints for Mobile Workflows with UML Activity Diagrams," in *The Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UbiComm 2009)*. Sliema, Malta: IEEE, October 2009, pp. 263–268.

[2] A. Oberweis, *Process-Aware Information Systems. Bridging People and Software Through Process Technology*. New York, USA, et al.: John Wiley & Sons, 2005, ch. Person-to-Application Processes: Workflow Management, pp. 21–36.

[3] M. Perry, K. O'Hara, A. Sellen, B. A. T. Brown, and R. H. R. Harper, "Dealing with mobility: understanding access anytime, anywhere," *ACM Transactions on Computer-Human Interaction*, vol. 8, no. 4, pp. 323–347, December 2001.

[4] M. Decker, "A Security Model for Mobile Processes," in *Proceedings of the International Conference on Mobile Business (ICMB 08)*. Barcelona, Spain: IEEE, July 2008.

[5] A. Küpper, *Location-based Services – Fundamentals and Operation*. Chichester, U.K.: John Wiley & Sons, 2007, reprint.

[6] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[7] M. Decker, "Location-Aware Access Control: An Overview," in *Proceedings of Informatics 2009 — Special Session on Wireless Applications and Computing (WAC '09)*, Carvoeiro, Portugal, 2009, pp. 75–82.

[8] *Unified Modeling Language (OMG UML), Superstructure, V2.1.2*, Object Management Group, 2007.

[9] R. Lake, D. S. Burggraf, M. Trninic, and L. Rae, *GML. Geography Mark-Up Language. Foundation for the Geo-Web*. Chichester, U.K.: John Wiley & Sons, 2004.

[10] D. S. Burggraf, "Geography Markup Language," *Data Science Journal*, vol. 5, pp. 178–204, October 2006.

[11] E. Bertino, E. Ferrari, and V. Atluri, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems," *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 65–104, 1999.

[12] R. S. Sandhu, "Separation of duties in computerized information systems." in *Results of the IFIP WG 11.3 Workshop on Database Security (DBSec)*, Halifax, U.K., 1990, pp. 179–190.

[13] J. B. Warmer and A. Kleppe, *The object constraint language: getting your models ready for MDA*, 2nd ed., ser. The Addison-Wesley Object Technology Series. Boston, Massachusetts, USA: Addison-Wesley, 2003.

[14] A. Rodriguez, E. Fernández-Medina, and M. Piattini, "Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes," in *Third International Conference on Trust and Privacy in Digital Business (TrustBus)*, Krakow, Poland, 2006, pp. 51–61.

[15] V. Stefanov, B. List, and B. Korherr, "Extending UML 2 Activity Diagrams with Business Intelligence Objects," in *Proceedings of Data Warehosuing and Knowledge Discovery (DaWaK 2005)*, Copenhagen, Denmark, 2005, pp. 53–63.

[16] H. Baumeister, N. Koch, P. Kosiuczenko, and M. Wirsing, "Extending Activity Diagrams to Model Mobile Systems," in *Proceedings of NetObjectDays (NOD)*, Erfurt, Germany, 2002, pp. 278–293.

[17] D. F. Ferraiolo, R. Sandhu, E. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[18] M. L. Damiani, E. Bertino, and P. Perlasca, "Data Security in Location-Aware Applications: An Approach Based on RBAC," *International Journal of Information and Computer Security*, vol. 1, no. 1/2, pp. 5–38, 2007.

[19] S. M. Chandran and J. Joshi, "LoT-RBAC: A Location and Time-Based RBAC Model," in *Proceedings of the 6th International Conference on Web Information Systems Engineering (WISE '05)*. New York, USA: Springer, 2005, pp. 361–375.

[20] F. Hansen and V. Oleshchuk, "SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems," in *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORDSEC)*. Gjovik, Norway: NTNU, 2003, pp. 129–141.

[21] M. Decker, "Prevention of Location-Spoofing. A Survey on Different Methods to Prevent the Manipulation of Locating-Technologies," in *Proceedings of the International Conference on e-Business (ICE-B)*. Milan, Italy: INSTICC, 2009, pp. 109–114.

[22] R. Hewett and P. Kijsanayothin, "Location contexts in role-based security policy enforcement," in *Proceedings of the 2009 International Conference on Security and Management (SAM'09)*, Las Vegas, Nevada, USA, 2009, pp. 404–410.

[23] M. Decker, "An Access-Control Model for Mobile Computing with Spatial Constraints - Location-aware Role-based Access Control with a Method for Consistency Checks," in *Proceedings of the International Conference on e-Business (ICE-B 2008)*. Porto, Portugal: INSTICC, July 2008, pp. 185–190.

[24] U. Leonhardt and J. Magee, "Security Considerations for a Distributed Location Service," *Journal of Networks and Systems*, vol. 6, no. 1, pp. 51–70, 1998.

[25] I. Ray and M. Kumar, "Towards a Location-based Mandatory Access Control Model," *Computers & Security*, vol. 25, no. 1, pp. 36–44, 2006.

[26] M. Decker, "Mandatory and Location-Aware Access Control for Relational Databases," in *Proceedings of the International Conference on Communication Infrastructure, Systems and Applications in Europe (EuropeComm 2009)*, ser. LNICST, R. M. et al., Ed., no. 16. London, U.K.: Springer, August 2009, pp. 217–228.

[27] OMG, *Business Process Model and Notation (BPMN) v. 1.2*, Object Management Group, January 2007.

[28] M. Decker, H. Che, A. Oberweis, P. Stürzel, and M. Vogel, "Modeling Mobile Workflows with BPMN," in *Proceedings of the Ninth International Conference on Mobile Business (ICMB 2010)/Ninth Global Mobility Roundtable (GMR 2010)*. Athens, Greece: IEEE, 2010, pp. 272–279.

[29] M. Decker, D. Eichhorn, E. Georgiew, A. Oberweis, J. Plaßmann, T. Steckel, and P. Stürzel, "Modelling and enforcement of location constraints in an agricultural application scenario," in *Proceedings of the Conference on Wireless Applications and Computing 2010 (WAC 2010)*, Freiburg, Germany, 2010, accepted, to appear.

# Network Prediction for Energy-Aware Transmission in Mobile Applications

Ramya Sri Kalyanaraman
Helsinki Institute for Information Technology HIIT
P.O. Box 15400, FI - 00076 Aalto
Espoo, Finland
ramya@hiit.fi

Yu Xiao, Antti Ylä-Jääski
Aalto University
P.O. Box 15400, FI-00076 Aalto
Espoo, Finland
{yu.xiao, antti.yla-jaaski}@tkk.fi

*Abstract*— **Network parameters such as signal-to-noise-ratio (SNR), throughput, and packet loss rate can be used for measuring the wireless network performance which highly depends on the wireless network conditions. Previous works on energy consumption have shown that the performance of wireless networks have impact on the energy efficiency of data transmission. Hence, it is potential to gain energy savings by adapting the data transmission to the changing network conditions. This adaptation requires accurate and energy-efficient prediction of the network performance parameters. In this paper, we focus on the prediction of SNR and the prediction-based network adaptations for energy savings. Based on the SNR data sets collected from diverse real-life networks, we first evaluate three prediction algorithms, namely, Autoregressive Integrated Moving Average, Newton Forward Interpolation, and Markov Chain. We compare these three algorithms in terms of prediction accuracy and energy overhead. Later we propose a threshold-based adaptive policy which controls the data transmission based on the predicted SNR values. To evaluate the effectiveness of using network prediction in adaptation, we use a FTP as a case study and compare the network goodput and energy consumption under different network conditions. The experimental results show that the usage of adaptations improves the network goodput. Furthermore, the adaptations using prediction can save up to 40% energy under specific network conditions when compared to the adaptation without prediction.**

**Keywords-prediction; adaptation; SNR; power; context-awareness; policy-based.**

## I. INTRODUCTION

Mobile hand-held devices are more and more used for accessing rich multimedia content and various social media services. These new applications and services often call for more transmission capacity, processing power and high-quality displays which increase the energy usage. The improved user experience might be compromised by the short battery lifetime of the device. Unfortunately the improvements in battery technology are rather slow that there is a strong motivation to invest in software techniques to save the energy usage of mobile devices.

In network applications, major part of the energy is consumed by data transmission. The cost incurred during data transmission is not only dependent on the amount of data being transferred, but also the network goodput [2]. There is an indirect connection between the network goodput and other network performance parameters such as signal-to-noise-ratio (SNR). For example, in the wireless networks with higher SNR, the network goodput is usually higher, and therefore the energy consumption per a unit of transmitted data is lower. Hence, it is possible to save energy by transferring data only under the conditions where SNR values are high.

In this paper we focus on the prediction of SNR and its potential for improving the efficiency of an energy-aware network adaptation. With the help of the predicted future SNR values, it is possible to make a priori decisions as to when to schedule the data transmission over a wireless link. We propose a general solution of the prediction-based adaptive network transmission, including prediction algorithms of SNR and an adaptive policy. We evaluate our solution using FTP as a case study, while our solution is scalable to other network applications such as real-time streaming by taking the application-specific performance constraint into account in our adaptive policy.

Although the prediction of network parameters [3, 4] and their use in context-aware mobile applications have been discussed in the literature [5, 6, 7, 8, 9, 10], using the predicted network parameters for energy efficiency has not been widely studied, especially the usage of SNR for adaptive network transmission. Our contributions include the following aspects.

a) Comparing the performance of the prediction algorithms, namely, Auto Regressive Integrated Moving Average (ARIMA), Newton's Forward Interpolation (NFI) and Markov chain (MC). We analyze the performance in terms of prediction accuracy, processing load and energy overhead.
b) Proposing and evaluating a prediction-based power adaptation. The results show that the potential in energy saving under favorable network conditions can be up to 40% with the help of the predicted SNR.

This work is an extended version of our earlier publication [1]. Major portions of the experimental results have been updated, with enhancement in the offline training and online computation. In this work, we refine the evaluation by classifying the network conditions and analyzing the effectiveness of energy savings under different network conditions. In addition, the Linear Regression method used in [1] is replaced by MC, as MC shows better performance in short-term prediction.

The remainder of this paper is structured as follows. Section 2 reviews the related work in network prediction and

prediction-based adaptations. The prediction algorithms used in this study are introduced in Section 3, and the model fitting using these algorithms is described in Section 4. Section 5 describes a prediction-based adaptation. In addition, Section 5 includes the evaluation of the adaptation using FTP as a case study. In Section 6, we discuss the obtained experimental results. Finally, we present the concluding remarks and potential future work in Section 7.

## II. RELATED WORK

Adaptive mobile applications are aware of and able to adapt to the changes in context such as the network signal strength, residual battery lifetime, and the user's location. The key for providing such adaptive applications lies in the decision-making of adaptations based on the available context data and predefined policies.

The context information includes the past, present, and future states of the contexts. Obviously, the past and present states of the contexts can be collected during runtime, whereas the future states are not available and have to be predicted based on the past and/or present states. The predicted states of the contexts are often used in context-aware applications for enhancing the effectiveness of adaptations, such as improving the resource utility and system performance. In [3] prediction of long range wireless signal strength is presented, where prediction is utilized for efficient use of resources in dynamic route selection in multi-hop networks.

For network-aware adaptive applications, network conditions are the most important contexts and can be described using signal strength, SNR, goodput, bit error rate and other contexts that reflect the status of the network transmissions. The so-called Box-Jenkins approach is the most widely used technique for network prediction. The basic idea is to train models with different parameters to fit the pre-collected data sets. This approach is adopted by many linear and non-linear models, such as ARIMA [13] and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) [16]. In addition, Linear Regression [3], the MC [4], the Hidden Markov Model [7], and the Artificial Neural Network [17] have also been proposed for network prediction. In this paper, we apply ARIMA, MC and NFI [14] to the prediction of network SNR.

As mentioned before, network conditions can be described using various parameters. Different prediction algorithms might fit different contexts better depending on the characteristics of the contexts and the algorithms themselves. Previous researchers have been trying to apply different prediction algorithms to the prediction of different contexts. For example, [4] and [20] predict the status of network connectivity and the future location of mobile device using the MC, [3] utilized a linear regression approach for long range signal strength prediction, and [21] used the Hidden Markov Model to predict the number of wireless devices connected to an access point at a given time. Link prediction and path analysis using Markov chains in the World Wide Web are presented in [22].

Different prediction techniques are compared in terms of k-step-ahead predictability and performance overhead. For example, one-step-ahead predictability can be measured by Mean Square Error (MSE), Normal Mean Square Error (NMSE)[3], Root Mean Square Error (RMSE) [11] and Signal-to-Error Ratio (SER). The performance overhead includes the computational cost such as CPU cycles and memory access counts, and also the energy consumption caused by the prediction itself.

In most of the research work focusing on prediction for context-aware and adaptive applications, the motivation lies in increasing and analyzing the prediction accuracy of various prediction methods. In [11] various prediction methods such as Neural networks and Bayesian networks, state and Markov predictors are modeled for performing the next location prediction using the sequences of previously visited locations. In this paper, prediction accuracy has been considered as the main indicator for benchmarking the performance of various prediction techniques. To the best of our knowledge, our paper is the first one presenting the energy overhead of different prediction algorithms.

Adaptive applications purely based on policies consist of the condition and action pairs for different contexts. Here the possibility of anticipating the future trend of the context is not utilized. In adaptive applications with the support of prediction based adaptation, the system is given a chance to know the future trend of the context, and act accordingly.

Most of the existing adaptive network applications make adaptive decisions according to predefined policies. Predicted network contexts can be used as conditions of adaptive policies. In [18], the feasibility of implementing policy-based network adaptations in middleware architecture is presented. However, prediction-based network adaptations emphasizing possible energy conservation have not been widely studied. In this paper, we propose an adaptive file download in WLAN based on the prediction of network SNR, and compare the efficiency of prediction and adaptations among three different prediction techniques.

## III. NETWORK PREDICTION ALGORITHMS

Dynamic adaptation to the environment and proactive behavior are key requirements for mobile adaptive applications. Prediction algorithms play a major role in providing a proactive nature for such applications [5]. In this paper we focus on predicting the network SNR that will be used for adaptive network applications.

In our approach, the nature of input data we observe to make the network prediction reflects the time series pattern [11]. We choose ARIMA [12], the most general model for forecasting a time series pattern. Selecting NFI [14] allows us to fit non-linear data over the curve. In addition, we experiment with MC representing a discrete time stochastic process. While evaluating the best suiting algorithm for our application, a trade-off is made between prediction accuracy, processing load and energy consumption, instead of compromising between the highest accuracy and the lowest energy consumption alone.

### A. Auto Regressive Integrate Moving Average(ARIMA)

Classical linear time series models are based on the idea that the current value of the series can be explained as a function of the past values of the series and some other independent variables. ARIMA is one of the most famous linear predictive models used for network prediction. It is an integration of three models, namely, an autoregressive model of order *p*, a moving average model of order *q*, and a differencing model of order *d*. According to the definition in [18], a process, $x_t$, is said to be ARIMA(p, d, q) if

$$\varphi(B)(1-B)^d x_t = \theta(B) w_t \,, \tag{1}$$

where $B$ is a backward shift operator, $\Phi(B)$ is an autoregressive operator, $\theta(B)$is a moving average operator, and $w_t$ is assumed to be a Gaussian white noise. The three operators can be expressed as below.

$$B^d x_t = x_{t-d}, \tag{2}$$
$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_p B^p \big(\varphi_p \neq 0\big), \tag{3}$$
$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q \big(\theta_q \neq 0\big), \tag{4}$$

where $\varphi_1 \cdots \varphi_p$ and $\theta_1 \cdots \theta_q$ are constants.

Let $\hat{Y}(t)$ be the predicted SNR value at time *t*, and *Y(t-i)* be the observed SNR value at time $(t-i)$. The time series of SNR is considered to be an ARIMA (p, d, q) model if

$$\hat{Y}(t) = \mu + \sum_{i=1}^{d} Y(t-i) + \varphi \sum_{i=1}^{p} Y(t-i) - \theta \sum_{i=1}^{q} e(t-i)\,, \tag{5}$$

where $e(t-i)$ is the residual of prediction at time$(t-i)$, and $\mu$ is the intercept.

The values of p, d, q are identified based on the analysis of Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), while the estimation of parameters $\theta$, $\varphi$, $\mu$ are based on the least squares method which aims at minimizing the MSE of residuals as shown in Equation (6). The procedure of model fitting will be detailed in Section 4.

$$MSE = E(Y(t) - \hat{Y}(t))^2, \tag{6}$$

### B. Newton's Forward Interpolation(NFI)

The observation at time *x* in a time series is defined as a function $f(x)$, and the values of *x* are tabulated at interval *h* as shown in Equation (4).

$$x = x_0 + i \times h, \tag{7}$$

where the first value of *x* is $x_0$, and *i* is the number of intervals between *x* and $x_0$. When only a few discrete values of $f(x), i = 0, 1, 2, \ldots$ are known, NFI aims at finding the general form of $f(x)$ based on known values by using the finite difference formulas.

Let $f_i \equiv f(x) \equiv f(x_0 + i * h)$, and the finite forward difference of a function $f_i$ is defined by $\Delta f_i \equiv f_{i+1} - f_i$. Higher orders, such as the $j^{th} (j \leq i)$ order forward difference $\Delta^j f_i$, can be obtained by repeating the operations of the forward difference operator *j* times. For example,

$$\Delta f_0 = f_1 - f_0, \Delta^2 f_0 = \Delta(\Delta f_0) = f_2 - 2f_1 + f_0, \quad \Delta^3 f_0 = f_3 - 3f_2 + 3f_1 - f_0. \tag{8}$$

NFI model fits the observation at time *x* with an $i^{th}$ degree polynomial as Equation (9).

$$f(x) = f_0 + (x - x_0)\frac{\Delta f_0}{h} + \frac{1}{2!}(x - x_0)(x - x_1)\frac{\Delta^2 f_0}{h^2} + \cdots + \frac{1}{(i-1)!}(x - x_0)(x - x_1)\ldots(x - x_{i-2})\frac{\Delta^{(i-1)} f_0}{h^{(i-1)}} + \frac{1}{i!}(x - x_0)(x - x_1)\ldots(x - x_{i-1})\frac{\Delta^i f_0}{h^i}. \tag{9}$$

When only the past *i* SNR observations are known, up to $(i-1)^{th}$ order forward difference can be developed. The prediction of SNR at time *x*, corresponding to the $(i+1)^{th}$ observation, is defined as a function *g(x)* in Equation(10).

$$g(x) = f_0 + (x - x_0)\frac{\Delta f_0}{h} + \frac{1}{2!}(x - x_0)(x - x_1)\frac{\Delta^2 f_0}{h^2} + \frac{1}{3!}(x - x_0)(x - x_1)(x - x_2)\frac{\Delta^3 f_0}{h^3} + \cdots + \frac{1}{(i-1)!}(x - x_0)(x - x_1)\ldots(x - x_{i-2})\frac{\Delta^{(i-1)} f_0}{h^{(i-1)}}. \tag{10}$$

Compared to *f(x)*, the error term *e(x)* is approximated as Equation (11).

$$e(x) \cong f(x) - g(x) \cong \frac{1}{N!}(x - x_0)(x - x_1)\ldots(x - x_{N-1})\frac{\Delta^N f_0}{h^N}. \tag{11}$$

Different from ARIMA, NFI could predict future values online directly based on past observations without offline training. Since the prediction accuracy of NFI depends on the size of N, the size of N can be selected by applying the least square method to the training data sets as used in ARIMA model fitting. The offline training and online implementation are described in Section 4.

### C. Markov Chain(MC)

MC is a discrete time stochastic process with the Markov property, where given the present state, the future and past states are independent. To determine the future state from the present state, a probabilistic approach is used. A state space is defined where all the possible states of the system are represented. The change of state from one to the other is called state transition. Let us consider a discrete time process, $X_n$: n>0 with discrete state spaces {i, j}. The transition probability from i to j is computed as given in Equation (12).

$$P_{ij} = P(X_{n+1} = j \mid X_{n=i}). \tag{12}$$

The state transition involved in our model, and the online prediction of the state transition is explained in Section 4.

## IV. NETWORK PREDICTION

Prediction of network related information adds value to ubiquitous applications, as the need for dynamic adaptation in a fluctuating network environment becomes a major requirement in such cases. In addition to network traffic and mobility prediction, predicting the value of SNR itself could provide a timely hint about possible changes in network conditions, and thus the application can prepare itself for adaptation.

In this section, we present the steps involved in offline training and online computation of prediction algorithms. The workflow is described in Figure 1. To perform offline training we carry out data collection. The spots chosen for data collection involve physical interferences and disturbances. Offline training enables us to estimate the parameters required for online computation, and the performance of prediction algorithms can also be analyzed using offline training. In online computation, the prediction algorithms are applied during runtime and the outcomes are matched with the adaptation policy.

### A. Data Collection

The offline training used in our methodology significantly depends on the data collection. The data sets were gathered considering the following factors.

a) The user walked around a defined path, where the path involved different types of physical obstacles, interferences and noise.
b) A total of 12 data traces were collected at different times during three days, where the collection of each set lasted for 17 minutes.
c) The user walked around at a normal walking speed, and the signal strength and the noise level were recorded for every 100 millisecond.

We performed the data collection in the Computer Science department building of the university. Figure 2 shows the route map of the data collection. The defined path covered the first and second floors of the Computer Science department, which includes the corridor of the Data Communications Software lab, department library, café and the stairs to the second floor. To cover different network environments, we chose the path to include different types of physical obstacles, and interference. For example, the department corridor consists of physical obstacles such as wall structures and glass doors, and the library has metallic book shelves, and wooden structures. The open area in the cafe remains as a place with possible interferences due to multiple access points, and the significant presence of Bluetooth enabled devices.

### B. Data Analysis and Pre-processing

By observing the collected data sets, we noticed that the SNR values remain unchanged for several milliseconds as shown in Figure 3. We therefore re-sampled the collected data sets at 1Hz, and used the new data sets for further processing. We chose 80% of the samples as the training data set, and the others for testing.

Data smoothing is often used for figuring out future trends, such as the trend in marketing size or stock prices. However, for short-term prediction like SNR prediction in our case, it remained an interesting question whether data smoothing still helped to improve the prediction accuracy. We therefore applied data smoothing over the training data set, and compared the prediction accuracy between the models using a smoothed data set and the original training data set.

