Word of Welcome

Dear participant,

We are happy to receive you at the 3rd Welcome the Wireless World (W3) workshop held on 27 Sep 2011. We hope that you will enjoy the programme as well as the personal contact with other researchers. This edition of the workshop offers you a diverse programme with topics spanning from sensor networks to vehicular applications with different abstraction level on technical content. The keynote and invited talks will enrich your experience and the panel discussion will ensure a live and interesting atmosphere to argue about the potential and future of wireless technologies in the society.

We wish you a pleasant and valuable experience!

With regards,

Desislava Dimitrova　　Koen Blom　　Nirvana Meratnia

Organizers

Computer Networks and Distributed Systems group, University of Bern
Pervasive Systems group, University of Twente
Computer Architecture for Embedded Systems group, University of Twente

Local Committee

Desislava Dimitrova, University of Twente
Koen Blom, University of Twente
Nirvana Meratnia, University of Twente

Technical Program Committee

Marco Bekooij, NXP
Hans van den Berg, TNO ICT
Arta Dilo, Pervasive Systems, University of Twente
Sonia Heemstra de Groot, Wireless & Mobile Communications, TU Delft
Paul Havinga, Pervasive Systems, University of Twente
Geert Heijenk, Design and Analysis of Communication Systems, University of Twente
Val Jones, Biomedical Signals and Systems, University of Twente
Massimiliano de Leoni, Architecture of Information Systems, TU/e
Remco Litjens, TNO ICT
Johan Lukkien, System Architecture and Networks, TU/e
Homayoun Nikookar, International Research Centre for Telecommunications and Radar, TU Delft
Florian Simatos, Probability and Stochastic Networks, CWI
Kees Slump, Signals and Systems, University of Twente
Gerard Smit, Computer Architecture for Embedded Systems, University of Twente
Anna Sperotto, Design and Analysis of Communication Systems, University of Twente
Haibin Zhang, TNO ICT

Sponsor

CTIT Wireless and Sensor Systems (WiSe)

CTIT
Centre for Telematics and
Information Technology

Keynote Speaker

prof. Torsten Braun (University of Bern)
Experimental Research on Reliability and Energy-Efficiency in Wireless Sensor Networks

Invited Speaker

dr. Maria Lijding (Smart Signs solutions B.V.)
Smarter Signs for Smarter Buildings

Contents

Full papers

- Synchronization of OFDM at low SNR over an AWGN channel

  A. Kokkeler and G. Smit

- Dynamic Master selection in wireless networks

  M. de Graaf

- A Proposal for Modelling Piggybacking on Beacons in VANETs

  W. Klein Wolterink, G. Heijenk and G. Karagiannis

- Efficient sharing of dynamic WSNs

  D. Bijwaard and P. Havinga

- Acoustic Scoring and Locating System for Rockets, Artillery & Mortars

  L. Stano, J. Wind, H. de Bree

- Analysis of Mobile WSNs over IP

  D. Bijwaard, P. Havinga and H. Eertink

Poster abstracts

- Exploring Patterns of Activities of Daily Living in the Home Environment

  T. Tönis, H. op den Akker, S. Boerema, F. van Polen and H. Hermens

Synchronization of OFDM at low SNR over an AWGN channel

André B.J. Kokkeler, Gerard J.M. Smit

University of Twente
Department of Electrical Engineering,
Computer Science and Mathematics,
P.O. Box 217,
7500 AE Enschede,
The Netherlands

Abstract. This paper is based on Extended Symbol OFDM (ES-OFDM) where symbols are extended in time. This way ES-OFDM can operate at low SNR. Each doubling of the symbol length improves the SNR performance by 3 dB in case of a coherent receiver. One of the basic questions is how to synchronize to signals far below the noise floor. An algorithm is presented which is based on the transmission of pilot symbols. At the receiver, the received signal is cross correlated with the known pilot symbol and the maximum magnitude is determined. The position of the maximum value within the cross correlation function indicates the time difference between transmitter and receiver. The performance of the algorithm in case of an Additive White Gaussian Noise (AWGN) channel, is assessed based on a theoretical approximation of the probability of correct detection of the time difference. The theoretical approximation matches with simulation results and shows that synchronization can be achieved for low (negative) SNRs.

Keywords: Correlation, Differential phase shift keying, Fourier transforms, Frequency division multiplexing, Modulation

1 Introduction

Orthogonal Frequency Division Multiplexing (OFDM) is the most popular multi carrier transmission scheme for already quite some years. It is being used in e.g. IEEE802.11a and 3GPPLTE [1]. In OFDM systems, data is spread over a large number of orthogonal carriers, each being modulated at a low bit rate. The modulation scheme for the carriers can be selected among e.g. multilevel-QAM, QPSK or BPSK, dependent on the channel conditions and the noise level at the receiver. Given a modulation scheme, a transmitter has to use a minimum amount of transmit energy to achieve acceptable Bit Error Rates (BERs) [2]. In case the power budget at the transmitter is constant, increasing noise levels at the receiver are generally counteracted by lowering the modulation level. Once arrived at the lowest modulation level (BPSK), other techniques have to be used to combat worsening noise conditions. In [3] and [4], repetition of symbols in time is analyzed. By means of Maximum Ratio Combining, multiple replicas of symbols are used to lower the BER, also lowering the bit rate. In [5], repetition of data in the frequency domain is elaborated.

However, repetition of data is not commonly adopted to provide acceptable BERs in low SNR scenarios. One of the reasons is that the options mentioned above result in more complexity at the transmitter and/or receiver. The most practical option available for changing data quality from problematic to acceptable is to use error-control coding [2].

In this paper we propose a computationally efficient OFDM technique we will refer to as coherent Extended Symbol OFDM (ES-OFDM). First, coherent ES-OFDM is presented in section 2. Coherent ES-OFDM is able to achieve acceptable BERs at SNRs far below the noise floor. The question that rises then is how to synchronize to signals deeply buried in noise. In section 3, a synchronization algorithm is presented which can accurately estimate the time difference between transmitter and receiver at low SNR levels. In section 4, the performance of this algorithm is analyzed in case of an Additive White Gaussion Noise (AWGN) channel.

2 Coherent ES-OFDM

2.1 Model of coherent Modulation

Coherent ES-OFDM is based on the assumption that the receiver is exactly synchronized in time, frequency and phase. In Figure 1, the relevant parts of a base-band equivalent model of a coherent ES-OFDM based transmitter-receiver pair are presented.