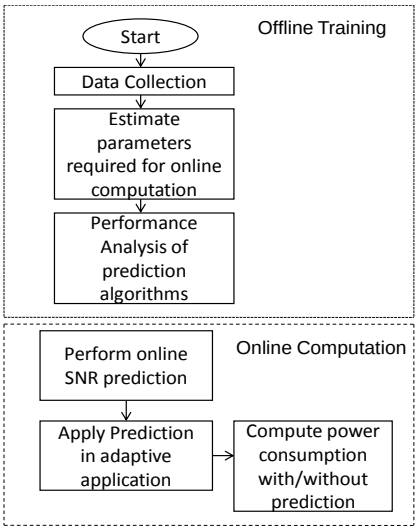



Figure 1: Workflow of offline training and online computation

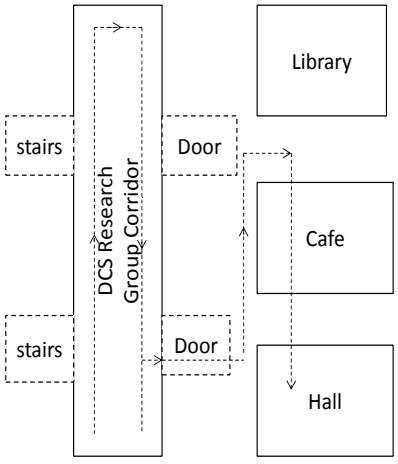



Figure 2: Route Map for Data Collection

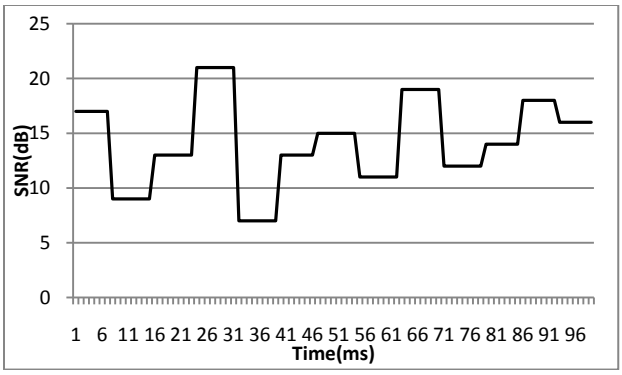



Figure 3: Samples of SNR values collected at 10Hz.

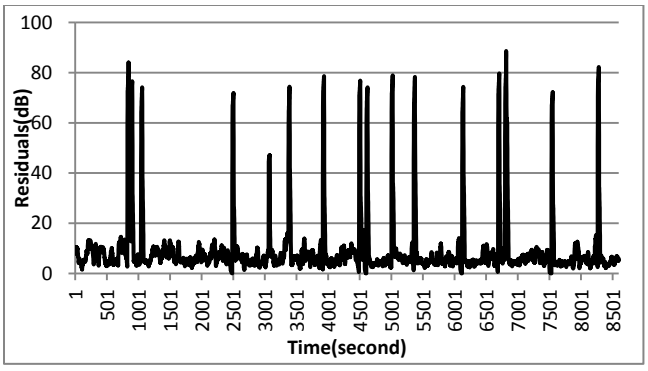



Figure 4: Residuals of smoothing with window size 20.

Table 1: Comparison of prediction accuracy among different models.

| | MSE | NMSE | SER |
|---|---|---|---|
| ARIMA(0, 1,1) | 97.58624 | 0.425374 | 7.456893 |
| ARIMA(1, 0,1) | 89.26294 | 0.389093 | 7.844067 |
| ARIMA(2, 1,0) | 174.4522 | 0.760430 | 4.934015 |
| NFI(N=3) | 692.8996 | 3.015758 | -1.0487 |
| NFI(N=4, 5) | 99.3080 | 0.432879 | 7.380937 |

To perform data smoothing, we apply the simple Moving Average algorithm [23] which is used for time series data set smoothing in the field of statistics. According to the moving average algorithm, from the collected *N* data points, we perform averaging for every *W* data set, where *W* is often termed as the window size. The general expression for the moving average is given as in Equation (13).

$$M_t = \frac{[X_t + X_{t-1} + \cdots X_{t-w+1}]}{w}. \quad (13)$$

In general as the window size increases the trend of the smoothed data set becomes clearer, but with a risk of a shift in the function. To choose an optimal window size we performed smoothing with a varied window size, and selected the best one with the minimum MSE of the residuals. The residuals are the differences between the original values and smoothed values. According to our experimental results, smoothing with a window size of 20 has the minimum MSE which is 9.5 as shown in Figure 4.

*C. Offline training/Model Fitting*

*1) ARIMA*

ARIMA offline training is to fit an ARIMA(p, d, q) model to the collected data sets. The outputs of the model fitting includes the orders of autoregressive, differencing, and moving average, *p*, *d*, *q*, and the estimated parameter values of the autoregressive operator, the moving average operator, and the intercept, $\varphi, \theta, \mu$, as defined in Section 3.1. The model fitting includes the following five basic steps [18].

*a) Data transform.* We plotted the data and observed the possible data transform. As introduced in Section 4.1, we applied smoothing to the raw data, and trained the models using both data sets and comparing the results.

*b) Model identification.* The orders of autoregressive, differencing and moving average are identified by observing the ACF and PACF of the residuals. For orginal data sets, both ARIMA(0, 1, 1) and ARIMA(1, 0, 1) fit the data well and the differences in terms of the ACF and PACF of the residuals are negligible. Hence we trained both models and chose the better one after diagnostics. Similarly, for smoothed data sets with window size 20, ARIMA(2, 1, 0) fits better than other models for smoothed data sets.

*c) Parameter estimation.*We estimated the parameters of the expected models which obtain the minimum MSE.

*d) Diagnostics.*We applied the models obtained from the last step to the testing data sets and compared the accuracy among the models. First, we calculated the MSE, NMSE, and SER for each model. NMSE is a function of the MSE normalized by the variance of the actual data as defined in Equation (17).

$$NMSE = \frac{MSE}{E[(E[Y(t)] - Y(t))^2]}. \quad (17)$$

SER is defined as in Equation (18).

$$SER = 10log_{10}\left(\frac{E[Y(t)^2]}{E[(Y(t) - \hat{Y}(t))^2]}\right). \quad (18)$$

The smaller the MSE and NMSE are, the higher the accuracy is. Conversely, the bigger the SER is, the higher the accuracy is. The results of MSE, NMSE and SER for the three ARIMA models are listed in Table 1.

*e) Model Selection.* According to Table 1, ARIMA(2, 1, 0), with training based on the smoothed data set, showed the lowest accuracy compared to the models obtained from original data sets. ARIMA(1, 0, 1) fits the testing data sets better than others due to the smallest MSE and NMSE, as well as the highest SER. Hence we chose ARIMA(1, 0, 1) which is shown in Equation (14) for online SNR prediction.

$$\hat{Y}(t) = 22.1650 + 0.7997 * (Y(t-1) - 22.1650) - 0.0052 * e(t-1). \quad (14)$$

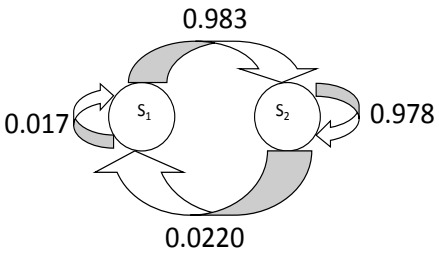



Figure 5: Transition probability between States, $S_1$ and $S_2$.

The implementation and online performance evaluation is presented in Section 4.3.

*2) NFI*

The accuracy of NFI prediction depends on the order of the forward difference *N* as explained in Section 3.3. We chose the size of N based on the least square method as used for ARIMA model fitting. As shown in Table 1, we compared the MSE when N is set to 3, 4, and 5 respectively. When N is 3, the MSE is much bigger than others, while the same predicted values and also the same MSE, NMSE, and SER values are obtained when either N= 4 or 5. Hence, we selected the order of the forward difference as 4 for online prediction.

*3) Markov Chain*

We designed a two-state Markov Chain model as shown in Figure 5, where $S_1$ and $S_2$ are the two possible states. Here $S_1$ represents the SNR value less than or equal to the threshold, and $S_2$ corresponds to a SNR greater than the threshold. Either $S_1$ or $S_2$ can represent the initial state depending upon the value of the SNR at the start of application.

There are four possible state transitions involved in the state machine, with four different transition probabilities. Let $S_t$ and $S_{t+1}$ represent the state of SNR at time t, and t+1. The four probable state transitions are as given below:

$$\begin{aligned}
SNR_{low,\,low}(t) &= Pr\{S_{t+1} = low \mid S_t = low\},\\
SNR_{low,\,high}(t) &= Pr\{S_{t+1} = low \mid S_t = high\},\\
SNR_{high,\,high}(t) &= Pr\{S_{t+1} = high \mid S_t = high\},\\
SNR_{high,\,low}(t) &= Pr\{S_{t+1} = high \mid S_t = low\}.
\end{aligned}$$

The values of the four probabilities are obtained from the training of the data sets. During the training, we set a threshold for discreetizing the samples into low and high states to 19, and calculated the probability distribution. The values of the four probabilities are shown in Figure 5. The precision of predicted outcomes using the Markov chain is 0.8191 as calculated according to Equation (19).

$$Precision = \frac{Number\ of\ correctly\ predicted\ values}{Total\ number\ of\ predicted\ values}. \qquad (19)$$

*D. Online prediction*

We present the online SNR prediction with the same monitoring and prediction frequencies in this Section, and set the initial values of the frequencies to 1Hz.

*1) ARIMA(1, 0, 1)*

The monitored and predicted SNR values are recorded in arrays s[] and p[] respectively. The pseudo code of ARIMA (1, 0, 1) online prediction is described as below.

```
//initialize the parameters of ARIMA(1,0,1);
Set ar=0.7997, ma=-0.0052, intercept=22.1650;
Initialize s[], p[] to 0;
i=0;
Repeat every 1 second{
Monitor SNR and write into s[i];

if (i>0){
        err = s[i-1]-p[i-1];
        p[i]=intercept + ar*(s[i-1]-intercept)
                +ma*err;
}
```

*2) NFI*

Set the order of forward difference to be 4. The monitored and predicted SNR values are recorded in array s[] and p[] respectively. The 1st, 2nd, and 3rd order of forward differences are calculated during runtime, and are written into arrays delta[], delta2[], and delta3[] respectively. The pseudo code of the online NFI prediction is described as below.

```
Initialize delta[], delta2[],delta3[],p[], s[] to 0;
i=0;
Repeat every 1 second{

Monitor SNR and write into s[i];

If (i>0)
        delta[i]=s[i]-s[i-1];
If (i>1)
        delta2[i]=delta[i]-delta[i-1];
If (i>2)
        delta3[i]=delta2[i]-delta2[i-1];

If (i>3)
    p[i]=s[i-4]+3*delta[i-3]+3*delta2[i-2]+delta3[i-
1];

        i++;
}
```

*3) Markov Chain*

The Markov chain SNR predictor computes the probable next state ($S_{t+1}$), given the current state ($S_t$) of SNR, according to the probability of different transitions obtained

Table 2: Comparison of performance and power overhead among different models.

| | ARIMA | NFI | MC |
|---|---|---|---|
| CPU_CYCLES:100000 | 445 | 432 | 623 |
| DCACHE_ACCESS_ALL:100000 | 31 | 25 | 73 |
| Power(W) | 0.365 | 0.366 | 0.474 |

Table 3. Classifications of the network conditions.

| | |
|---|---|
| 1 | SNR mean is smaller than 15; SNR threshold is set to 15. |
| 2 | SNR mean is between 15 and 20; SNR threshold is set to 15 and 20. |
| 3 | SNR mean is bigger than 20, and standard deviation of SNR is smaller than 5. SNR threshold is set to 20. |
| 4 | SNR mean is bigger than 20, and standard deviation of SNR is not smaller than 5. SNR threshold is set to 15 and 20. |

from offline training. The pseudo code for predicting the transition probability of the future state of SNR is given as follows:

```
//Monitor SNR and discreetize the received value.
Get current_SNR;

if (current_SNR <= Threshold)
        Current_state = S1;
else current_state = S2;

//Determine next state. SNRt[]is the transition matrix
obtained from offline training. SNRt[0] and SNRt[1]
represents the probability of the state transition from S1 to
S2 and vice versa .
Generate a random number, X.
if (current_state == S1)
{
        if (X ≤ SNRt[0])
                next_state = S1;
        else next_state = S2;
}
else
{
        if (X ≤ SNRt[1]
                next_state = S2;
        else next_state = S1;
}
```

### D. Model Evaluation

We measured the overhead of online SNR prediction in terms of CPU cycle count, data cache access rate and power consumption. We ran oprofile [26] on the Nokia N810 while running the online prediction algorithms for 1 minute, and monitored the events, namely, CPU_CYCLES and DCACHE_ACCESS_ALL. The CPU cycle count and data cache access counts caused by the SNR prediction are listed in Table 2.

We used Nokia power measurement software to measure the power consumption during runtime. During measurements the display was turned off and the WLAN interface was turned on. The average power is 0.361 W when the system is idle and the power saving mode of WLAN is turned on, and 0.362 W when SNR monitoring is running at 1Hz at the same time. We measured the average power consumption when different online SNR predictors were running, and the comparisons are listed in Table 2.The energy overhead is calculated as the difference between the power consumption when the prediction algorithms are running and the power consumption of the device when it is in IDLE state. The power overhead caused by online prediction is between 0.004W and 0.113W depending on the prediction algorithm used.

## V. PREDICTION-BASED NETWORK ADAPTATIONS

Wireless transmission is considered to cost much more energy than local processing on mobile devices [15], and the energy consumed by wireless transmission varies with the performance of the wireless networks. The performance is highly dependent on the network conditions. Generally, wireless transmission under better network conditions is more energy-efficient. Hence, adaptive mobile applications are expected to adapt wireless transmission that the transmission can be conducted under relatively good network conditions in order to save energy. In this section, we describe an application performing power adaptation using network prediction.

### A. Overview

Based on the comparison and analysis of prediction algorithms made in Section 4 we are able to distinguish between the pros and cons of using different prediction methods, especially in terms of prediction accuracy, performance and energy overhead. After performing the offline training, and the analysis of different prediction methods we continue forward to the second phase of network prediction, online computation. In this section we present a real time user scenario that demonstrates the need for adaptation in ubiquitous applications. Realizing the requirements for network adaptation from the presented scenario we apply network prediction in the application.

Let us consider a practical scenario showcasing the need for prediction enabled adaptive applications.

*Alice's mobile phone is connected to a WLAN network and she has been in the process of downloading a file. The network quality associated with the device fluctuates between poor to strong. The application with the help of prediction foresees the status of network, and hence chooses an adaptive action accordingly in order to gain the optimal power saving. The adaptive action can be pausing, stopping, or continuing the file download.*

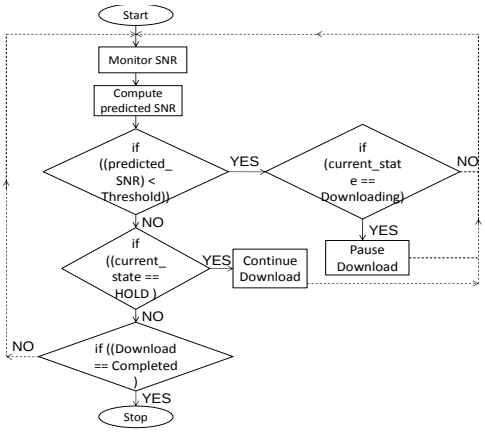



Figure 6: Flow chart representing online adaptation

### B. Adaptation Policy

Adaptation policy defines the actions to be performed based on the changing context data. Adaptation is triggered in mobile applications as the outcome of the prediction is matched with a simple predefined adaptation policy. The adaptation policy can be defined by either the application developer or end-user. In the case of the end-user, policies can be in the form of user preferences for adaptive applications. The priority and conflict management between prediction results and current network condition however are outside the scope of this paper.

In the file download scenario, we designed an adaptive policy which enables the application to pause or continue the file download depending on the predicted future SNR. The basic idea is to transfer the data only when the SNR is above a threshold. We classify the network condition into four types using the mean and standard deviation of SNR as described in Table 3. The threshold value is set depending on the type of the network condition. It is described as below.

*if (predicted_SNR < threshold) && (current_state == downloading)*
*Pause download;*
*else if ((predicted_SNR ⩾ threshold) && (current_state == hold))*
*Continue download;*
*else*
*// Continue the same action state.*
*Continue download | Pause download;*

While applying the above mentioned adaptive policy for the applications such as real-time streaming which is not delay-tolerant, the settings of the threshold value must take the tolerant delay into account. For example, the delay caused by the pause operation can be predicted based on the predicted SNR value. Only when the delay is tolerable, the download can be paused. The prediction based adaptation continues till the file download is completed. Figure 6 describes the online adaptation performed in the file download scenario using the online prediction and defined policy. The implementation for online computation of the prediction forecast is combined with the adaptation code. Thus the prediction outcome is directly used for choosing the adaptation policy, and executes an adaptive operation in the file download application.

### C. Experimental results

The prediction-based power adaptation is evaluated in terms of energy consumption, network goodput, and prediction accuracy, under different network conditions. For different network conditions, we test the adaptations with an SNR threshold of 15 and/or 20 with an exception for the MC method. For the MC method we set the threshold to be 19, as the offline training and online computation has to follow the same threshold value. Hence, our definitions of test cases include the network conditions in terms of the mean and standard deviation of SNR values, SNR threshold value, and possible network goodput range. Though the MC method follows a different threshold, the measurements obtained using the MC as the prediction method can still be accommodated within the test cases described. Based on the analysis of our experimental network conditions, we divide our test cases into four scenarios as listed in Table 3.

The adaptation code is implemented using C language. The power measurements for the adaptive file download with and without prediction were carried out on a public 802.11g network in our campus with the power saving mode on. A TCP client *wget* (http://www.gnu.org/software/wget/) running on the N810 downloaded a MySQL software package from a remote file server [26]. The file name is MySQL-shared-compat-5.1.42-0.rhel3.x86_64.rpm with a size of 5072982 Bytes. We set both the prediction and monitoring frequency to 1Hz and performed one-step-ahead prediction. Adaptation is performed for each prediction output, which means the adaptation policy is executed every one second.

We measured the power consumption in different scenarios without adaptation, and used the values as base line for comparison. The power measurement results are listed in Table 4. Since the energy consumption depends on the network goodput, and the network goodput can vary even in the same scenario, we list multiple samples in each scenario with possible network goodput. The testing results of each scenario are listed in Table 5.

The download duration is the duration from the beginning to the ending of the file download, while the total energy includes the energy consumption of the mobile device during the download duration. Pause duration is the total duration when the file download is paused due to power adaptation, and the actual download duration is the download duration deducted by the pause duration. We calculate the network goodput as the downloaded file size divided by the actual download duration. The prediction accuracy is the ratio of the number of correctly predicted samples to the total number of predicted samples. For example, if the predicted SNR is higher than the SNR threshold and the real SNR is lower than the SNR threshold, or if the predicted SNR is lower than the SNR threshold while the real one is higher, it

Table 4: Baseline of energy consumption in different network scenarios.

| Type | SNR mean | SNR standard deviation | Energy(J) | Network Goodput (KB/s) |
|---|---|---|---|---|
| 1 | 14.245 | 5.806 | 92.075 | 54.590 |
| 1 | 11.391 | 5.454 | 118.945 | 36.765 |
| 2 | 19.9 | 3.872 | 52.903 | 158.521 |
| 2 | 19.346 | 6.425 | 43.943 | 178.526 |
| 2 | 15.660 | 5.010 | 45.918 | 165.136 |
| 3 | 24.273 | 4.548 | 38.990 | 225.196 |
| 3 | 26.690 | 4.635 | 55.213 | 147.878 |
| 3 | 26.952 | 3.290 | 63.745 | 120.828 |
| 4 | 22.325 | 9.024 | 62.775 | 60.047 |
| 4 | 21.900 | 8.110 | 45.813 | 176.925 |
| 4 | 24.204 | 5.500 | 194.58 | 18.264 |

Table 5: Experimental results for test cases under different network conditions.

| Type | SNR mean | SNR standard Deviation | Download duration(s) | Pause duration(s) | Network goodput (KB/s) | Energy(J) | Accuracy (%) | Model | Threshold |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 13.450 | 4.625 | 78.751 | 44 | 142.559 | 68.463 | 85.00 | ARIMA | 15 |
| 1 | 13.310 | 6.468 | 101.750 | 37 | 76.511 | 55.200 | 82.76 | ARIMA | 15 |
| 1 | 14.926 | 7.387 | 54.001 | 25 | 170.825 | 59.296 | 77.78 | NFI | 15 |
| 1 | 12.615 | 6.446 | 64.748 | 38 | 185.213 | 60.523 | 87.02 | NFI | 15 |
| 2 | 17.567 | 6.026 | 40.499 | 12 | 173.834 | 49.098 | 78.38 | ARIMA | 15 |
| 2 | 16.514 | 8.862 | 35.249 | 15 | 244.658 | 45.47 | 83.78 | NFI | 15 |
| 2 | 17.283 | 4.662 | 133.999 | 90 | 115.298 | 108.733 | 92.54 | ARIMA | 20 |
| 2 | 18.113 | 5.106 | 81.999 | 53 | 170.836 | 59.590 | 86.25 | NFI | 20 |
| 2 | 18.901 | 4.867 | 60.000 | 31 | 170.800 | 59.718 | 78.43 | MC | 19 |
| 3 | 27.864 | 4.560 | 22.001 | 1 | 235.898 | 39.030 | 90.91 | ARIMA | 20 |
| 3 | 25.519 | 2.513 | 22.000 | 0 | 225.186 | 39.970 | 100 | NFI | 20 |
| 3 | 24.296 | 3.831 | 30.501 | 4 | 190.538 | 46.87 | 74.07 | MC | 19 |
| 4 | 34.206 | 7.014 | 32.251 | 0 | 153.61 | 38.213 | 100 | ARIMA | 15 |
| 4 | 30.842 | 8.772 | 37.999 | 3 | 141.549 | 44.068 | 89.47 | NFI | 15 |
| 4 | 28.926 | 11.198 | 30.251 | 9 | 233.122 | 38.423 | 96.43 | ARIMA | 20 |
| 4 | 26.758 | 6.083 | 30.748 | 3 | 178.538 | 47.648 | 100 | NFI | 20 |

Table 6: Experimental results with low network goodput.

| SNR mean | SNR standard deviation | Download duration(s) | Pause duration(s) | Network goodput(KB/s) | Energy(J) | Accuracy (%) | Model | Threshold |
|---|---|---|---|---|---|---|---|---|
| 28.436 | 5.543 | 276.250 | 1 | 17.998 | 189.645 | 98.91 | ARIMA | 15 |
| 21.089 | 8.971 | 216.509 | 90 | 40.1 | 137.753 | 87.68 | ARIMA | 20 |
| 24.5 | 7.104 | 195 | 50 | 34.165 | 137.617 | 70.97 | MC | 19 |

is considered to be a wrongly predicted sample.

In the network scenarios of Type 1, SNR fluctuates between 0 and 20 with the mean less than 15. When the threshold is set to 15, the file download is paused when the predicted values are lower than 15. According to Table 5, the pause durations take around 36% to 59% of the download duration, whereas the network goodput increase more than 100% compared to the base line. The energy consumption with adaptation is reduced by 40% on average.

For Type 2, when the threshold is set to 20, the pause duration becomes very big. The overhead caused by the adaptation includes the energy overhead of prediction itself and the energy cost during the pause duration. The total overhead is bigger than the energy savings caused by the adaptations. Hence, when the SNR is relatively stable with the standard deviation less than 5 and the mean value bigger than 15, it is not energy-efficient to adopt adaptation with a threshold of 20 even though the prediction accuracy is relatively high. When the threshold is set to 15, the download duration decreases and thus leads to energy savings when compared with the corresponding Type 2 scenario.

For Type 3, the SNR values are high and the standard deviation is very small. The pause duration is very short, and the impact on network goodput is negligible. The energy consumption depends on the network goodput, and the energy overhead is caused by the prediction algorithm, which is around 0.1W. For type 4, when the threshold is set to 15, it leads to energy savings, whereas when the threshold is set to 20 the energy saving is comparatively less.

Due to the workload of the file server and network overload, the network goodput sometimes becomes much smaller under the same SNR conditions. As shown in Table 6, for Type 4, when network goodput is relatively lower, the energy consumption is more than 190J. Compared to the similar case in Table 4, the adaptation can help save energy especially when the threshold is set to 20.

## VI. DISCUSSION

It is well-known that network transmission at a higher data rate results in less energy consumption. Our adaptation aims at adapting the network transmission to network conditions in terms of SNR values by controlling the transfer operations depending on the one-step-ahead SNR prediction. In our adaptation based on the prediction method, if the predicted SNR value is lower than a predefined threshold, transmission will be paused or else continued. Through such adaptation, network goodput could be increased or maintained since the download speed is relatively higher under better network conditions in terms of higher SNR values. In addition, it is possible to reduce some unnecessary retransmission causing high loss rate in a noisy network environment.

The effectiveness of our adaptation depends on the trade-off between the energy overhead caused by the SNR prediction and the energy savings made by increased network goodput. The former one is considered to be stable and independent of network conditions, whereas the latter one varies with network scenarios. To figure out the impact from different network scenarios on the effectiveness of our adaptation, we divide the experimental network scenarios into 4 types based on the mean and standard deviation of SNR values. The type classification is mainly due to the fact that in real time network measurements it is hard to get a stable SNR value.

According to our evaluation results presented in Section 5, when SNR values are generally low as in Type 1, the increase in network goodput is significant, and hence the adaptation is profitable. In the scenarios where the SNR values fluctuate heavily even though the mean of SNR is high, such as Type 4, our adaptation also could save energy to a certain extent. If the network conditions are relatively stable, for example, in Type 2 and 3, when the standard deviation is less than 5, there is not much advantage for the threshold-based adaptation. Energy consumption could not be saved, but is wasted due to the energy overhead caused by network adaptations. In addition to the SNR range, the selection of threshold has an impact on the effectiveness of our adaptation. For example, in the network conditions of Type 2, when the threshold is set to 20, the pause duration is close to 0. However, when the threshold is changed to 15, the adaptation becomes more energy efficient. Hence, in summary, threshold-based adaptation is energy-efficient when network conditions fluctuate in a big range or remain in a relatively bad state. In other words, when SNR values are generally low, such as lower than 15, or when the standard deviation of SNR values is big, network adaptations help save energy up to 40%.

In this paper, we compare the prediction accuracy of different prediction algorithms during offline and online experiments. Our experimental results show that all the three algorithms could attain reasonable high prediction accuracy, and that the energy savings depend more on network conditions than prediction accuracy in our case. In the case that the predicted values are smaller or higher than the real values, it does not always follow that there is an impact on the results of adaptations. For example, when the wrong prediction causes an unnecessary pause of the download, or does not pause the download when the SNR value is very low, it might increase the download duration and waste energy. When both predicted and real values are bigger or smaller than the threshold, it does not change the adaptive operation.

Among the three prediction algorithms, ARIMA has relatively higher prediction accuracy, and NFI performs better than MC. However, ARIMA requires offline learning. Hence in a new network environment which has not been trained, the accuracy might be lower and the energy savings might be reduced too. To choose the prediction algorithm to use for prediction, it would be better to apply ARIMA if the user's moving paths could be predefined. For the application scenarios where the network conditions are totally new to mobile devices, it is wiser to choose NFI which can adjust itself to the new network scenarios quickly.

## VII. CONCLUSION AND FUTURE WORK

To summarize our contribution in this paper, we have explored the possibility of applying network predictions to network-based power adaptation, and implemented three prediction algorithms, ARIMA, NFI, and the MC in adaptive file transfer on mobile devices. We compared the effectiveness of the algorithms while taking the performance of prediction like prediction accuracy and performance overhead into account. The experiments were conducted in a number of settings in public WLANs.

From the lessons learned through prototype level implementation and experimental evaluation, we have figured out the future roadmap for our work that helps to improve the prediction and adaptation techniques used. The temporal difference (TD) method which is often used for reinforcement learning can be a potential topic to be investigated for enhancing our prediction techniques. The advantage of utilizing the TD method is the higher prediction accuracy, and it requires less memory for computation [24]. In addition, TD is applicable for multi-step prediction [25]. As indicated in Section VI the threshold based adaptation could lead to more energy savings, and we will explore the possibility of using dynamic threshold values. The dynamic threshold especially could be helpful in situations where the network behavior fluctuates.

## ACKNOWLEDGMENTS

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology, and Innovation in the Field of ICT). We also extend our thanks to Chengyu Liu for

implementing the prediction algorithms, and Gopalacharyulu, PV for providing valuable suggestions while choosing prediction algorithms.

## REFERENCES

[1] R. Sri Kalyanamaran, Y. Xiao, and A. Ylä-Jääski, "Network Prediction for Adaptive Mobile Applications", UBICOMM'09: In Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, pp. 141 – 146, Malta, October 2009.

[2] Y. Xiao, P. Savolainen, A. Karpanen, M. Siekkinen, A. Ylä-Jääski, "Practical power modeling of data transmission over 802.11g for wireless applications", e-Energy'10: In Proceedings of the 1st International Conference on Energy-efficient Computing and Networking, pp. 75 – 84, Passau, Germany, April, 2010.

[3] X. Long and B. Sikdar, "A Real-time Algorithm for Long Range Signal Strength Prediction in Wireless Networks", WCNC 2008: In proceedings of IEEE Wireless Communications and Networking Conference, pp. 1120 – 1125, 2008.

[4] Y. Vanrompay, P. Rigole, and Y. Berbers, "Predicting network connectivity for context-aware pervasive systems with localized network availability", WoSSIoT'07: In Proceedings of 1st International Workshop on System Support for the Internet of Things, Lisbon, Portugal, March, 2007.

[5] R. Mayrhofer, "An Architecture for Context Prediction", In Advances in Pervasive Computing, Volume 176, Austrian Computer Society (OCG), pp. 65 – 72, 2004.

[6] C.Anagnostopoulos, P. Mpougiouris, and S. Hadjiefthymiades, "Prediction intelligence in context-aware applications", In Proceedings of the 6th international conference on Mobile data management, MDM'05, pp. 137 – 141, 2005.

[7] Y. Wen, R. Wolski, and C. Krintz, "Online Prediction of Battery Lifetime for Embedded and Mobile Devices", In Proceedings of 3rd International Workshop on Power-Aware Computer Systems, pp. 57 – 72, December, 2003.

[8] A. Gupta and P. Mohapatra, "Power Consumption and Conservation in WiFi Based Phones: A Measurement-Based Study", SECON'07: In Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks, pp. 122 – 131, June, 2007.

[9] M . Vukovic, I. Lovrek, and D. Jevtic, "Predicting user movement for advanced location-aware services", SoftCom: In Proceedings of 15th International Conference on Software, Telecommunications, and Computer Networks, pp. 1 – 5, 2007.

[10] D. Narayanan, J. Flinn, and M. Satyanarayanan, "Using History to Improve Mobile Application Adaptation", WMCSA '00: In Proceedings of the Third IEEE Workshop on Mobile Computing Systems and Applications, pp. 31 – 40, 2000.

[11] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer, "Next Location Prediction Within a Smart Office Building", ECHISE'05: In proceedings of 1st International Workshop on Exploiting Context Histories in Smart Environments at the 3rd International Conference on Pervasive Computing, Munich, Germany, May, 2005.

[12] A. S. Weigend, and N. A. GershenField, "Time Series Prediction: Forecasting the Future and Understanding the Past", SFI Studies in the Sciences of Complexity, Proc. Vol XV, Adisson-Wesley, 1993.

[13] G. E. P Box, and G. M. Jenkins, "Time Series Analysis: Forecasting and Control". San Francisco: Holden Day, 1970, 1976.

[14] F. B. Hildebrand, "Introduction to Numerical Analysis" (2nd edition) McGraw-Hill. *ISBN 0-070-28761-9*, Pp 43-59, 1974.

[15] K. C. Barr and K. Asanovic, "Energy-aware Lossless Data Compression", ACM Transactions On Computer Systems. Vol. 24, No. 3, pp. 250 – 291, August, 2006.

[16] Y. Hao, L. Chuang, S. Berton, and L. Bo, and M. Geyong, "Network traffic prediction based on a new time series model", International Journal of Communication Systems, Volume 18, Issue 8, pp. 711–729, October, 2005.

[17] N. Sadek, and A. Khotanzad, "Multi-scale network traffic prediction using k-factor Gegenbauer ARMA and MLP models", AICCSA'05: In Proceedings of the ACS/IEEE 2005 International Conference on Computer Systems and Applications, 2005.

[18] J. Z. S. Tenhuen, and J. Sauvola, "CME: a middleware architecture for network-aware adaptive applications", PIMRC'03: In Proceedings of the 14th International Symposium on Personal, Indoor, and Mobile Radio Communications, , pp. 839 – 849, Beijing, China, Septemer, 2003.

[19] R. H. Shumway, and D.S. Stoffer, "Time Series Analysis and Its Applications: With R Examples", Springer Texts in Statistics, 2nd Edition, pp. 85 – 154, 2006.

[20] D. Katsaros, and Y. Manolopoulos, "Prediction in wireless networks by markov chains", IEEE Journal on Wireless Communications, Volume 16, No. 2, pp. 56 – 63, 2009.

[21] MD. O. Gani, H. Sarwar, and C.M. Rahman, "Prediction of state of wireless network using markov and hidden markov model", Journal of Networks, Volume 4, No. 10, pp. 976 – 984, 2009.

[22] R. R. Sarukkai, "Link prediction and path analysis using markov chains", In Proceedings of the 9th International World Wide Web Conference on Computer Networks, pp. 337 – 386, 2000.

[23] Y. L. Chou, "Statistical Analysis", Holt International, 1975

[24] R. S. Sutton, "Learning to predict by the Methods of Temporal Differences", Journal for Machine Learning, Volume 3, Number 1, pp. 9 – 44, Springer Netherlands, August, 1988.

[25] X. Z. Gao, S.J. Ovaska, and A.V. Vasilakos, "Temporal difference method-based multi-step ahead prediction of long term deep fading in mobile networks", In Computer Communications, Volume 25, Issue 16, pp. 1477 – 1486, October, 2002.

[26] OProfile, System profiler for Linux, http://maemo.org/development/tools/doc/chinook/oprofile/

[27] MYSQL Software Download Site, http://vesta.informatik.rwth-aachen.de/mysql/Downloads/MySQL-5.1

# Highly Accurate Location-Aware Information Delivery With PosPush

Zhao Junhui, Wang Yongcai
*NEC Laboratories, China*
*zhao_junhui@nec.cn*

***Abstract*—This paper proposes PosPush, a highly accurate location-based information delivery system, which utilizes the high-resolution 3D locations obtained from ultrasonic positioning devices to efficiently deliver the location based information to users. This system is designed especially for applications where a 3D space is partitioned into a set of closely neighboring small zones, and as a user moves into one of the zones, the corresponding information will be timely transferred to the user. In order to identify precise zones, PosPush defines a zone model by a set of key location points extracted by a location clustering algorithm, and the zone model is used for online zone identification based on hierarchical searching. In order to determine the appropriate delivery time, an Adaptive Window Change Detection (AWCD) method is proposed to detect the fast change along the location stream. To evaluate the system performance, a MV (Music Video) shelf demo prototype was built in which the information of commodities on the shelf can be delivered based on PosPush. We verified the feasibility and effectiveness of our proposed system from objective and subjective perspectives.**

***Keywords*-Location Based Service, Ultrasonic Positioning, Ubiquitous Computing**

## I. INTRODUCTION

Location data can be contextual indexing information that a system will use to provide services to users, which is called Location Based Services (LBS)[1-6]. Location based Information Delivery System (LIDS) is one typical example of LBS, in which relevant information are predefined for different zones so that when a user enters one of these zones, the related information will be delivered automatically to the user. So far, there are already a lot of research and engineering efforts focusing on LIDS. These conventional LIDS always utilized a very coarse location data of user obtained from either Bluetooth[7,8], Wifi[9-11], RFID[12-14] or GPS[15,16] to provide proximity based information delivery service. Generally, in conventional LIDS, the location zones are relatively large (over tens of square meters). Because of the location coarseness, these zones are roughly defined and have fuzzy boundaries between each other. In addition, the delivery time in conventional LIDS only depends on the instant location of user, i.e., the location based information will be sent to user only if the user is at one of the location zones.

In this paper, we propose PosPush, a highly accurate location-based information delivery system, which utilizes the 3D locations obtained from ultrasonic positioning device [17] to efficiently deliver the location based information to users. In PosPush, a 3D space is partitioned into a set of closely neighboring small zones; a user holds a stick-style ultrasonic tag and moves it into one of zones to get more related information. PosPush can be used in many application scenarios. For example, as shown in Figure 1, in the shopping mall, the commodities on the shelf are placed so closely that the distance between two commodities may be just several tens of centimeters. The vicinity of each commodity forms a small zone. If a customer shows much interests in one specific commodity and wants to know more related information, he/she may approach the commodity and point at it with a stick-style ultrasonic tag, and then the information related to the interested commodity will be pushed to the public display near the commodity shelf. Another example is the exhibition scenario, where many exhibits are placed very closely on the booth table. Visitors can get more information related to one exhibit by moving the ultrasonic stick close to the interested exhibit. It will be a fantastic experience for users because user can get what he/she wants by just pointing a stick towards what he/she is interested in, so we call such a ultrasonic stick as ***Magic Stick***.

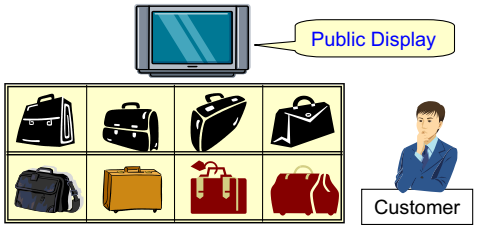



Figure 1. PosPush for Shopping Mall Scenario

PosPush is a novel LIDS with fine-grained location zones and highly accurate location information. Compared with the conventional LIDS, more complex mechanism is highly desirable in PosPush. Exactly, there are two unaddressed problems to the conventional LIDS:

1) Precise zone modeling and identification
   Since the zones in PosPush are small and closely neighboring, precise zone identification is extremely necessary, which poses high requirement to the offline zone modeling and online zone identification.
2) Appropriate delivery time determination
   Considering that there are many neighboring small

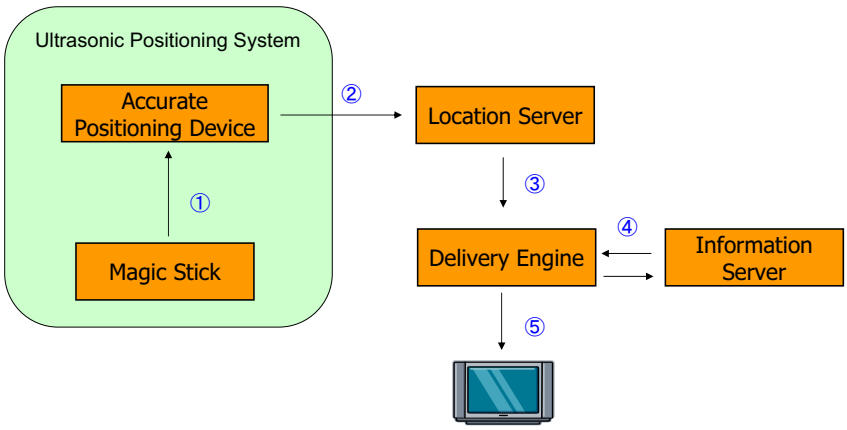



Figure 2. System Overview of PosPush

zones in PosPush, user may unconsciously pass these small zones. If information delivery is triggered whenever a user is detected within a small zone, the user will receive many unwanted information. An appropriate delivery time determination method is highly desired.

To address the above problems, two crucial mechanisms are designed in PosPush. For precise zone modeling and identification, in offline phase, we make use of a location clustering algorithm to precisely calibrate the small zone, which extracts a set of key location points from the randomly collected location trajectory inside a small zone, and in online phase, a hierarchical searching method is used to accurately and quickly determine which small zone the target is located in. To determine the appropriate delivery time, we propose Adaptive Window Change Detection (AWCD) method to tolerate the zone identification error and optimize information delivery time by monitoring the fast change along the location stream. We implement PosPush in a prototypical application for commodity information delivery in shopping mall. Based on this prototype, we carry out a number of experiments to evaluate the performance of PosPush.

The rest of this paper is organized as the follows. we present the system overview in Section 2. Zone calibration and identification algorithm, and delivery time determination algorithm are introduced in Section 3 and 4, respectively. Section 5 presents the prototype implementation and a privacy-preserving approach is also described. The performance evaluation covering not only objective evaluation but also subjective evaluation is in Section 6. Finally, Section 7 concludes the paper.

## II. System Overview

The system architecture of PosPush is illustrated in Figure 2. Such architecture contains several main components:

- ***Ultrasonic Positioning System***

  In PosPush, a ultrasonic positioning system is deployed for locating the mobile targets, which is composed of Accurate Positioning Device and Magic Stick. The positioning device is named as Positioning on One Device (POD) that integrates multiple ultrasonic receivers. POD is mounted, facing downwards, in the room to be covered. Magic Stick is a ultrasonic transmitter carried by user for tracking. It works in active transmission mode and can be located by POD with the accuracy of less than 10 centimeters. For more information about the ultrasonic positioning system, please refer to our prior work [17,18].
- ***Location Server***

  The function of Location Server is to collect the accurate location of Magic Stick. In offline phase, the model for each small zone is calibrated by clustering a sequence of location data samples. In online phase, the real-time location of Magic Stick is used to determine which small zone the Magic Stick is located in by searching the location models.
- ***Delivery Engine***

  Delivery Engine aims to efficiently determine the appropriate time for information delivery. Particularly, Delivery Engine performs an AWCD based method to detect the delivery time by sequentially evaluating the location stream of Magic Stick. At the delivery time, Delivery Engine will send a query for retrieving the location based information from Information Server and then forward the information to the Public Display.
- ***Information Server***

  Information Server contains a location based information database that stores various information associated with the small zones.

Based on the system architecture, the work flow of PosPush is as follows.

① The Magic Stick carried by user emits the ranging signals to POD.

② The accurate location of Magic Stick is calculated by POD and sent to the location server.
③ Location server identifies which small zone the Magic Stick belongs to, and sends the corresponding zone index to Delivery Engine.
④ Delivery Engine determines the appropriate delivery time along the location stream and sends it as a query to Information Server to retrieve the location based information.
⑤ The location based information is delivered to the Public Display for rendering timely.

## III. Zone Calibration and Identification

In PosPush, location server is an important component that performs both offline zone model calibration and online zone identification. The zone defined in PosPush refers to a small 3D space containing a specific object. For example, in shopping mall, a vicinity area around a commodity on the shelf is regarded as a zone. Qualitatively, assume the distance between two neighboring objects is $d$ cm, such a small zone can be described as a cubic space with the edge length of $d$ cm. Figure 3 gives an illustration.

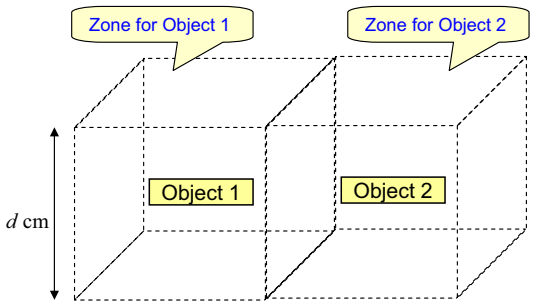



Figure 3. Cubic Space Description for Small Zone

The characteristics of these zones are small and closely neighboring. As far as the cubic space is concerned, there are two basic methods for zone calibration and identification.

1) Center based method
The coordinate of central point is recorded in offline phase, which is taken as the zone model. In online phase, the distance between the real-time location of Magic Stick and the central point of the zone is calculated. The most probable zone can be determined if such a calculated distance is less than $d$ cm.
2) Boundary based method
In offline phase, Magic Stick is moved along the boundary of the cubic space so that the moving trajectory is recorded as the zone model. In online phase, the real-time location is compared with the zone boundary for determining if the tag is located at this zone.

However, the above methods are not so efficient because it is difficult to make a correct judgment when Magic Stick moves between the boundaries of the neighboring zones. The experiment in Section 6 also confirms it. In addition, since the Boundary based method requires user to move Magic Stick strictly along the boundary for calibration, lots of manual efforts will be needed.

To solve the problems of above methods, we propose a new method for precise zone calibration and identification.

### A. Offline Calibration

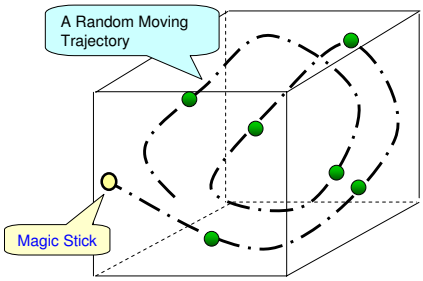



Figure 4. Random Location Collection Inside a Zone

During the offine phase, we perform location clustering method to compute the location model for each small zone. Firstly, we collect a sequence of 3D location data for a small zone by just moving Magic Stick along a random trajectory inside the small zone, which is show in Figure 4.

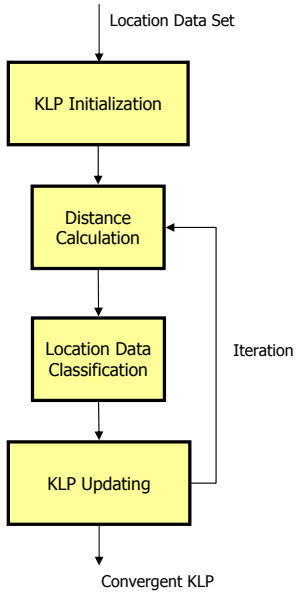



Figure 5. Block Diagram of Zone Calibration

For users, such a random collection approach is flexible and convenient. Afterwards, we apply a clustering algorithm to extract a set of key location points (KLP) form the collected location data to represent the precise model for each zone. There are a fair number of research literatures that

describe various clustering algorithms [19-21]. However, we choose the classical k-means algorithm [22] to find the location model since it is a simple but efficient way to classify the collected location data set into $n$ clusters in which each location data belongs to the cluster with the nearest distance. The whole process is shown in Figure 5. After an iterative process, $n$ KLP are finally refined from the convergent clusters, which are taken as a location model for each small zone. It can be denoted as $P_m = (x_{mj}, y_{mj}, z_{mj}), j = 1, 2, ..., n, m = 1, 2, ..., M$, where $n$ is the number of KLP in one model and $M$ is the total number of the zones. It should be noted that at the case of $n = 1$, i.e., the number of KLP is 1, the proposed clustering based method will be reduced into the above mentioned Center based method.

### B. Online Identification

During the online phase, a hierarchical searching method is used to quickly find the most probable small zone where the Magic Stick is located in. In this method, there are two search stages, which are shown in Figure 6.

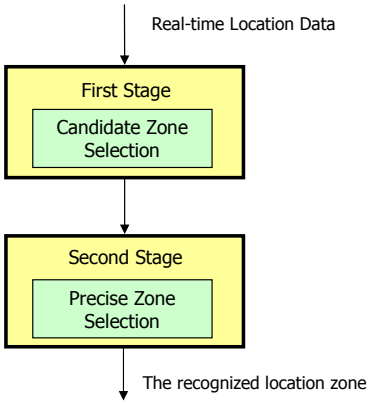



Figure 6. Block Diagram of Zone Identification

The first stage is Candidate Zone Selection, in which for each small zone, only one of KLP is randomly selected for distance calculation to the real-time location data. The distance data are filtered by comparison with a predefined threshold so that only $M_c$ zones from all $M$ zones with minimum distance to the real-time location data are selected as the candidates for next stage processing, especially, $M_c << M$. The second stage is Precise Zone Selection. For each candidate small zone, the average distance between location model of $m^{th}$ zone from $M_c$ candidates and the real-time location data $X_t = (x_t, y_t, z_t)$ is calculated according to the following equation.

$$Dist_m = \sqrt{\frac{\sum_{j=1}^{n} [(x_t - x_{mj})^2 + (y_t - y_{mj})^2 + (z_t - z_{mj})^2]}{n}} \quad (m = 1, ..., M_c) \quad (1)$$

Finally, the index of the most probable zone will be obtained using minimum criteria among the $M_c$ distances, which is depicted as

$$Loc_t = \arg\min_m Dist_m \quad (2)$$

Through the online zone identification to the real-time location data of Magic Stick, the location stream of Magic Stick can be mapped into the matched zone index stream.

## IV. Delivery Time Determination

For purpose of providing friendly user experience, it is important to find the appropriate and reasonable time for information delivery. This is especially true for PosPush where the small zones are close neighboring so that user may unconsciously pass through some of the small zones. If we apply the conventional instant-location based delivery method, many unnecessary information delivery will be triggered, which would be rather annoying to the users. So we define the time for starting and stopping information delivery:

**Information Delivery Time (IDT)**: the moment when the location based information should be delivered to user client as user is close or inside one of location zones.

**Information Stop Time (IST)**: the moment when the location based information delivery should be terminated as user is leaving the location zone.

IDT and IST are detected by sequentially evaluating the newest location data in the location stream to find an adaptive window whose accumulate departing significance is compared against a IDT-threshold and a IST-threshold respectively. The IDT, IST determination problem is illustrated in Figure 7. The upper stream is the location stream and the lower stream is the corresponding zone index stream.

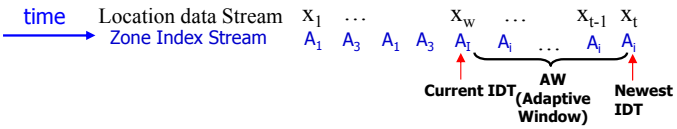



Figure 7. IDT Determination from Streaming Data

The total $M$ small zones can be denoted as $\{A_i, i = 1, 2, ..., I, ..., M\}$. Suppose the time when the location data is $X_w$, which located at zone $A_I$ is the current IDT. At this IDT, $A_I$ related information is delivered to user. Our objective is to find a IST to stop the information delivery to $A_(I)$, as the magic stick leaves $A_I$. After the information delivery stops at IST, we need to find the next IDT to

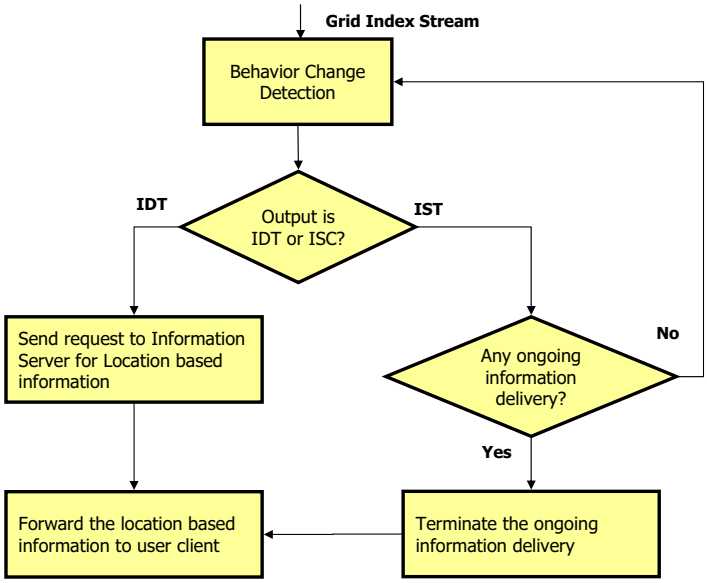



Figure 8. The Block Diagram of IDT and IST Detection

start new information delivery process as the magic stick enters another zone. Particularly, we focus on detecting the moving trend within a time window that is started at $X_w$ and ended at the newly-received $X_t$. Since the window size is always adaptive to newly-received location data, we call the proposed ISC and IDT determination method Adaptive Window Change Detection (AWCD).

The key idea of AWCD is that we calculate Accumulate Departing Significance (ADS) in the adaptive window, which is used as a measurement to online check the trend of Magic Stick departing from the current IDT zone $A_I$. Since we have known the last location at IDT, which is $A_I$, AWCD algorithm tolerates noises by following schemes:

1) if the following location data in the stream departs from $A_I$ only a little, we predict that the location of Magic Stick has not changed.
2) If the trend that the data in adaptive window departs from $A_I$ is larger than a user defined IST-threshold, an IST event is detected and the location based information delivery should be stopped.
3) If the trend that the data in adaptive window departs from $A_I$ is larger than a user defined IDT-threshold, A new IDT is detected to inform the new information delivery.

In details, the AWCD algorithm includes the following steps.

**1) Reset Step**

At the time of the current IDT (that is, $X_w$ located at $A_I$ as shown in Figure 7), the ADS to all of zones except $A_I$ are reset to zero.

$$ADS(A_i, X_w) = 0, \forall A_i, (i = 1, ..., M, i \neq I) \quad (3)$$

**2) Update Step**

Each time the new location data $X_t$ is received, the ADS to all of zones except $A_I$ are calculated in the adaptive window from $X_w$ to $X_t$ by a recursive approach.

$$ADS(A_i, X_t) = ADS(A_i, X_{t-1}) + DS(A_i|A_I, X_t) \quad \forall A_i, (i = 1, ..., M, i \neq I) \quad (4)$$

where $DS(A_i|A_I, X_t)$ is Departing Significance that is defined as

$$DS(A_i|A_I, X_t) = \begin{cases} -\alpha, & \text{if } X_t \in A_I \quad (a) \\ \beta, & \text{if } X_t \in A_i \quad (b) \\ 0, & X_t \notin A_i \text{ and } X_t \notin A_I \quad (c) \end{cases} \quad (5)$$

where $\alpha$ and $\beta$ are positive values. From the above equations, we can find that if $X_t$ is located at $A_I$, all ADS will be decreased by $\alpha$, but if $X_t$ is located at one zone $A_i$, the ADS of $A_i$ will be increase by $\beta$ and all other ADS will keep unchanged.

**3) Maximum Step**

As the real-time location data is received continuously, we can obtain the spatial-temporal distribution of ADS, which is shown in the following equation.

$$\begin{matrix} ADS(A_1, X_w) & ADS(A_1, X_{w+1}) & & ADS(A_1, X_t) \\ ADS(A_2, X_w) & ADS(A_2, X_{w+1}) & & ADS(A_2, X_t) \\ . & . & ..... & . \\ . & . & & . \\ ADS(A_M, X_w) & ADS(A_M, X_{w+1}) & & ADS(A_M, X_t) \end{matrix} \quad (6)$$

Each line represents a temporal trajectory of the ADS of one zone; and each column represents the ADS distribution for all zones at one time. According to the update step, the ADS to $A_i$ will be monotonically increasing if Magic Stick has an apparent trend of leaving $A_I$ and entering $A_i$; On the other hand, all ADS will be relatively small if the location steam of Magic Stick concentrate on the vicinity of $A_I$. So, for each time, the maximum value of ADS should be picked out.

$$M(X_t) = \max_z(ADS(A_z, X_t)) \quad (7)$$

**4) Judgment Step**

Finally, the maximum value of ADS is checked against the predefined thresholds for IST and IDT. The threshold of IST is predefined as $T_{IST}$. It means that when the accumulate departure trend is larger than $T_{IST}$, the current information delivery should be stopped. The threshold of IDT is predefined as $T_{IDT}$, which means that when the accumulate departure trend to another grid is large enough, a new information delivery process should begin. Generally $T_{IST}$ is smaller than $T_{IDT}$.

Figure 8 shows the block diagram of the IDT&IST detection. If $M(X_t)$ is larger than the IST threshold (i.e., $T_{IST}$), an IST is detected, the current information delivery stops.

If $M(X_t)$ is larger than the IDC threshold (i.e., $T_{IDT}$), an IDT is detected, and

the new IDT zone will be found accordingly.

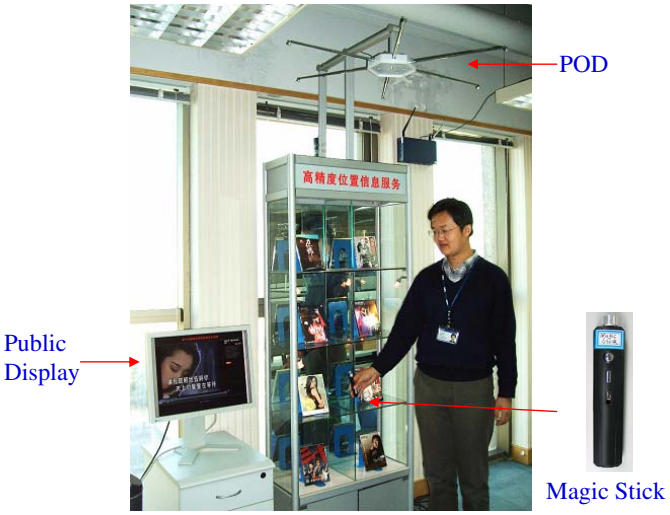
$$A_z = \arg\max_z(ADS(A_z, X_t)) \quad (8)$$

## V. Prototype Implementation - Information Delivery of Music Video Disc

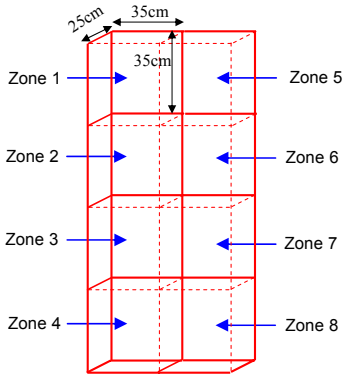
Based on PosPush, we implemented a prototypical application of commodity information delivery in shopping mall scenario. Particularly, considering that Music Video (MV) is popular to most people, we choose MV disc as the commodity in our prototype, in which if customer is interested in a MV disc, he/she will move the Magic Stick close to it and soon a video clip from this MV disc will be rendered on the Public Display. Figure 9(a) illustrates our prototype.

Basically, the prototype mainly includes:

- **Commodity Shelf and MV Disc:** In the prototype, a commodity shelf with eight grids is used. Each grid contains one MV disc so totally eight discs are placed on the shelf. It should be noted that the MV discs we used are from the popular singers nowadays for attracting the attentions of the visitors. The grids on the shelf are closely neighboring and each grid has a size of 35cm height and 35cm width. Since ultrasonic signal may be obstructed by the grid board, we define a vicinity area outside the shelf but close to a MV disc as the small zone for information delivery. Exactly, the zone size is $35cm \times 35cm \times 25cm$, as shown in Figure 9(b). The zone index is also given.
- **POD and Magic Stick:** A POD is fixed on the top of the commodity shelf to cover all zones of eight MV discs. It has a hexagon shape containing six surrounding ultrasonic receivers. A stick-style ultrasonic tag is used as Magic Stick to be held by user. As the user moves the Magic Stick to the interested MV disc, POD can calculate the accurate 3D location of Magic Stick and sent the location data to a server computer. It should be pointed out here is that the updating rate of Magic




(a) Demonstration




(b) Zone Definition

Figure 9. PosPush Prototype

Stick's location is 5Hz, i.e., the transmission interval of Magic Stick is 200ms.

- **Server Computer:** On server computer side, the video clips of all MV discs are stored at an information database, and a Java-based PosPush software is developed to perform both precise zone calibration&identification and IDT determination. PosPush software also provides a user-friendly GUI for ease of use. In offline phase, PosPush aggregates a sequence of location data in the vicinity area of a MV disc and extracts a set of KLP as the zone model. Then, the location model is associated with the corresponding video clip of the MV disc. In online phase, PosPush obtains the appropriate IDT from the location stream of Magic Stick. At the IDT, the video clip is retrieved from information database and is played on the Public Display.

### A. User Privacy

A potential problem to Magic Stick mode is user privacy issue. Considering that public display is used to render the video clips and there are always many people in shopping mall, it is unavoidable that the video clip of a user interest will be seen by others, which leads an uncomfortable experience. Another problem is that only one public display is placed in a shelf so that more than two users can not view the interested information simultaneously. To solve this problem, we further designed a small USB-style ultrasonic transmitter, which can be easily plugged into user's personal mobile device, such as PDA or mobile phone. As user want to know more information to the interested commodity, he/she just move the personal mobile device close to such a commodity and soon, more information will be delivered to user's personal mobile device via wireless connection, e.g., Wifi, Bluetooth, etc. Figure 10 gives an illustration to this scenario.

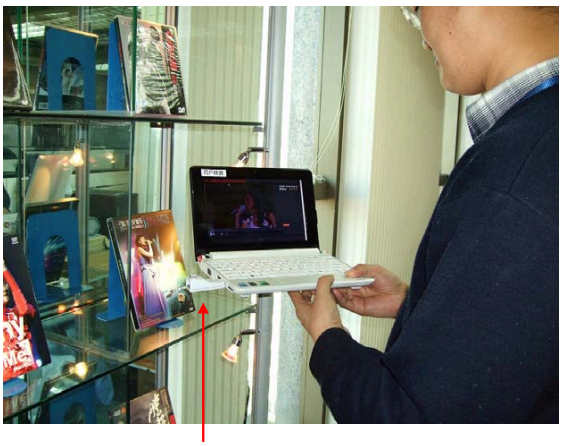



Figure 10. Prototype for User Privacy Preservation

## VI. Performance Evaluation

Based on the prototype, a number of experiments are carried out to evaluate the PosPush performance. In addition, we showed the MV demo in a formal exhibition to visitors for subjective evaluation.

### A. Evaluation of Zone Calibration&Identification

In PosPush, zone models are offline calibrated by applying k-means clustering to a randomly collected location sequence with length $L$, so that $n$ KLP can be extracted as the zone model. The selection of parameters $L$ and $n$ will affect the complexity of zone calibration and the accuracy of the zone identification. On one hand, larger value of $L$ and $n$ can create a more precise zone model. On the other hand, larger $L$ means more time consumed for location collection processing. Thus we expect to find an optimal combination of parameter $L$ and $n$ so as to obtain a precise location model while minimizing the calibration efforts as much as possible.

In our experiment, the length $L$ of collected location sequence during offline calibration phase varies from 30, to 60 and 90. Considering that the update frequency of Magic Stick is 5Hz, the corresponding collection time is 6s, 12s and 18s respectively. The number of KLP ($n$) ranges from 1 to 20. For each pair of parameter $L$ and $n$, zone models are built for all the eight zones of the prototype. In online phase, we collect a set of real-time location as the test data. Particularly, most of the test data are collected near the boundary of zones. The ground truth of zone index stream is manually labeled. We use the error rate of zone identification for performance evaluation. The results are shown in Figure 11. From this figure, we can see that, firstly, the length of collected location sequence has little effect on the zone identification accuracy. Especially, when $n > 10$, although the length of the location sequence is different, the identification accuracy are almost identical. Secondly, the identification accuracy is improved with the increase of the number of KLP, but the accuracy will increase very slightly when $n > 10$.

Therefore, according to the experiment results, we select $L = 30$ and $n = 11$ as the optimal parameters. With this parameter setting, a precise zone model can be built for online identification through moving Magic Stick in a specific zone for just about 6s.

Next, we perform experiment to compare the error rate between our proposed clustering based method and the two baseline methods mentioned in Section 3. In online identification phase, we use the same test data with the above experiment to evaluate the performance of the three methods. Figure 12 shows the comparison result. From this figure, it is obvious that our proposed method can achieve the best accuracy compared to the other two methods. We also find that the error rate of Center based method is very close to

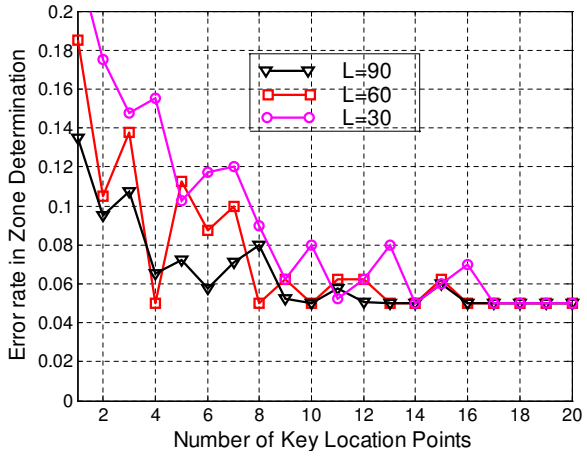



Figure 11. Effect of $L$ and $n$ to the Zone Determination Accuracy

the case of $n = 1$ in the clustering based method as shown in Figure 11.

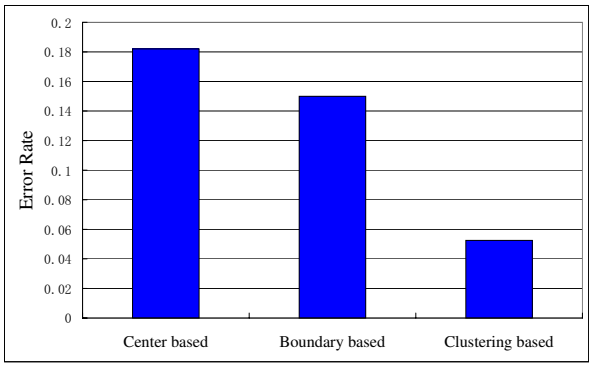



Figure 12. Comparison between Different Methods

### B. Evaluation of AWCD

To evaluate the performance of the proposed AWCD method, we need a metric to check if there is a good match between the IDT determined by AWCD and the user's true interest. We propose Matching Rate between the user's true interest and the actually delivered information as the evaluation metric. At each time instance, user's true interest is manually labeled. If the delivered information is the same with the user's true interest, a "match" is counted; otherwise, a "mismatch" is counted. After the user has used the PosPush system for a period, a sequence of "match" and "mismatch" will be recorded. Matching Rate is defined as the percentage of the whole sequence occupied by the "match" number. For convenience of evaluation, we index all zones. The four zones on the left column are indexed as Zone 1 to Zone 4 from top to bottom, and the four zones on the right column are indexed as Zone 5 to Zone 8 from top to bottom.

The experiment result of one test is shown in Figure 13. The Y-Axis is the zone index, and the X-Axis is time. A user moves the Magic Stick in the shelf area. His true interest is the commodities placed in the Zones 1, Zone 6, Zone 3 and Zone 8. The green dotted line shows the zone determination results. We can see the trace starts from Zone 1 for a while, then passes Zone 5 and stays in Zone 6 for a while; next fast passes Zone 7 and stays in Zone 3 for a while; finally fast passes Zone 4 and ends in Zone 8. The true interest is labeled by the user himself and is shown in the dotted black curve. According to the zone determination result, the Magic Stick traveled all of zones considering that there are fast-pass zone result and error determination result. If the delivery time is just dependent on the instant location like the conventional LIDS, the information related to all zones will be pushed to user. It will bring uncomfortable experience to users. For comparison, the delivered result generated by our proposed AWCD is plotted in the solid purple curve. It can be found that the information delivered by AWCD well matches the user's true interest and AWCD can tolerant the zone identification error and fast-pass zone result. Exactly, the Matching Rate of AWCD based method is as high as 96.5%.

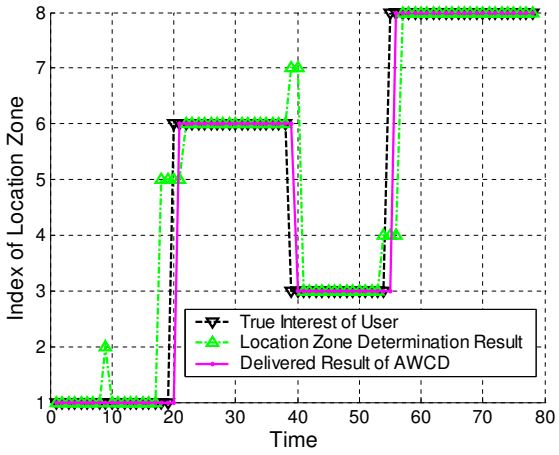



Figure 13. Experiment of AWCD based Delivery

We carry out 50 times of above test to compare the Matching Rate between AWCD and the conventional instant-location based method. The average Matching Rate are summarized in Table I. We can see AWCD-based method can improve Matching Rate over 10% compared with the conventional method.

Table I
MATCHING RATE COMPARISON

| | **Instant-Location based Method** | **AWCD based Method** |
|---|---|---|
| Matching Rate | 86.1% | 96.7% |

### C. Subjective Evaluation

To better understand user's subjective affection to PosPush, we showed our MV shelf prototype at NEC Solution Fair 2008 (as shown in Figure 14).

During this exhibition, we presented the MV information delivery demo to many visitors who had never seen PosPush before. It is worth noting that almost all of the visitors were from information technology domain, which means they may have a stronger willing to experience new applications. Firstly, we introduced the visitors to PosPush and then explained MV shelf demo to them. Such a live demonstration attracted the attention of visitors immediately. Most of the visitors took the Magic Stick or PDA in their hands and experienced PosPush system by themselves. After the visitors had experienced the functionalities of PosPush, they gave us their feedback. One hand, we got some praising words, such as

1) *"It is really the first time for me to see such a technology."*
2) *"It is an interesting experience for me."*
3) *"It is definitely high accurate locating system."*

On the other hand, we also got some advice, such as

1) *"If I use Barcode reader to scan the goods ID and send it to server via wireless, I can also get the more information of my interested goods."*
2) *"PosPush may just be applied for MV shop, but it is more common in shopping store that the commodities (such as shampoo, cookie) are placed very close, so it is difficult to realize PosPush in such real scenario. "*

## VII. Conclusion

In this paper, we designed, implemented and evaluated PosPush - a highly accurate location based information delivery system, which utilizes the 3D locations obtained from ultrasonic positioning device to efficiently deliver the location based information to users. To meet the requirement of practical application, we proposed two key mechanisms. 1) Precise zone modeling and identification 2) AWCD based delivery time determination. Based on PosPush, we implemented a prototypical application of commodity information delivery in shopping mall scenario. It verified the feasibility and effectiveness of our proposed system. In the future, there are several directions to pursue in order to further improve PosPush. Firstly, we investigated the PosPush system into the zone with the size of tens of centimeters in this paper. But it is common in current supermarket that the commodities (such as cookie, shampoo etc.) are placed one by one so that the zone has a smaller size of less than $10cm$. So we plan to apply PosPush system to such smaller zones for performance evaluation. Secondly, the current PosPush is based on the ultrasonic positioning system. We will investigate the use of other 3D positioning system [23-25] for PosPush in near future.

## VIII. Acknowledgement

The authors would like to thank Wang Yanzhe for his effort and assistance in the testing experiments and in helpful discussions regarding system development. The authors would also like to thank Toshikazu Fukushima and Zhang Hongming for their valuable support.

## References

[1] E. Beinat, *Location-Based Services: Market and Business Drivers*, GeoInformatics, vol. April, pp. 6-9, 2001.

[2] ABI Research, *Location Based Services*, Allied Business Intelligence, 2004.

[3] R. Bharat and M. Louis, *Evolution of Mobile Location-based Services*, Journal of Communications of the ACM, Vol. 46, pp. 61-65, 2003.

[4] JOS, R., et al. *An Open Architecture for Developing Mobile Location-Based Applications over the Internet* , 6th IEEE Symposium on Computers and Communications, July, pp.3-5, 2001.

[5] KAGAL, L. et al. *A highly adaptable infrastructure for service discovery and management in ubiquitous computing*, Technical Report TR CS-01-06, Department of Computer Science and Electrical Engineering, University of Maryland: Baltimore, United States, 2001.

[6] Barnes, S. J. *Known By the Network: The Emergence of Location-Based Mobile Commerce*. Advances in Mobile Commerce Technologies, 171-89, 2003

[7] Anastasi, G., Bandelloni, et al. *Experimenting an Indoor Bluetooth-Based Positioning Service*. ICDCSW' 03: 23rd International Conference on Distributed Computing Systems Workshops 2003, 480-83.

[8] L. Aalto, N. Gõthlin, J. Korhonen and T. Ojala T, *Bluetooth and WAP Push based Location-aware Mobile Advertising System*, Second International Conference on Mobile Systems, Applications and Services, pp. 49-59, Boston, MA, 2004.

[9] S. Yamada, Y. Watanabe, H. Kitagawa and T. Amagasa, *Location-Based Information Delivery Using Stream Processing Engine*, International Conference on Mobile Data Management (MDM06), pp. 57-61, 2006.

[10] Anthony Lamarca, Yatin Chawathe, et al.,*Place Lab: Device Positioning Using Radio Beacons in the Wild*, In Proceedings of the Third International Conference on Pervasive Computing (May 2005)

[11] P. Bahl and V. Padmanabhan. *RADAR: An In-Building RF-based User Location and Tracking System*. In Proc. IEEE INFOCOM (Tel-Aviv, Israel, Mar. 2000).

[12] N. Michiko, O. Naoya, T, Ryuji, *Just-in-Time Information Delivery System for Passenger Assistance*, Journal of Q Rep RTRI, Vol.47; No.4; pp.187-191, 2006.

[13] J. Hightower, C. Vakili, G. Borriello, and R. Want. *Design and Calibration of the SpotON Ad-Hoc Location Sensing System*. August 2001.


Figure 14. User Experience in Exhibition

[14] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. *LANDMARC: Indoor Location Sensing Using Active RFID*. In Proceedings of IEEE PerCom 2003, Dallas, TX, USA, March 2003.

[15] N. Marmasse and C. Schmandt, *Location-Aware Information Delivery with ComMotion*, International Symposium on Handheld and Ubiquitous Computing , pp. 361-370, London, UK, 2000.

[16] A. Hinze and A. Voisard. *Location- and Time-based Information Delivery in Tourism*, International Symposium on Advances in Spatial and Temporal Databases (SSTD 2003), pp. 489-507, July 2003.

[17] J. Zhao and Y. Wang, *Autonomous Ultrasonic Indoor Tracking System*, IEEE International Symposium on Parallel and Distributed Processing with Applications, pp.532-539, Sydney, Austrialia, 2008.

[18] Y. Wang and J. Zhao and T. Fukushima, *LOCK: A Highly Accurate, Easy-to-Use Location-based Access Control System*, 4th International Symposium on Location and Context Awareness (LOCA'09), Tokyo, Japan, 2009.

[19] J. Hartigan, *Clustering Algorithms*. Wiley.

[20] MacKay and David, *Chapter 20. An Example Inference Task: Clustering*, Information Theory, Inference and Learning Algorithms, Cambridge University Press. pp. 284-292, 2003.

[21] Luger, George; Stubblefield, William . *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (5th ed.). The Benjamin/Cummings Publishing Company, Inc. ISBN 0-8053-4780-1.(2004)

[22] J. Hartigan and M. Wong, *Algorithm AS 136: A K-Means Clustering Algorithm* Applied Statistics Vol.28, No.1, pp. 100-108, 1979.

[23] R. FONTANA, *Recent System Applications of Short-pulse Ultra-wideband (UWB) Technology*.IEEE Trans on Microwave Theory and Techniques, 52(9), pp. 2087-2104, 2004.

[24] W. Chung and D. Ha, *An accurate ultra wideband (UWB) ranging for precision asset location*, in Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies (UWBST'03), pp. 389-393, Reston, Va, USA, November 2003.

[25] R. Fleming, C. Kushner, G. Roberts, and U. *Nandiwada, Rapid acquisition for ultra-wideband localizers*, in Proceedings of IEEE Conference on Ultra Wideband Systems and Technologies (UWBST'02), pp. 245-249, Baltimore, Md, USA, May 2002.

# Micro-Mobility Solution Based on Intra-domain multicast and Congestion Avoiding for Two-Nodes Mobile IP Network

Yacine Benallouche and Dominique Barth
PRiSM-UMR 8144
Université de Versailles Saint-Quentin
45, Avenue des Etats-Unis
78035 Versailles Cedex, France
Email: {bey, dominique.barth}@prism.uvsq.fr

***Abstract*—In this paper, we deal with micro-mobility in TWINBOARD network, which is a two nodes mobile network architecture based on an all IP infrastructure. The two nodes are the Base Station (BS) and the Access Gateway (AG). To manage micro-mobility, we propose a new approach providing efficient and smooth handover, while being able to coexist and inter-operate with existing technologies. Specifically, we propose an intra-domain multicast-based handover approach combined with an Alert mechanism. Alert approach is a distributed mechanism that provides routers with information regarding the congestion state of other routers without any modifications on existing routing protocol. Our solution achieves an efficient intra-domain handover and avoids flooding in the network. The simulations used to evaluate our scheme and compare it to other multicast scheme - DVMRP (Distance Vector Multicast Routing Protocol) show that our solution presents a good performance and outperforms DVMRP scheme. Our main contribution consists on an efficient new approach to manage IP micro-mobility using intra-domain multicast with alert mechanism.**

***Keywords* - Micro-mobility, intra-domain handover, multicast algorithm, alert algorithm**.

## I. INTRODUCTION

Current mobile networks are composed of several network elements interconnected by specific network infrastructures, leading to important development, deployment and maintenance costs. For example, in the data-path of 3G networks defined by 3GPP there are at least four types of interconnected nodes: node B, RNC, SGSN and GGSN. The architecture under definition at the WiMAX Forum presents a step forward towards simplification in defining an IP based infrastructure connecting 3 nodes [23]: Base Station (BS), Access Gateway (AG), and Anchor Point/Home Agent (HA).

Considering the topics and the objectives of the Next Generation Mobile Networks (NGMN) initiative launched by major European and North American mobile network operators [20], the European CELTIC project TWINBOARD [1] investigates the performances and the cost of a two nodes IP architecture for a mobile network (see Figure 1). In this project, we consider a mobile network composed of BSs connected to one or some enhanced AGs through a dedicated network that we call **Access Aggregation Network (A2N)**. The major objective of the TWINBOARD project is to propose a novel **A2N architecture** considering a specific features of IP networks especially in terms of load distribution, reliability, and flexibility. Here all mobility related functions and associated features are ensured by BS-AG tandem.

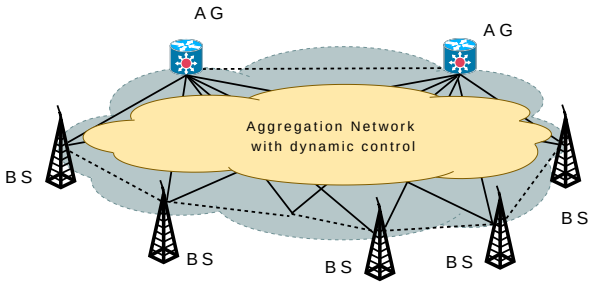



Fig. 1. TWINBOARD network Architecture

The target architecture defined by TWINBOARD recommendations is an optimized Packet Switched (PS) network architecture, which will provide a smooth migration of existing 2G and 3G networks towards an IP network with improved cost competitiveness and broadband performance. The A2N network is an IP based network and loosely meshed with tree-like traffic

[1]TWINBOARD European CELTIC PROJECT, see http://www.celtic-initiative.org/Projects/TWINBOARD/default.asp.

pattern -mostly GW from/to BS- that is changing due to mobility. Due to these peculiarities, load sharing and resilience mechanisms known from the Internet are expected to yield suboptimal results.

**Related works:** Several studies on micro-mobility [7], [16] show that Mobile IP (MIP) [11], the proposed standard, has several drawbacks from its network overhead and its end-to-end delays due to the triangle routing problem. Many micro-mobility approaches attempt to improve MIP [13], [14] in current IP mobile networks. However, such approaches suffer from complexity and handover performance [6]. To the best of our knowledge, the proposed utilization of multicast combined with alert message diffusion in Two-Nodes Mobile IP network has never been studied to manage intra-domain handover.

The rest of the paper is organized as follows. In Section II, we give an overview of multicast protocols. In Section III, we describe our proposed algorithms and we prove the NP-completeness of the related problem. Section IV presents our TWINBOARD simulator and its environment (used topologies, multicast group and traffic model). Section V gives evaluation and simulation results. In Section VI, we present conclusions and outline perspective future works.

In our case, we propose two distributed algorithms mainly implemented on GW and BS nodes to guaranty QoS dealing with the real time constraints of mobility. The paradox we deal with is to guarantee moving mobile connections by insuring enough flexible resource use on an IP routing without considering too expensive mechanisms on each router of the aggregation network. Considering these attempts, the solution we propose focuses in particular on optimization of the route tables and on traffic load balancing techniques between BS and GW, without using resource allocation mechanisms in the IP interconnection network.

The architecture we propose is based on two collaborative algorithmic mechanisms. Firstly, to avoid congestion in the network and to insure flexibility in the use of the bandwidth of the network, we adapt the routing alert algorithm proposed in [19] for inter-domain network by introducing a hierarchy concept in the IP network based on the particularity of the considered traffic (from BS to GW and from GW to BS). Secondly, we propose a distributed process to control multicast functionalities in each IP router to obtain a constraint delay multicast tree compatible with the embedded IP routing. Here multicast is used to anticipate handover when it is considered as probable. Then traffic to the target mobile user is multicasted to the BS to, which it is connected and to the geographical neighboring BS. The aim of the proposed process using only IP basic mechanisms, is to limit congestion overhead due to the multicast.

These algorithmic solution takes benefits of forwarding and routing of datagrams and presents several natural advantages:

- Cheap installation and exploitation on adapted and inexpensive existing IP infrastructure.
- Good load distribution (Alert mechanism).
- Ability to coexist and to inter-operate with existing technologies.

## II. Multicast Overview

We categorize algorithms for the multicast tree construction in two categories [17]:

1) Source-Based Algorithms (SBA).
2) Core-Based Algorithms (CBA).

In SBA algorithm the tree's root is the source node and the leaves are the multicast group's components. SBA is currently used as the tree construction algorithm for Distance Vector Multicast Routing Protocol (DVMRP) [18], Protocol Independent Multicast Dense Mode (PIM-DM) [3], and Multicast Open Shortest Path First (MOSPF) [10].

The CBA or the core-based algorithm selects a core node as a multicast tree's root. Afterwards, a tree rooted at the core node is constructed to reach all the multicast group's members. In this case, the core node is different from the source and it is very important to select the best one as much as possible. Therefore the source send messages to the core node, which distribute those messages to the destinations. Among the protocols that use the CBA we can cite Protocol Independent Multicast Sparse Mode (PIM-SM) [4] and the Core-Based Tree (CBT) protocol [2]. The core-based algorithms are highly suitable for sparse groups and for large networks. Indeed they provide excellent bandwidth conservation for receivers. With the multicast technology, multimedia applications, such as videoconferences, require an efficient management of the QoS. An essential factor of these real-time strategy is to optimize the DVBMT problem [12].

The multicast delay variation is the difference of the maximum end-to-end delay and the minimum end-to-end delay among the paths from the source node to all the destination nodes. Minimizing this parameter

allows all the destination nodes to receive the same data simultaneously as much as possible. One issue to the DVBMT problem is to minimize multicast delay variation under multicast end-to-end delay constraint. In [12], authors propose a heuristic solution called Delay Variation Multicast Algorithm (DVMA), where they construct at first the tree by considering only the end-to-end delay constraints. Afterwards, they enhance the tree by considering the the multicast delay variation constraint. Nevertheless, DVMA presents a high time complexity, which is does not fit in modern applications.

Another heuristic solution with lower time complexity than DVMA is called Delay and Delay Variation Constraint Algorithm (DDVCA). DDVCA is based on the Core-Based Tree (CBT), where the core node is selected as the node with minimum delay variation with all other multicast group's nodes. However, the DDVCA exhibits high network charge around the core node. Indeed, all the multicast packets transit through the core node, this last one resends these packets to the leaves.

Our multicast algorithm overcomes these limitations and it is used in wireless networks to manage the handovers by constructing an optimized tree from one from an Access Gateway (AG) to some Base Stations (BS) on, which mobility can be predicted. Unlike DDVCA where the tree construction is based only on one core node, our distributed solution extends this construction on several core nodes. This allows us to minimize the bandwidth consumption as we spread the charge on different core nodes.

## III. MULTICAST&ALERT-BASED MICRO-MOBILITY

In this section we present our solution to manage micro-mobility with congestion avoiding. We propose two algorithms: Multicast algorithm and alert algorithm.

### A. Problem Context

Handover performance and router congestion are a significant factors in evaluating performance of IP mobile network. With the Internet growth it becomes crucial to design efficient, scalable and robust handover protocols. We propose a new architecture for providing efficient and smooth handover with congestion avoiding. Our approach consists of two distributed algorithms:

- Multicast Algorithm used to construct an optimized multicast tree from an Access Gateway (AG) to some Base Stations (BS) on, which mobility can be predicted [1].
- Alert Algorithm used to avoid IP router congestion [19].

Note that the multicast will occur only within the period during, which the BS communicates the candidates cells to the moving Mobile Node. Once this information is acquired by the Mobile Node and one cell is selected, then the unicast routing runs again and the multicast is interrupted. The handover trigger is based on a simple power signal comparison between the current BS and the candidate one. In our case, Handover Trigger event are invoked when the received signal level in the current cell becomes lower than the pre-defined thresholds. A *handover_trigger* notification message is sent by the current cell to the AG in order to launch the multicast to the candidate cells.

We consider a single domain as shown in Figure 2. The Border Router (BR) connects the network to the internet and one Access Router (AR) serves a number of BSs. When a mobile node moves from one BS to another without changing its AR, we talk about an intra-AR handover case that is not considered in this paper because it is specific to AR implementation.
Each BS on, which mobility can be predicted is assigned a multicast address and it sends a notification to its attached $AG$ . We do not focus on mobility prediction, so we assume that we know the Base Stations set on, which mobility can be predicted [8], [15]. Considering this set to be the multicast group $M_i$, it appears suitable to match them to the local group given by the neighboring set $NS_i$ of each Base Station $BS_i$ in the planar graph $PG = (V_p, E_p)$, where $V_p$ is the set of Base Stations and $E_p$ is a set of directed links. Therefore, for a given $BS_i$ a multicast group $M_i \subset V_p$ is constructed by $m$ Base Stations belonging the set $NS_i$ and distributed geographically on the TWINBOARD access network.

Actually, the multicast is triggered at the AG level considered as the intelligent entity responsible of processing the handovers and other functions in TWINBOARD network. In the simulated handovers scenario (Figure 2), one mobile node attached to the Base Station $BS_i$ is in communication with another mobile node attached to the Base Station $BS_j$ that may roam outside of its serving cell. Before handover trigerring, unicast packets are sent by $BS_i$ to its attached AG along the IP routers network by using an unicast routing protocol combined with the Alert mechanism. Once the handover is detected, and according to the address of $BS_j$ obtained from the unicast packet, the AG selects the multicast group members $m_j$ from the $NS_j$ set. Thus, it constructs the multicast packets to be forwarded to each multicast group member using the proposed Multicast Algorithm combined with the Alert Algorithm.

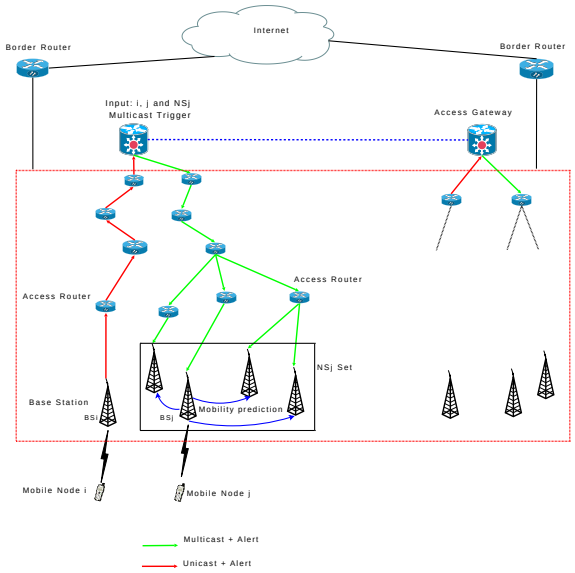



Fig. 2. TWINBOARD Architectural view

### B. Multicast Algorithm

When a mobile user moves from one point of access to another within a domain, a handover event takes place between the two points of access. Handover involves to redirect the incoming traffic flow to the new access point. In proactive handover the link between the mobile user and the new access point is established prior to its disconnection with the old access point. Hence a smooth handover, i.e., handover with low packet loss, can take place by exploiting the fact that the new access router is known a priori and that multicasting allows proactive path setup to the new access router. The packets are multicasted to the mobile nodes within the domain where handover can be predicted. Mobility prediction need not necessarily be a part of the micro-mobility algorithm as it can be better achieved with additional information from lower layers.

**Problem definition:**
We define the related multicast problem from a graph theory point of view (see Figure 3 for an example). Then, we show that it is NP-complete and not approximable problem. We consider a symmetric digraph $G = (V, E)$ within a coherent routing function $R$, a vertex $s \in V$ (called transmitter) and a vertex subset $D$ (called destination set).

*Definition 1: We define a* **Broadcast scheme** *from $s$ to $D$ as a subtree of $G$ rooted in $s$, with depth equal to $\max_{v \in D} |R(s, v)|$, with leaves in set $D$ and where the edge set can be decomposed in a set of paths such that*:

1) *The initial extremity of each path is $s$ or an intermediate vertex with outgoing degree $> 1$ in the subtree.*
2) *The final extremity of each path is a leaf.*
3) *Each path $< i, f >$ corresponds to the route $R(i, f)$.*

The number of edges in the subtree defines the *size* of the broadcast scheme.

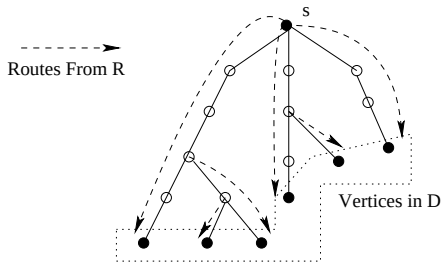



Fig. 3. An example of multicast scheme

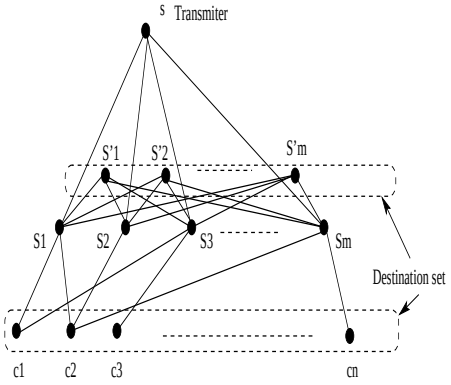



Fig. 4. Polynomial reduction

The problem of finding a multicast tree minimizing the number of edges under an IP routing constraint can be defined as follows:

**Problem:** Destination_Mobility (DeMo)
**Given:** A symmetric digraph $G = (V, E)$, a coherent routing $R$, a transmitter $s \in V$ , a destination set $D$ and an integer $k$.
**Question:** Does there exist a Broadcast scheme with size at most $k$ ?

**Complexity:**
*Theorem 1:* For some $c > 0$, Problem **DeMo** is NP-complete and not approximable within $c \log(n)$,where $n$ is the number of vertices of the graph.
To prove this theorem, we propose a polynomial reduction from the **SET-COVER** problem.

*Proof:* Problem DeMo is clearly in NP. Indeed, it is easy to check in polynomial time for a given structured trees if one of them is (or not) a broadcast scheme with at most $k$ edges. Counting the number of edges requires at most $\mid E \mid$ elementary operations. The trees generation can be done in polynomial time within the number of the structured trees by choosing recursively the initial extremity for each element of $D$.
To prove that Problem DeMo is NP-complete, we propose a polynomial reduction from the **SET-COVER** problem defined as follows:

**Problem:** Set Cover (**SET-COVER)**
**Given:** A set $C = \{c_1, \ldots, c_n\}$ of $n$ elements, a set $S = \{S_1, \ldots, S_m\}$ of $m$ subsets of $C$ such as: $\bigcup_{i=1}^{m} S_i = C$ and an integer $k'$.
**Question:**Does there exist a subset $S'$ of $S$ (called cover of the set C: $\bigcup_{i|S_i \in S'} S_i = C$) such that $|S'| \leq k'$ ?

The problem **SET-COVER** has been shown to be NP-complete in [5].
Consider any instance $I$=$(C, S, k')$ of the Problem **SET-COVER**. We transform this instance to the instance $I'$=$(G, R, D, s, k)$ of the Problem **DeMo**.

1) $G$ is defined as follows:
   - We define the sets of vertices:

     $\zeta = \{c_1, \ldots, c_n\}$, $\varphi = \{S_1, \ldots, S_m\}$, the vertex $s$ and the set $\varphi' = \{S'_1, \ldots, S'_m\}$.
   - We connect by a symmetric edge the vertex $s$ to each vertex of the set $\varphi$.
   - We connect the vertex $c_j$ to the vertex $S_i$ by an edge if and only if $c_j \in S_i$.
   - So, the vertices set of the graph $G$ is composed by: $V = s \cup \zeta \cup \varphi \cup \varphi'$.
   - We consider a complete bipartite graph between $\varphi$ and $\varphi'$.
2) The set of destinations is : $D = \zeta \cup \varphi'$ , and the transmitter is the vertex $s$.
3) The routing $R$ is defined as follows :
   - For $i \in \{1, \ldots, k'\}$ , the route $R(s, S'_i)$ is the shortest path crossing first the arc to $S_i \in \varphi$, then the arc from $S_i$ to $S'_i$;
   - For each couple $(i, j)$ the route $R(S_i, S'_j)$ is the arc that connect them in the complete bipartite graph;
   - For $j \in \{1, \ldots, n\}$ , if and only if $c_j \in S_i$ the arc from $S_i$ to $c_j$ is the route $R(S_i, c_j)$;
   - For each other couples of vertices in the graph $G$, we consider a shortest path routing.
4) We choose $k = k' + m + n$.

It's clear that the number of vertices in the graph $G$ is polynomial considering $|C|$. Thus, we obtain a polynomial reduction $\mathbf{SET-COVER} \rightarrow \mathbf{DeMo}$. This construction is shown in the Figure 4.

Consider now that the answer to Problem **SET-COVER** for $I$ is positive. So, it exists a set $C \subset S$ of $k'$ elements, which cover the set $C$. We define a Broadcast scheme where the tree is induced by the union of the edges of the following routes:

1) The routes of length 1 from $s$ to each vertex $S_j$ such that $S_j \in C$.
2) For each $i$ such that $S_i \notin C$, a route made of length 1 from a vertex $S_j \in C$ to $S'_i$.
3) For each $c_i \in \zeta$, a route made of length 1 from a vertex $S_j \in C$ to $c_i$ such that $c_i \in S_j$.

First, it is clear that this set of routes induces a subtree of $G$ rooted in $s$ with $k' + m + n$ edges, depth 2 and leave set $D$. Secondly, by considering routes from $s$ to vertices $S'_j$ such that $S_j \in C$, and all other edges as routes, we can conclude that this tree defines a broadcast scheme.

Conversely, consider now that the answer to Problem **DeMo** for $I'$ is positive. So, it exists in the graph $G$ a Broadcast scheme of at most $k$ edges with $s$ as transmitter and where each destination (final extremity) is in $D$. Since $\varphi$ is a vertex set disconnecting $s$ from $D$ and since the maximal delay is $m + n + 1$, then the tree can only consists in some path from $s$ to a subset $S_\varphi$ of $\varphi$ and then one arc from a vertex in $S_\varphi$ to each vertex in $D$. Thus, $|S_\varphi| = k'$ and since $S_\varphi$ is also a subset of $S$, the answer is positive for instance $I$ to Problem **SET-COVER**.

We conclude that Problem **DeMo** is NP-complete. Now, let us demonstrate that Problem **DeMo** is not approximable within $c \log(n)$, for some $c > 0$ with $n$ the number of vertices. We consider Problem **DeMo** as an minimization problem. We have seen that any tree in a broadcast scheme for any instance $I'$ described above consists in some path from $s$ to a subset $S_\varphi$ of $\varphi$ and then one arc from a vertex in $S_\varphi$ to each vertex in $D$. Thus, the number of arcs in such a tree is equal to $a + m + n$ and that this number is equal to $opt + m + n$ where $opt$ is the minimum size of a solution for instance $I$ to Problem **SET-COVER** considered as a minimization problem.

Consider that for some constant $c$, Problem **DeMo** is $c$-approximable, i.e., there exists a polynomial algorithm providing a solution size $a \times (m + n + 1) + n$ for $I'$, and a solution of size $a$ for $I$, such that

$\frac{a+m+n}{opt+m+n} \leq c$. Thus, $\frac{a}{opt} \leq c\left(1+\frac{1}{opt}\right) \leq 2c$ , i.e., Problem **SET-COVER** is $2c$-approximable, which is a contradiction with the fact that for some $c > 0$, this problem is not approximable within $c\log(m+n)$ [9]. We conclude the proof of the Theorem 1. ■

**Distributed algorithm to solve Problem DeMo:** We give now more details about our proposed algorithm to achieve an efficient intra-domain handover. A multicast datagrams are composed of two portions: a fixed length header and a data field. The header contains a main destination address considered by the routing function, and a set of secondary destination addresses. An IP router receiving such a multicast packet has to decide if it creates and sends new multicast packets through all the outgoing links or remains on the main destination route. For each empty part on a link, no packet is sent. Note that the complexity of this distributed algorithm is low and that no large memory size is required in routers. Moreover, this algorithm just consists in piloting the usual IP multicast functions in these routers.

The header of a multicast packet $P$ consists in:

- A set of destination $Dest(P)$ and a main destination $Maindest(P) \in Dest(P)$.
- A delay $Delay(P)$ being the maximal number of remaining steps for each destination in $Dest(P)$ to receive the packet.

For each couple of nodes $u$ and $v$ in the graph, let us denote by $succ_R(u,v)$ the successor of $u$ on the route $R(u,v)$. Consider now a node $u$ receiving a multicast packet $P$ at a given step. Any distributed algorithm to solve Problem **DeMo** consists for $u$ to send a multicast packet $P_w$ to each $w \in \Gamma_G^+(u)$ at the next step. Such a distributed algorithm has to respect the following rules. For each node $w = succ_R(u,v)$, let us first define:

$$Sent(w,P) = \{v \in Dest(P) s.t. |R(u,v)| = Delay(P)\}.$$

Then,

1) $Sent(w,P) \subset Dest(P_w)$,
2) $Delay(P_w) = Delay(P) - 1$,
3) if $w = succ_R(u, Maindest(P))$ then $Maindest(P_w) = Maindest(P)$,
4) the set of subsets $\{Dest(P_w)|\ w \in \Gamma_G^+(u)$ and $Dest(P_w) \neq \emptyset\}$ is a partition of $Dest(P)$.

If for any $w$, $Dest(P_w)=\emptyset$ then no packet is sent by $u$ to $w$. Considering $u'$ as the neighbor of $u$ having sent $P_u$ to $u$, we define for any $v \in Dest(P_w)$: $Last(v)=|R(u',v)|$.

We Consider $w_0=succ_R(u, Maindest(P))$. Then for any $w \in \Gamma_G^+(u)$ and respecting the previous rules, the distributed multicast algorithm consists in the following instructions:

**Algorithm 1** Node $U$ receiving a multicast packet $P$

**Require:** Packet $P$; $w = succ_R(u,v)$.
**Ensure:**

```
1: if w ≠ w_0 then
2:    Dest(P_w) = Sent(w, P)
3:    for all v ∈ Dest(P) − ⋃_{w∈Γ_G^+(u)} Sent(w, P) do
4:       if |R(u, v)| ≥ Last(v) then
5:          Put v in Dest(P_{succ_R(u,v)})
6:       end if
7:    end for
8:    for all w ∈ Γ_G^+(u) and any v ∈ Dest(P_w) do
9:       Set Last(v)=|R(u, v)|
10:   end for
11: end if
12: Maindest(P_w)= the least number vertex in Dest(P_w).
13: Dest(P_{w_0})=Dest(P)- ⋃_{w∈Γ_G^+(v)−{w_0}} Dest(P_w)
```

The step number 1 guarantees the respect of the previous rules. The step 2 is the optimization step of the algorithm consisting in identifying on the current branch of the tree, if for any destination $v \in Dest(P) - \bigcup_{w \in \Gamma_G^+(u)} Sent(w,P)$, the current vertex $u$ is or not the much closer vertex from $v$.

To initiate the process, we consider that the transmitter $s$ has received a packet $P$ with a node in $D$ such that the distance from $s$ to this node is maximum (this distance initializes $Delay(P)$). This node will be the main destination and all the other nodes in $D$ compose the secondary destinations. For any $v \in Dest(P)$, we also set $Last(v)=|R(s,v)|$.

### C. Alert Algorithm

This algorithm is based on works proposed in [19] for inter-domain network. We adapt this routing alert algorithm for intra-domain network by introducing a *hierarchy concept* in the TWINBOARD network similar to the one used in inter-domain network. Alert algorithm uses the existing intra-domain routing protocol functionalities. It acts directly on routing tables by disabling, activating or replacing routes. Our goal is to provide routers with information regarding the congestion state of other routers without any change in the routing protocol. We say that a router is:

- Perturbed or in *red state*: if the total amount of traffic transitting through it, emitted by it, and sending to it exceeds its capacity.
- Stable or in *green state* : if its capacity exceeds its traffic loads.

Each router informs its neighbors when it becomes perturbed in order to allow them to change their routing and it also keeps them informed when it returns to an operational state. Each router is provided with:

- $RT_i$ containing the next hop for the intra-domain routing protocol.
- Routing table $LT_i$ containing a lists of the next hops towards every destination. This table is altered by classical intra-domain routing mechanism only.
- Priority table $PT_i$ storing a list of potential congestioned-free routes. It is same as the $RT_i$ table if there are no perturbed routers.
- State Table $st_i$ containing states of its neighbors. This table is updated by alerts sent by neighboring routers when their states change. The default values in this table are *green*.

The alert message is composed of: an identifier of the router (ID), a new state, and a delay $d$. Once in the router scheduler, the received message will be processed after $d$ delay (unit of time) and will replace any older message that arrived from the same emitting router. The delay is set according to an exponential distribution in order to avoid synchronization in the network. The mean of the distribution is small for *red* alert and big for *green* alert. By this way, we limit the emergence of oscillations that can occur by exchanges of *green* and *red* messages.

Algorithm 2 details the behavior of a router $i$ treating a packet $p$ received from the node $j$. First, the router $i$ selects destinations for, which the router $j$ (in *red* state) is the next hop. The router $i$ then chooses uniformly the alternative next hops among routers in *green* state and stored in the set $LT_i[dest]$. If no node can be chosen, the node $i$ selects the routing path stored in $RT_i$ table in spite of its state.

## IV. Simulator description

In this section we simulate the behavior of the TWINBOARD network in case of intra-domain handovers by combining the two algorithms: Multicast and Alert Algorithms. The two proposed algorithms are compatibles with the embedded IP routing. The purpose of our simulation is to show how the proposed scheme can be adopted and compatible with the existing IP routing protocols.

**Algorithm 2** Node $i$ treating a message $p$

**Require:** message $p = (j, state, delay)$; tables $RT_i$, $PT_i$, $LT_i$, $st_i$
**Ensure:**
delete $p$ from the scheduler
$st_i[j] \Leftarrow state$
**if** $state$ = $red$ **then**
  **for all** $dest \in V$ **do**
    **if** $PT_i[dest] = j$ **then**
      **let** $S = \{x \in LT_i[dest] / st_i[x] = green\}$
      **if** $S = \emptyset$ **then**
        $PT_i[dest] \Leftarrow RT_i[dest]$
      **else**
        $PT_i[dest] \Leftarrow$ *choose_uniformly_in(S)*
      **end if**
    **end if**
  **end for**
**else**
  **for all** $dest \in V$ **do**
    **if** $st_i[PT_i[dest]] = red$ **then**
      **if** $j \in LT_i[dest]$ **then**
        $PT_i[dest] \Leftarrow j$
      **end if**
    **if** $RT_i[dest] = j$ **then**
      $PT_i[dest] \Leftarrow j$
  **end for**
**end if**

### A. Simulation environment

In order to prove our concepts and the advantages of our algorithms, we have developed a simulator using OMNeT++ [21], which is a free, open-source discrete event simulation tool, similar to other tools like PARSEC, NS, or commercial products like OPNET. Otherwise, OMNeT++ contains definitions of many popular protocols (UDP, TCP, IPv4, IPv6) as well as models of basic network nodes (routers, hubs, access points, gateway, base station etc.) that help us on modelling intra-domain mobility without worrying about the underlying mechanisms.

### B. Topology and traffic Assumptions

The simulated network has been modelled according the TWINBOARD architecture as shown in Figure 2. The main elements are Base Station (BS) and the Access Gateway (AG) interconnected by a network of basic IP routers. We used topologies with 80 nodes where the number of Gateway, Base Stations and their degree of connectivity are considered as parameters of simulation:

- The degree of the gateway is chosen greater than 1.
- The degree of the BS is equal to 1 (Access Points).

- The topology of the IP routers network is generated by the BRITE generator [22].

The traffic is generated by the set of BS (sources) then it flows through the IP routers to reach the gateway to be finally multicasted to the Base Stations members of the multicast group. $TV_i(j)$ represents the traffic amount to be delivered from the source $BS_i$ to the destination $BS_j$ member of the multicast group. The traffic amount $TV_i(j) = \frac{\alpha}{|V_p|^2}\mathcal{N}(10, 8)$ generated by the $BS_i$ is a random quantities given by a normal distribution with the mean equal to 10 and the standard deviation equal to 8.

The parameter $\alpha$ is the *flow's desired traffic parameter* used to vary the level of the traffic in the network. To simulate the intra-domain handovers event, we vary periodically the traffic matrix as follows:

- We choose randomly and with certain probability $P_{ij}$ two traffic matrix elements $TV_i(j)$ and $TV_j(i)$.
- We reduce the value of the traffic matrix element $TV_i(j)$ of a certain quantity $q$.
- We increase the value of the traffic matrix element $TV_j(i)$ of a certain quantity $q$.

Indeed, we simulate the user handovers by a tolerate traffic matrix fluctuations. Although networks are engineered to tolerate some variation in the traffic matrix, large changes can lead to congested links and break the assumptions used in most designs. To simulate realistic scenarios, the cellular network (set of BSs) is represented by a random planar graph with degree equal to 6.

### C. Simulator Modules

Our simulator deals with the IPv6 Mobility Extention. We can easily build different network scenarios by providing a few simple parameters from, which the simulator constructs the network automatically.

According to OMNeT++, the structure of our simulator is modular. We defined the modules and implemented their functions in C++. The main modules are:

- **Gateway:** The intelligent component implements the mobile extention management (Multicast and Alert Algorithms).
- **Base Station:** These elements represents all physical radio access points belonging to the same network.
- **Router:** This component stands for the whole wired network between Base Station, servers and Gateways. It is responsible for routing packets and simulates network delays as well.

## V. PERFORMANCE ANALYSIS

With the limited charge capacity of routers and network overhead the optimization of multicast and the Alert mechanism are very important. We simulated three scenarios by varying the level of the traffic flowing through the network. This controlled-load function is invoked by specifying the parameter $\alpha$ (described above):

- The first scenario depicts an unloaded network obtained with the value $\alpha$=0.06. This value engenders a minor traffic fluctuations.
- The second scenario is given by an average network load ($\alpha$=0.13). This value increases the network load of 200% compared to the one in unloaded network.
- The third scenario corresponds to a heavily loaded network ($\alpha$=0.5) enough to saturate 20% of routers in the network.

Note that these three values of $\alpha$ are obtained by experiences.

For each scenario, we investigate the following statistics to evaluate the performance of our approach and compare it to the DVMRP (Distance Vector Multicast Routing Protocol) combined with the Alert algorithm :

- *Number of perturbed routers*: A router is *perturbed* if its traffic loads exceeds its capacity.
- *Number of perturbed routes*: A *perturbed route* is a route containing at least one perturbed node.
- *Volume of a perturbed traffic*: Is the sum of traffic passing on perturbed routes. Traffic from $i$ to $j$ is perturbed if the route from $i$ to $j$ is perturbed.

We chose DVMRP protocol because it is an $Interior\ Gateway\ Protocol$ (IGP); suitable for use within an Autonomous System (AS), but not between different ASs. DVMRP provides an efficient mechanism for connectionless message multicast to a group of hosts.

To verify the functioning of our approach we performed three series of simulation runs on the same network topologies (50 routers) for three values of $\alpha$. All the graphs follow a common format. Each one shows simulation statistics obtained by the proposed and DVMRP algorithms (combined with Alert algorithm) for the three scenarios mentionned above. In what follows, we denote by **standard approach** the combination of the two algorithms: DVMRP and Alert. While proposed approach will denote the combination of the multicast algorithm with the Alert mechanism.

**Figure 5** illustrates simulation results for a heavily

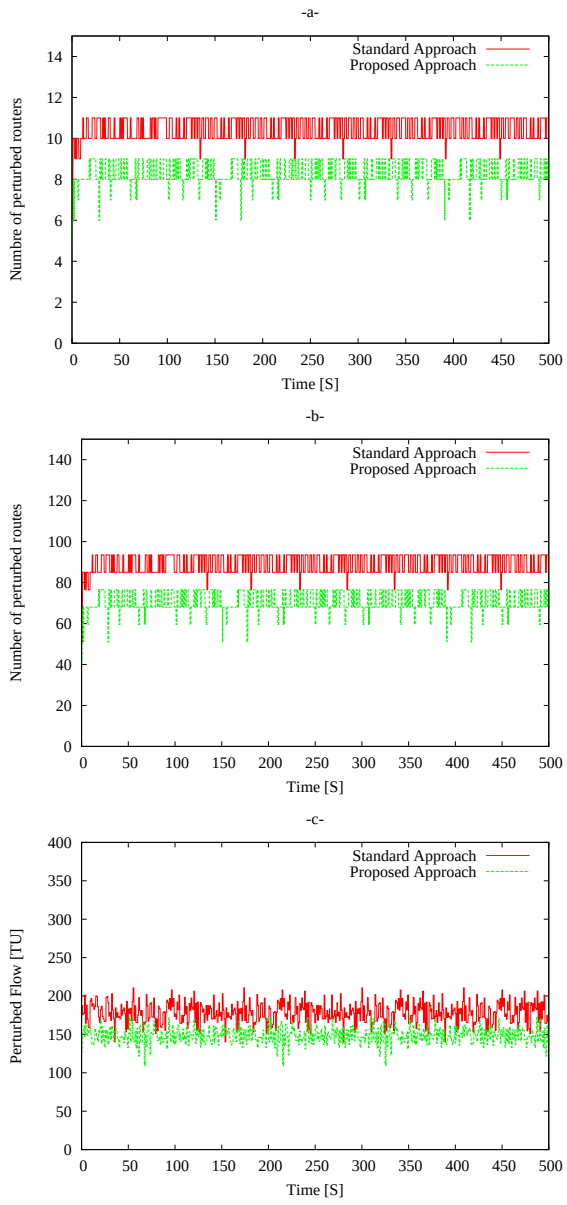



Fig. 5. Simulation results for 80 nodes network and high network charge (one simulation run, $\alpha$=0.5).

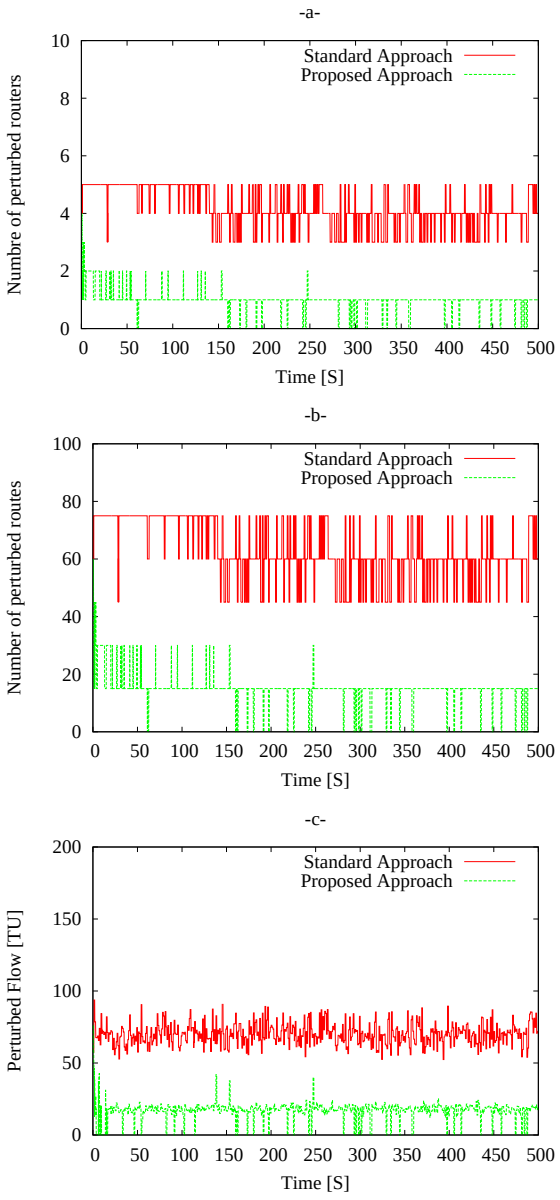



Fig. 6. Simulation results for 80 nodes network and average network charge (one simulation run, $\alpha$=0.13).

loaded network ($\alpha$=0.5). Figure 5.a shows that the number of perturbed routers obtained with the standard approach reaches eleven routers when the traffic matrix is filled up and it oscillates around this values. Our approach reduces this value to oscillates around eight routers and it reaches six routers in the best cases. We observe that for the traffic value $\alpha$=0.5 almost 20% of the routers are perturbed. In this case the performance of our algorithms approaches those of DVMRP. The same analysis can be done for the Figures 5.b and 3.c. Indeed, because of the high network load, our algorithms can not find substitute routes to achieve an efficient load-balancing and consequently it can not reduce the number of perturbed routers, perturbed routes and the quantity of the perturbed flow.

For an average network charge ($\alpha$=0.13), **Figures 6** show that our approach results are significantly better than those obtained by the standard approach. The numbers of perturbed routers, perturbed routes and the quantity of the perturbed flow, presented in **Figure 6.a**, **Figure 6.b** and **Figure 6.c** , exhibit that our algorithm presents performance four times better than DVMRP. In these figures, the spades correspond to the fluctuations

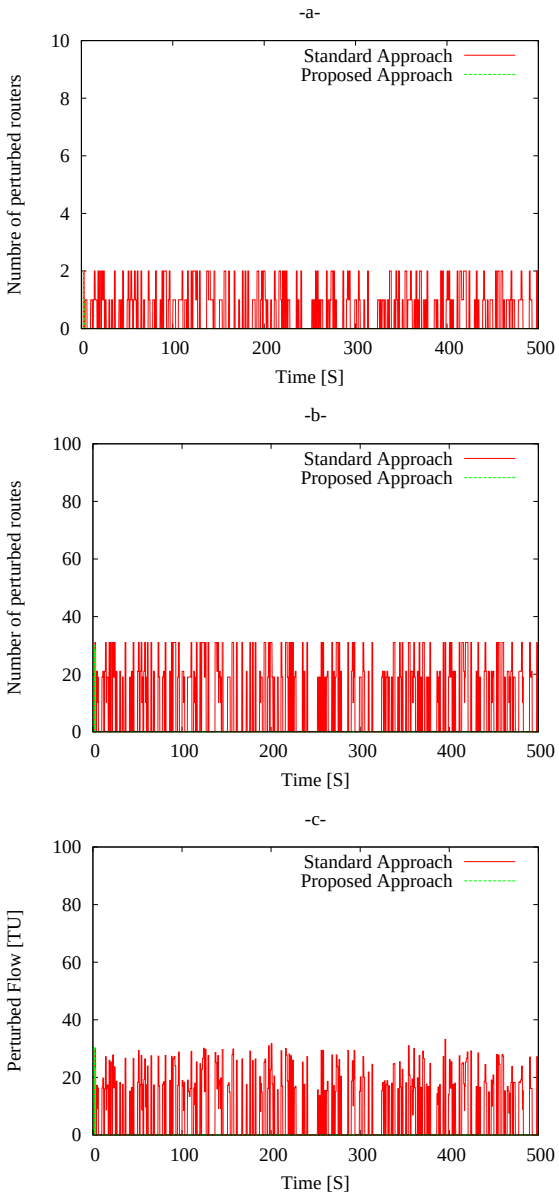



Fig. 7. Simulation results for 80 nodes network and low network charge (one simulation run, $\alpha$=0.06).

in the traffic that provoke additional router saturation. Our algorithms react by finding a new (better) routes to reduce immediately the number of perturbed routers. In **Figure 6.c**, we observe that our algorithms reduce the quantity of perturbed flow to a third.

The graphs in **Figure 7**, show that even in unloaded network ($\alpha$=0.06), the standard approach presents its limits. Since we register in some cases two perturbed nodes. On the opposite, our proposed approach finds quickly an efficient routing with no perturbed routers even with the traffic fluctuations.

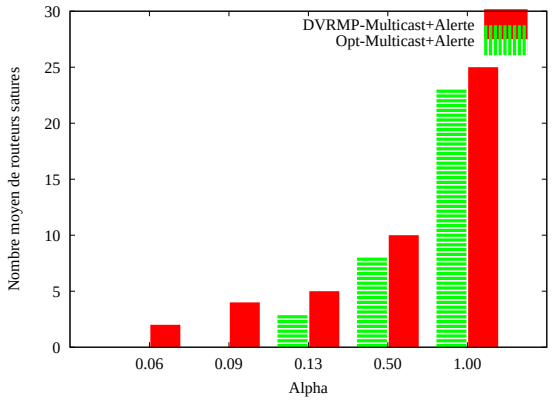



Fig. 8. The number of perturbed routers depending on the traffic load averaged for series of simulations runs.

To investigate the influence of the traffic load on the performance of our approach, we used averaged series of simulation runs (100 network topologies). Each series had a different load by specifying the parameter $\alpha$, which varies from light load $\alpha = 0.06$ to saturating load $\alpha = 1$. Figure 8 shows that for the saturating load value almost half of the network routers are perturbed. In this case the performance of our algorithms approaches those of the standard approach. The average values in Figure 8 confirm those shown in Figures 5, 6 and 7. Clearly, the performance of our approach decrease when the the parameter $\alpha$ increases.

## VI. Conclusion

We have presented a novel approach to manage IP micro-mobility using intra-domain multicast with alert mechanism. Our algorithms achieve efficiently two major functions in the mobile network: mobility management and network congestion avoiding. In terms of multicast performance, our algorithm achieves an optimized multicast in terms of the number of the used links. Also, it provides minimal break in service since it is based on handovers prediction.

In this paper, and with the context and the goals of TWINBOARD project, our simulation results show that with a good and a priori established traffic engineering, our proposed approach performs a reliable intra-domain handover with congestion avoiding. Our multicast algorithm outperforms DVMRP because of its minimal number of perturbed routers.

In the future, we plan to conduct further simulations to investigate real handover scenarios. We also would like to investigate the extention of our approach to support dynamic multicast group membership.

## REFERENCES

[1] Y. Benallouche and D. Barth, "Optimized multicast tree for handover in a two-nodes mobile network architecture based on an all-IP infrastructure," *Proceedings of the Fourth International Conference on Wireless and Mobile Communications (ICWMC'08), Athens, 2008, pp. 179-184*.

[2] A. Ballardie, "Core Based Trees (CBT Version 2) Multicast Routing:Protocol Specification," IETF, Internet Draft, RFC 2189, September 1997.

[3] S. Deering, "Protocol Independent Multicast-Dense Mode (PIM-DM):Protocol Specification," IETF, Internet Draft, RFC 2365, July 1998.

[4] D. Estrin, "Protocol Independent Multicast-Sparse Mode (PIM-SM):Protocol Specification," IETF, Internet Draft, RFC 2362, June 1998.

[5] M. R. Garey and D. S. Johnson, "Computers and intractability. A guide to the theory of NP-completness," San Francisco, W. H. Freeman, 1979.

[6] A. Helmy, M. Jaseemuddin and Ganesha Bhaskara, "Efficient Micro-Mobility using Intra-domain Multicast-based Mechanisms (M&M)," In ACM SIGCOMM Computer Communication Review, Vol. 32, No. 5, 2002, pp. 61-72.

[7] A. Helmy, "A Multicast-based Protocol for IP Mobility Support," In ACM SIGCOMM Second International Workshop on networked Group Communication (NGC 2000), Novembre 2000.

[8] N.K. Karthirkeyab and P. Narayanasamy, "A Novel Mobility Based Bandwith Reduction Algorithm in Cellular Mobile Network," *In International Journal of Computer Science and Network Security, Vol.8 ,No.5, 2008, pp. 62-69*.

[9] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *In Journal of the ACM (JACM), v.41 n.5, p.960-981, Sept. 1994*.

[10] J.Moy, "Multicast Extension to OSPF," IETF, Internet Draft, RFC 1584, March 1994.

[11] C. Perkins, "IP Mobility Support," RFC 2002, Internet Engineering Task Force, October 1996.

[12] G.N. Rouskas and I. Baldine, "Multicast Routing with end-to-end delay and delay variation constraints," *In IEEE JSAC, April 1997, pp. 346-356*.

[13] T.C. Schmidt and M. Wählisch, "Analysis of Handover Frequencies for Predictive, Reactive and Proxy Schemes and their Implications on IPv6 and Multicast Mobility," *Proceedings of the fourth International Conference on Networking (Networking – ICN 2005), Reunion Island, 2005, Part II, Lecture Notes in Computer Science 3421, pp. 1039–1046*.

[14] T.C. Schmidt and M. Wählisch, "Performance Analysis of Multicast Mobility in a Hierarchical Mobile IP Proxy Environment," *Proceedings of the TERENA Networking Conference, 2004*.

[15] W. Su, S.J. Lee and M.Gerla, "Mobility Prediction and Routing in Ad hoc Wireless Network," *In International Journal of Network Management, vol.11(1), 2001, pp.3-30*.

[16] K. Suh, D.-H. Kwon, Y.-J. Suh, and Y. Park, "Fast Multicast Protocol for Mobile IPv6 in the fast handovers environments," IETF, Internet Draft, February 2004.

[17] B. Wang and J.C. Hou. "Multicast Routing and its QoS Extension:Problems, Algorithms, and Protocols," IEEE Networks, January/February 2000.

[18] D. Weitzman and C.Partridge, "Distance Vector Multicast Routing Protocol," IETF, Internet Draft, RFC 1075, November 1998.

[19] M. Weisser, J. Tomasik and D. Barth, "Congestion Avoiding Mechanism Based on inter-domain Hierarchy," In Springer Berlin/Heidelberg, editor, IFIP Networking, volume 4982, pp. 470-481. LNCS, 2008.

[20] The Next Generation Mobile Networks project WEB pages. See URL http://www.ngmn-cooperation.com.

[21] The OMNET++ Simulator. See URL http://www.omnetpp.org.

[22] The Brite Generator. See URL http://www.cs.bu.edu/brite/.

[23] WiMAX Forum. WiMAX Forum Mobile System Profile. 06-07.

# Analysis of Communication Overhead for a Clustering-Based Security Protocol in Ad Hoc Networks

C. Maghmoumi
University of Haute-Alsace
F-68000 Colmar, France
chadi.maghmoumi@uha.fr

H. Abouaissa
University of Haute-Alsace
F-68000 Colmar, France
abdelhafid.abouaissa@uha.fr

J. Gaber
Belfort University
F-90000 Belfort, France
jaafar.gaber@utbm.fr

P. Lorenz
University of Haute-Alsace
F-68000 Colmar, France
pascal.lorenz@uha.fr

***Abstract*— In this paper, we study and evaluate the overhead for a security algorithm based on clustering in MANET networks. The analysis of the communication overhead in Ad Hoc networks is an important issue because it affects the energy consumption and the limited battery life time of the mobile nodes. The algorithm partitions the network into clusters based on affinity relationships between nodes and two types of keys which are generated by a clusterhead. The first one is shared by a clusterhead and its local members and the second one is shared by the clusterhead and its parent cluster. The performance evaluation and communication overhead analysis of the proposed protocol are presented using simulation.**

***Keywords: ad hoc networks, clustering, security, key management.***

## I. INTRODUCTION

Moving from wired networks to wireless mobile networks leads to use pervasive networks with new network-computing challenges. Ubiquitous Computing (UC) is a concept that deals with providing available services in a network by giving users the ability to access services anytime and irrespective to their location. Pervasive Computing (PC) is often considered as the same as ubiquitous computing but the main objective in PC is to offer spontaneous services created on the fly by mobiles nodes that interact with ad hoc connections [1].

Mobile ad hoc networks (MANET) are autonomous systems created by mobile nodes without any infrastructure. Currently, MANET has gained popularity in mobile pervasive applications, such as electronic business, emergency teams, etc. These applications support group communications, auto-adaptive discovery and composition services. Most of research on the pervasive communications in MANET mainly focused on admission control and resource management (like bandwidth, energy consumption, interferences, etc.) to perform the communication in these mobile networks. However, in these types of applications, secure group communications is very critical and is a major concern.

Clustering in MANET is a challenging issue because of the dynamic network topology changes. Clustering algorithm partitions a network into different clusters, creating a network hierarchy in the network. In general, clustering algorithms can be divided into cluster formation stage and cluster maintenance stage. A particular node is elected in a cluster to manage the cluster information is known as the clusterhead, and the other nodes are its members.

In MANET, to ensure a confidential communication between two or several mobile nodes, traffic can be encrypted and only receivers can decrypt data [2, 3]. Furthermore, MANET may be highly versatile, involving short-lived communications between nodes that may never have met before, thus complicating the initial trust establishment and trust maintenance. Thus, new solutions should be introduced to support efficient and secure group communication in mobiles pervasive networks with respect to the dynamic network topology induced by the node mobility and unreliable communication. Also, this type of network does not have any trust node for key management, like a central reference, to ensure the message encryption/decryption. This cannot actually satisfy MANET dynamic environments. To solve this problem, one of the approaches is to share a secret key called "group key" [4]. When a member joins a group, the group key is rekeyed to ensure that the new member cannot decrypt previous messages, a security requirement known as backward secrecy [5]. When a member leaves the group, the group key is re-keyed to ensure that future communications cannot be decrypted by the leaving member, a security requirement known as forward secrecy.

In MANET, it is not easy to control mobile members of a cluster and the frequency of their adhesions. The security algorithm must support the mobility problem and the clusters' scalability. Therefore, to solve these problems and ensure trusted communications in a MANET environment, the major solution is to introduce an efficient key management algorithm, adequate to manage and distribute keys to cluster members in order to encrypt/decrypt multicast

data. An efficient security algorithm should provide a rapid re-keying process and be adaptive to frequent topology changes.

The analysis of communication overhead in Ad Hoc network is related to different parameters, e.g. network size, node mobility, node transmission range and network density. An efficient clustering and key management algorithm must support all these network parameters in order to minimize the messages overhead.

The rest of the paper is organized as follows. Section II overviews the related work. In section III, we present an overview on security in ad hoc networks. Section IV introduces the proposed key management protocol. Section V specifies proofs for the proposed protocol. In Section VI, we implemented and evaluated the proposed protocol and section VII draws conclusions and future works.

## II. RELATED WORK

Recently, many clustering algorithms have been proposed for mobile ad hoc networks in order to improve the efficiency of routing protocols and save energy or to implement efficient flooding and broadcasting mechanisms. Haddad and Kheddouci presented in [6] a classification of topology-based approaches to define an efficient organization over the network to optimize communication protocols for routing, service discovery, resource sharing and management.

Many group security algorithms or protocols have been proposed for MANET in the literature. They can be divided into two categories: centralized and distributed protocols [11]. In centralized protocol, only a single node controls all the other nodes. Therefore, the re-encryption process is managed only by this node. This protocol can optimize network resources. However, since there is only one key manager in the group, it is probable that this node breaks down [12, 13]. In [14], the authors proposed two key agreement protocols based on the threshold cryptography using the Lagrange interpolation theorem. This approach seems theoretically efficient; however, it focuses only on a special case of scenario. In [7], a hierarchical protocol based on multicast source key is proposed. The source node provides keys to its local members and to groups' leaders. A new node that will be joined to a group should negotiate with group's leader, then, the latter informs the source node to get a new key. The only role of leaders is to manage received keys from the source node. Although, the protocol secures the network, its complexity is high due to multiple key's generation to maintain group communication security. Luo et al. [17], [18] chose a different method to distribute the certification process. They use a specially crafted key sharing algorithm distributing the key amongst all nodes instead of a subset only. Upon this, Luo et al. build an access control system based on signed tickets issued by neighbors of the node seeking access.

In [15], the authors proposed an analysis of the overhead involved in clustering for one-hop clustered ad hoc networks. This analysis captures the effects of different network parameters, e.g. node mobility, node transmission range, and network density on the amount of overhead that clustering algorithms may incur in different network environments. But the authors in this analysis focused mainly on the cluster maintenance stage and only one-hop clustering algorithms are considered. In [16], the authors introduced a cluster-based architecture for a distributed public key infrastructure. This architecture is adapted to the highly dynamic topology and varying link qualities in ad hoc networks but the overhead is very high.

The proposed protocol in this paper differs from previous studies in three ways:

1) We don't require any centralized key control component to manage and distribute keys. Encryption keys are generated by clusterhead and re-encrypted by participating sub-clusterheads.

2) A dynamic clustering algorithm that is adaptive to frequent topology changes is used.

3) Since the key distribution process is totally decentralized and the keys are shared by different communication groups, the proposed protocol can be used to build a generic security service for multicast communication.

## III. SECURITY IN AD HOC NETWORKS

The security of a multicast group requires that only group members can access the data transmitted by the source, even if these data are diffused in the network. To ensure this confidentiality, a symmetric key is used by the source to encrypt data, and by the members to decrypt data. This key is called TEK (Traffic Encryption Key). The life of a session of a secure group is represented by a set of time intervals, each interval is defined by a change in the status of the group (join or leave a member) as shown in figure 1. To preserve data confidentiality of the group, it is necessary to renew the encryption key after each event (join or leave the group). A member who leaves the group should no longer be able to decrypt data.

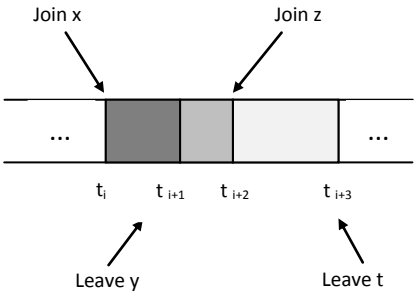



Figure 1. Life's changes of a secured group.

### A. Classification of group key management approaches

Several architectures for group key management in networks have been proposed and developed; we can classify them into three approaches according to the number of TEKs used as shown in figure 2.

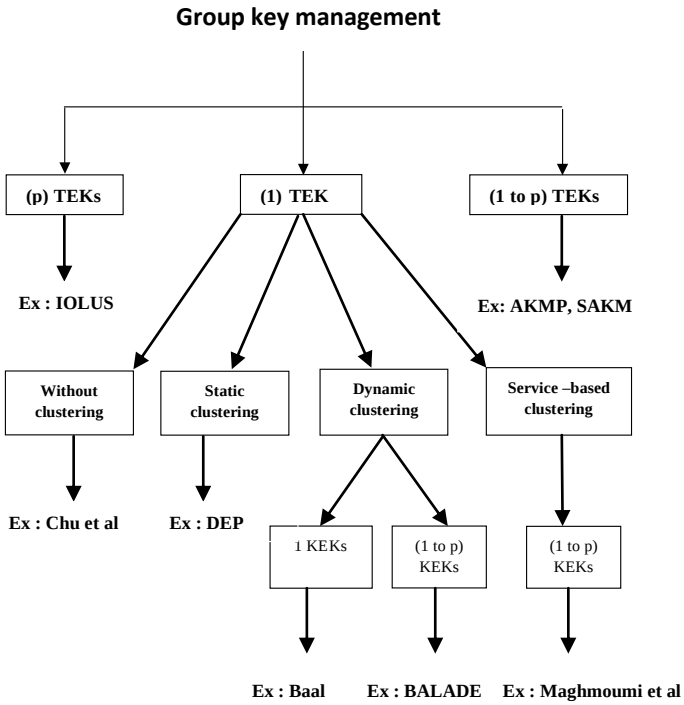



Figure 2. Classification of group key management protocols.

In the approach p TEK (Traffic Encryption Key), each subgroup shares a local TEK generated by a local controller, p is the number of sub-groups of the multicast group. This hierarchical approach is to address the problem 1 affects n. At each arrival/departure to/from the group, only the subgroup affected by this change will change its local TEK. In the approach (1 to p TEKS) the protocol begin with a centralized approach (1 affects n) and dynamically switches towards a subdivision of the group into subgroups (p TEK). The approach (1 TEK) is to share a one encryption key TEK used by the source and members in order to encrypt/decrypt multicast data, this approach can be used in a centralized architecture (no clustering) or hierarchical ( static or dynamic clustering). Table 1 shows the classification of different protocols from the literature.

| Protocol | Number of KEK | Clustering |
|---|---|---|
| IOLUS | p | static |
| AKMP | 1 to p | dynamic |
| SAKM | 1 to p | dynamic |
| Chu et al | 1 | no clustering |
| DEP | p | static |
| Baal | 1 | dynamic |
| BALADE | 1 to p | dynamic |
| Maghmoumi et al | 1 to p | dynamic |

Table. 1. Comparison of group Key Management protocols

## IV. CLUSTERING BASED SECURITY PROTOCOL

### A. Clustering approach

As the topology of a MANET changes, clustering messages are generated by nodes to update a node of changes to its cluster members or clusterhead. The execution of clustering algorithms can usually be divided into cluster formation stage and cluster maintenance stage [20, 21]. Different clustering protocols may use different schemes and generally there are three types of clustering messages:

a) **Join message**, for nodes to know the neighbor's identities. The HELLO message is often used.

b) **Acknowledgement message** to accept new node in the cluster.

c) **Leave message** to remove a node from a cluster.

In what follows, for the clustering overhead analysis, we denote the network size by N, a cluster size by $NC_i$ (the number of members in the cluster $C_i$), the network density by ρ, and the transmission range is r. The average cluster size $NC_i$ is given by $NC_i = N/n$ where n is the number of clusters in the network.

Two properties for clustered networks should be ensured and any violation will trigger clustering messages at relevant nodes [15]:

P1. No cluster-heads directly connected to each other.

P2. Each node should belong to only one cluster.

The main idea underlying this protocol is to divide the ad hoc network into clusters according to affinity relationships between involved nodes [8] and uses the Key Management Protocol proposed initially in [9]. Once the clusterhead is selected, it handles two KEKs (Key Encryption Key), one shared by clusterhead and its local members, and the second is shared by the clusterhead and the parent cluster.

Affinity relationships between the nodes can be determined according to the services they provide. A service can be described by four main parts [8]:

1) the attributes.
2) the capsule.
3) constraints and requirements.
4) set of relevant semantic keywords.

Attributes contain the characteristic of the service, such as operations that can be invoked and their input and output parameters. The capsule includes information about the service localization, the invocation protocol and the port number. The constraints and requirements give information about the resources needed to execute the service. The set of semantic keywords are used by for matching relevant keywords to each nearby service. [1].

Ad hoc networks are characterized by the node's mobility of nodes; several nodes can move with different speeds. Our goal is to form stable clusters; in this case, we set a given threshold to separate the clusters formed by high-speed or slow nodes. In basic ad hoc networks, nodes can exchange [RTS, CTS, DATA, and ACK] messages, via a complete virtual graph, in order to guarantee group self-stability by the homogeneous mobility of nodes and thus ensure a reliable communication between wireless mobile nodes.

In this protocol, the source node generates a TEK encrypted in a Key Encryption Key $KEK_i$ that should be sent to its local members. Once, each clusterhead receives the encrypted data, it decrypts it and re-encrypts it with its own $KEK_j$, then forwards it to its descendents. The join or leave events within each cluster results in the $KEK_j$ re-keying by the clusterhead. Therefore, the proposed protocol belongs to the dynamic clustering algorithm with one TEK and *1* to *p* KEK, where *p* is the number of clusters that constitute the ad hoc network. That makes it possible to optimize the cost of data encrypting and decrypting processes and to reduce the 1 affects n phenomena [10].

The clustering-based key management protocol consists of two tasks:

*1) Cluster Formation*

- The cluster formation starts when a node $N_i$ boots up and sends a cluster join message JOINreq to its neighbors. This message contains the description of a service D ($S_i$) and a number $ID_i$ that identifies this service.

- When a node $N_j$ receives the message JOINreq (D ($S_i$), $ID_i$), it examines the compatibility of the service $S_i$ with a service $S_j$ using MATCH (D ($S_i$), D ($S_j$)) and sends a response message Rep to $N_i$ that contains the rate of available energy f($E_j$) expressed by (1) [19]:

$$f(E) = \frac{E_{max} - E_{cons}}{E_{max}} \qquad (1)$$

$$E_{cons} = E_{cons} + E_{req} + \varepsilon \quad ; \varepsilon \geq 0$$

Where $E_{max}$ is the maximum energy of the node, Econs is is the energy consumed, $E_{req}$ is the energy required to transmit a packet and ε is the energy that can be lost in the environment due to factors not anticipated [22].

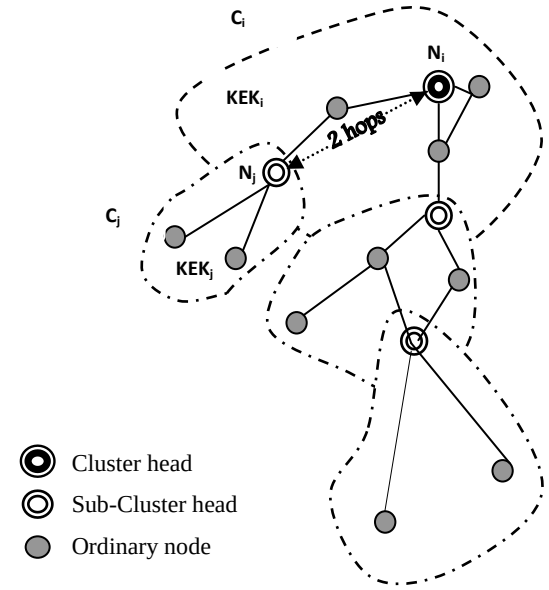



Figure 3. Ad hoc network partitioned into in 4 clusters

- When the node $N_i$ receives the response message Rep (ID, f ($E_j$)), it verifies the RTT (Round Trip Time) and the f($E_j$) and sends back an acknowledgement message ACK containing a flag that indicates the validation of theses parameters. Then, it adds the node $N_j$ to the list of the cluster Ci's members that consider $N_i$ as clusterhead (the first member of the list). Once the clusterhead is chosen, it generates a KEK that should be sent to its local members.

- If a member $N_j$ is at two hops from the clusterhead and if there are at least two nodes belonging to the cluster via $N_j$, then $N_j$ becomes a clusterhead for a sub-cluster that contains these nodes. It generates thus a $KEK_j$ for its own cluster as shown in figure 3.

The formal description of the protocol process is described in figure 4.

**When** a node $N_j$ receives JOINreq (D ($S_i$), $ID_s$) **then**
$m_{ij}$ = MATCH(D ($S_i$),D ($S_j$))
**if** $m_{ij} \geq \sigma$ **then**
send RepN ($ID_i$, $ID_s$,f(E))
**fi**

**When** a node $N_i$ receives RepN ($ID_i$, $ID_s$,$f$(E)) **then**
**if** RTT $\leq \beta$ **then**
**if** $f(E) \leq \alpha$ **then**
send ACK ($ID_j$, $ID_s$, non)
**else**
send ACK ($ID_j$, $ID_s$, ok)
$CL_i = \{CL_i \cup N_j\}$
Send ($\{CL_i\}$, $KEK_i$)
**fi**
**fi**

**When** a node $N_j$ receives ($\{Cl_i\}$, $KEK_i$) **then** $CL_j = CL_i$

**When** a node $N_j$ is at 2 hops from the clusterhead **then**
**if** $|H| \geq 2$ **then**
$CL_j = \{N_j \cup \{H\}\}$
send ($\{CL_j\}$, $KEK_j$)
**fi**
// $N_i \in \{H\}$ => $N_i$ is at 3 hops at least from the clusterhead // and $N_i$ received the ACK via $N_j$

**When** a node $N_i$ receives Leav$N_j$ ($ID_s$) **then**
$CL_i = \{CL_i \setminus N_j\}$
send($\{Cl_i\}$, $KEK_i$)

**When** a node $N_j$ receives Leav$H_i$ ($ID_s$) **then**
**If** $N_j \in \{CL_i\}$ then
send JOINreq (D ($S_j$), $ID_s$)
**else**
$CL_j = \{CL_j \setminus N_i\}$
send ($\{CL_j\}$, $KEK_j$)

Figure 4. Clustering-based security protocol

Consequently, each clusterhead handles two KEKs:
1. $KEK_i$: shared between clusterhead and its local members.
2. $KEK_j$: shared between the clusterhead and its parent cluster.

*2) Cluster Maintenance*

- When a member leaves the cluster $C_i$, it sends a leave message to the clusterhead which removes it from its list of nodes $CL_i$, regenerates a new KEK and transmits it to its local members except the departing member. In this step the number of messages sent is $NC_i - 1$.Where $NC_i$ is the number of nodes in the cluster $C_i$.

- If a clusterhead $N_j$ leaves the cluster, it sends to its members a leave message in addition to its parents cluster members. When the clusterhead of the parent cluster $N_i$ receives this message, it removes it from the list, regenerates a new $KEK_i$ and transmits it to its local members except the departing member. Each member of the leaving clusterhead send join message JOINreq to its neighbors. In this step the number of messages sent is $NC_i + NC_i - 2$.

The total number of messages is thus:

$$M_{leav} = 2NC_i + NC_j - 3 \quad (2)$$

- When a new node $N_j$ joins a cluster, it sends messages to its neighbors. The clusterhead that accepts this node regenerates a new KEK for its new list of nodes. The number of messages sent in this step is:

$$M_{join} = NC_i + \mu \quad (3)$$

Where μ is the expected number of network neighbors of a randomly node that depends on ρ, r and N.

The figure 5 shows the process of joining the cluster and the figure 6 shows the process of leaving the cluster.

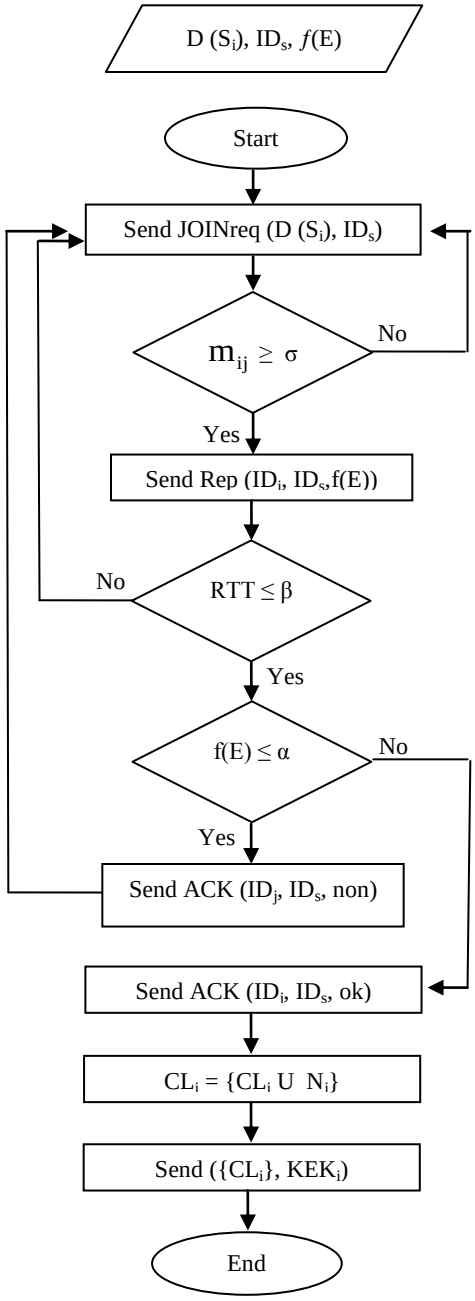



Figure 5. Process of joining the cluster

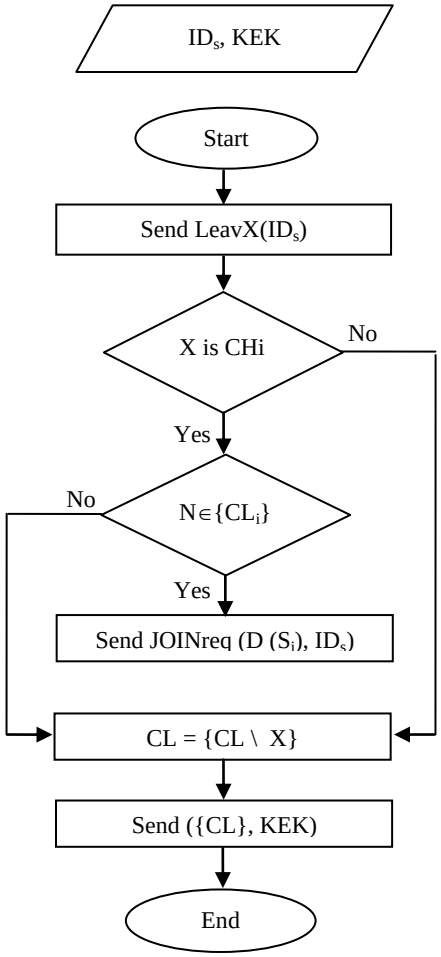



Figure 6. Process of leaving the cluster

*B. Correctness proof*

In this part, we explain the expected value of the number of network neighbors of a randomly chosen node μ and we analyze the capabilities of our protocol that ensures communication trust when a node joins or leaves the cluster.

*Lemma 1:*

The expected number of network neighbors of a randomly selected node is:

$$\mu = O((N-1)\frac{r^2}{S}) \qquad (4)$$

*Proof:*

The expected number of network neighbors of a randomly selected node is given in [15]:

$$\mu = (N-1)\frac{r^2\rho}{N}\left(\frac{r^2}{2}\frac{\rho}{N} - \frac{8r}{3}\sqrt{\frac{\rho}{N}} + \pi\right)$$

within an area S with density ρ, we obtain equation (4)

*Lemma 2:*

The new node that joins the cluster cannot decrypt past encrypted data.

*Proof:*

Assume that a new node $N_i$ sends a cluster join message to clusterhead. $N_i$ cannot decrypt messages because any node cannot decrypt data as long as it does not receive the acknowledgment from its clusterhead. In fact, when the clusterhead receives a new join message, it updates the list of members and regenerates a new key $KEK_i$, then sends it to its local members. The proof could drive to that every node must have a KEK key to decrypt and encrypt data traffic which proves that the proposed protocol guarantees the backward secrecy.

*Lemma 3:*

The node which leaves the cluster cannot decrypt the future data.

*Proof:*

Leaving of ordinary node from a cluster is uncomplicated. The node sends a leave message to the clusterhead that leaves this node from the list of members and regenerates a new KEK broadcasted to all local members in the new list. However, when a clusterhead wants to leave the network, it must inform the upper clusterhead to re-encryption its key and secure the data transmission of the upper cluster. Also, nodes belonging to the same clusters should re-construct a new key. Therefore, the forward secrecy is guaranteed.

*Theorem:*

In Ad Hoc networks, the security in multicast communications is guaranteed.

*Proof:*

From lemma 2, it is proved that backward secrecy is guaranteed. In lemma 3, we have proved that the forward secrecy is guaranteed. Therefore, the security in multicast communications is guaranteed.

## V. SIMULATION RESULTS

From the equations (2) and (3), we can calculate the total number of KEK messages sent during the clustering for one cluster (n=1):

$$M_{total} = M_{join} + M_{leav}$$

$$\Rightarrow \quad M_{total} = 3\mathrm{NC}_i + \mathrm{NC}_j + \mu - 3$$

⇨ The total number of KEK messages sent for n clusters is:

$$M_{total} = \sum_{1}^{n-1} 3\mathrm{NC}_i + \mathrm{NC}_j + \mu - 3 \qquad (5)$$

The aim of simulations we have performed is to study the impact of transmission range of nodes r and density ρ on cluster formation and KEK messages overhead. We are also interested in studying the impact of the number of clusters with respect to the number of nodes on the number of KEK messages sent in ad hoc network.

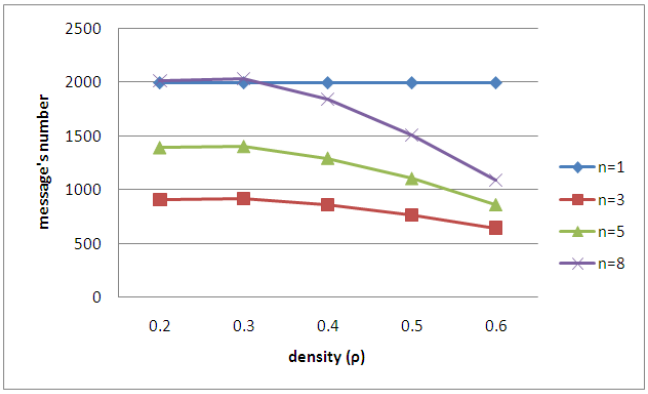



Figure 7. Number of KEK messages sent with and without clustering for N = 200 nodes and r = 30 m

The first simulation is performed (figure 7) with 95 nodes for a transmission range r = 22 meters and the network density ρ is varied from 0.2 to 0.6 (the number of nodes per unit area). In the second simulation (figure 8), we increased the number of nodes to N = 200 for a transmission range r = 30 meters with the same variation of density (ρ = 0.2, 0.3, 0.4, 0.5, 0.6). The two simulations are evaluated for a different number of clusters (n = 3, 5.8) and compared with the case where there is one cluster (n = 1) or we can say that there is no clustering in the network. The two figures (7, 8) show that when the number of clusters increases, the number of KEK messages will be decreased because each node joins or leaves the network affects a single cluster. The case (1 affect N) has been avoided because each node in this case affects only K nodes (where k is the number of nodes in a cluster).

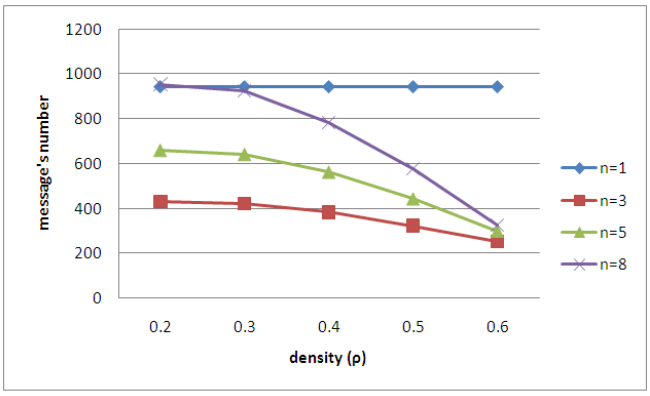



Figure 8. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 22 m

The simulation results show also the benefits of using clusters for the management and maintenance of keys. Increasing the density allows us to have clusters with very high cardinality, which reduces the number of KEK messages in ad hoc network and ensures efficient key management.

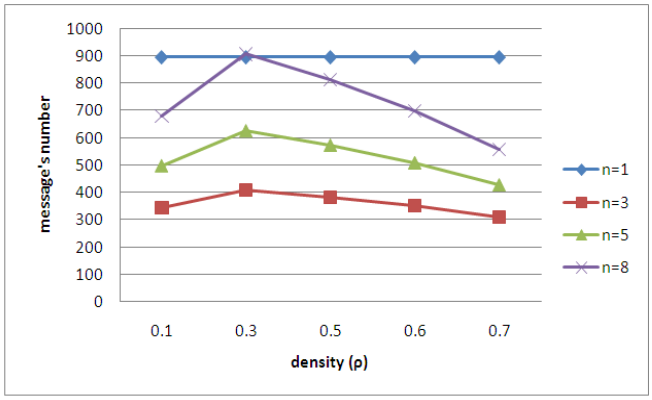



Figure 9. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 18 m

In this part of simulation, the aim is to study the impact of transmission range of nodes r and density ρ on cluster formation and KEK messages overhead. The number of nodes is fixed at 95 nodes with a transmission range varied (r = 18, 20.22). We observed for each simulation the change in the number of KEK messages sent during cluster formation for several values of density ρ.

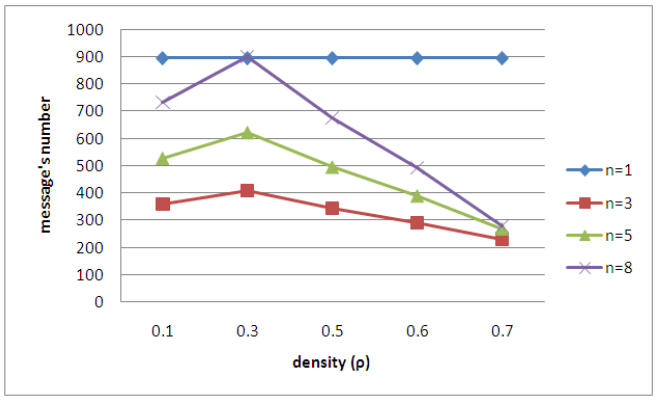



Figure 10. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 20 m

Figures 9, 10, 11 show that when the transmission range of node r increases, the number of KEK messages decreases. Similarly, getting clusters with a huge amount of nodes and wide coverage will increase the probability of staying these node within the cluster, and this will lead up to decrease the number of KEK messages and increase the stability of the cluster.

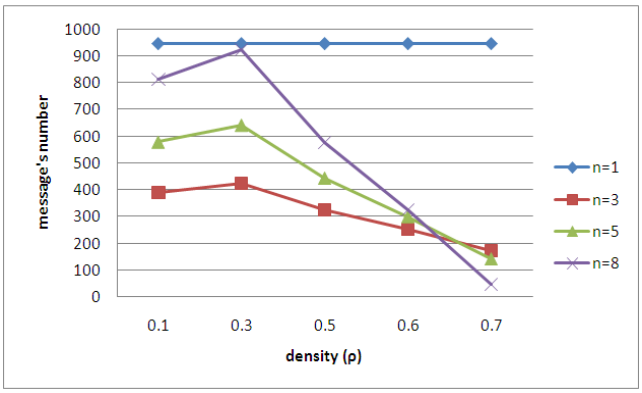



Figure 11. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 22 m

In this part of simulation, we fixed transmission range of nodes r = 20 m and we increase the number of nodes with the same change of the number of clusters (n = 1, 3, 5, 8). The aim is to check the impact of the number of nodes with the number of clusters on KEK messages overhead.

In the figures 12, 13, the simulations show that when we used 95 nodes and 100 nodes with a different number of clusters (n = 3, 5, 8), the number of KEK messages decreased, but when we used 250 nodes with the same number of clusters, the number of KEK messages increased over a number of KEK messages sent in the network without clusters (n = 1) as shown in figure 14. So the number of clusters must be compatible with the number of nodes in the ad hoc network.

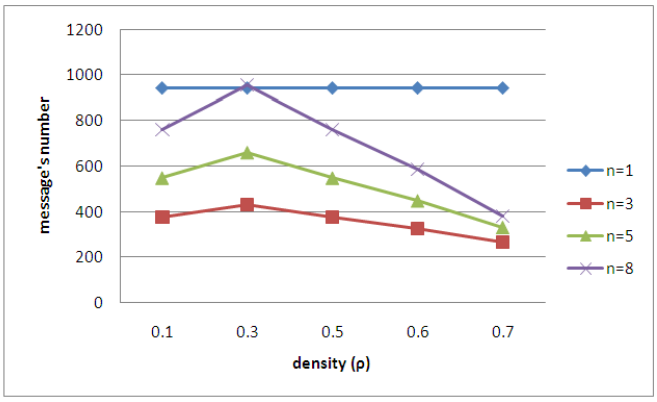



Figure 12. Number of KEK messages sent with and without clustering for N = 95 nodes and r = 20 m

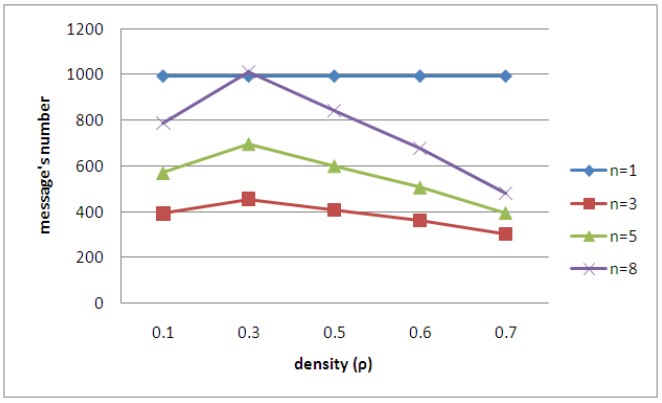



Figure 13. Number of KEK messages sent with and without clustering for N = 100 nodes and r = 20 m

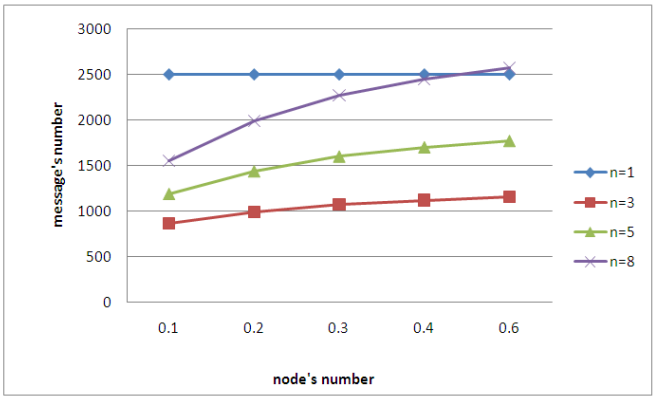



Figure 14. Number of KEK messages sent with and without clustering for N = 250 nodes and r = 20 m

## VI. CONCLUSION

In this paper, we have presented an analysis of communication overhead for a security protocol based on dynamic clustering in ad hoc networks. The main idea of this protocol is based on the affinity relationships between the nodes for cluster formation, when a node is chosen as clusterhead, it generates two *KEKs*. The first key is shared by clusterhead and its local members and the second key is shared by the clusterhead and its parent cluster. The proposed protocol is scalable for large and dynamic multicast groups. For evaluate the performance of the proposed protocol, we have calculated the number of KEK messages sent during protocol steps and we have performed several simulations for analysis of communication overhead, we have studied the impact of transmission range of nodes r, density ρ and the number of clusters with the number of nodes on KEK messages overhead in ad hoc network. The work presented provides a good basis for further analysis on the performance of clustering protocols for MANET networks.

In future work, the communication overhead analysis will be investigated and compared with different clustering based protocols.

## VII. REFERENCES

[1] J. Gaber, "Spontaneous Emergence Model for Pervasive Environments," Proc. IEEE GLOBECOM Workshop. Washington, November 2007.

[2] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," Proc. 7th International Workshop on Security Protocols, 1999.

[3] F. Stajano, "The Resurrecting Duckling: What Next?," Proc. 8th International Workshop on Security Protocols, B. Crispo, M. Roe, and B. Criso, Eds., Lecture Notes in Computer Science, Vol. 2133, Berlin: Springer-Verlag, April 2000.

[4] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," Proc. IEEE/ACM Transactions on Networking, vol. 8, 2000, pp.16–30.

[5] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," ACM Transactions on Information and System Security, vol. 7, 2004, pp.60–96.

[6] M. Haddad and H. Kheddouci, "A survey on graph based service discovery approaches for ad hoc networks," Proc. IEEE International Conference on Pervasive Services. Istanbul, 2007.

[7] M. S. Bouassida, I. Chrisment, and O. Festor, "Validation de BALADE. INRIA research rapport N 5896, April 2006.

[8] M. Bakhouya and J. Gaber, "An affinity-driven clustering approach for service discovery and composition for pervasive computing," Proc. IEEE International Conference on Pervasive Services, Lyon, France, 2006.

[9] N. Kettaf, A. Abouaissa, P. Lorenz, and H. Guyennet. "A self organizing algorithm for ad hoc networks, " In Proceedings of the 10th IFIP Int. Conf. on Personal Wireless Communication, Colmar, France, August 2005.

[10] Y. M. Tseng, C. C. Yang, and D. R. Liao, "A Secure Group Communication Protocol for Ad Hoc Wireless Networks," Advances in Wireless Ad Hoc and Sensor Networks and Mobile Computing, Book Series "Network Theory and Applications," Springer 2006.

[11] K. C. Chan and S. H. Chan, "Key Management Approaches to Offer Data Confidentiality for Secure Multicast," Proc. IEEE Journal on Network, pp. 30-39, October 2003.

[12] C. C. Chang and C. Y. Chung, "An Efficient Session Key Generation Protocol," Proc. IEEE International Conference on Communication Technology, Beijing, China, pp. 203-207, April 2003.

[13] I. R. Chen, J. H. Cho, and D. C. Wang, "Performance Characteristics of Region-Based Group Key Management in Mobile Ad Hoc Networks," Proc. IEEE International Conference on Sensor Networks, Vol. 1, pp.411-419, June 2006.

[14] J. Pieprzyk and C. H. Li, "Multiparty Key Agreement Protocols," IEE Journal on Computers and Digital Techniques, pp.229-236, July 2000.

[15] X. Mingqiang, E. R. Inn-Inn and K. G. S. Winston, "Analysis of Clustering and Routing Overhead for Clustered Mobile Ad Hoc Networks", (ICDCS'06) 26th IEEE International Conference on Distributed Computing Systems, pp.46, 2006.

[16] M.Bechler, H. J. Hof, D. Kraft, F. Pahlke and L. Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks," Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, Hongkong, March 2004.

[17] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," in Proc. 7th IEEE Symp. on Comp. and Communications (ISCC), Taormina, 2002.

[18] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in Proc. 9th International Conference on Network Protocols (ICNP). Riverside, California: IEEE, Nov. 2001, pp. 251–260.

[19] C. Maghmoumi, T. A. Andriatrimoson, J. Gaber, and P. Lorenz, "A Service Based Clustering Approach for Pervasive Computing in Ad Hoc Networks," in Proc. IEEE GLOBECOM 2008, December 2008.

[20] G. Venkataraman, S. Emmanuel and T. Srikanthan, "Size Restricted Cluster formation and Cluster Maintenance Technique for Mobile Ad-hoc Networks", International Journal of Network Management, Wiley InterScience, 2007, Vol.17, pp. 171-194.

[21] N. S. Yadav, B.P. Deosarkar and R.P.Yadav, "A Low Control Overhead Cluster Maintenance Scheme for Mobile Ad hoc NETworks (MANETs)," ACEEE International Journal on Network Security, Volume 1. Number 1. May 2009.

[22] C. Maghmoumi, H. Abouaissa, J. Gaber, and P. Lorenz, "A Clustering-Based Scalable Key Management Protocol for Ad Hoc Networks," in Proc. Second International Conference on Communication Theory, Reliability, and Quality of Service, France, July 2009.


www.iariajournals.org

**International Journal On Advances in Intelligent Systems**
ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2679

**International Journal On Advances in Internet Technology**
ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
issn: 1942-2652

**International Journal On Advances in Life Sciences**
eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
issn: 1942-2660

**International Journal On Advances in Networks and Services**
ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
issn: 1942-2644

**International Journal On Advances in Security**
ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS
issn: 1942-2636

**International Journal On Advances in Software**
ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2628

**International Journal On Advances in Systems and Measurements**
ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
issn: 1942-261x

**International Journal On Advances in Telecommunications**
AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
issn: 1942-2601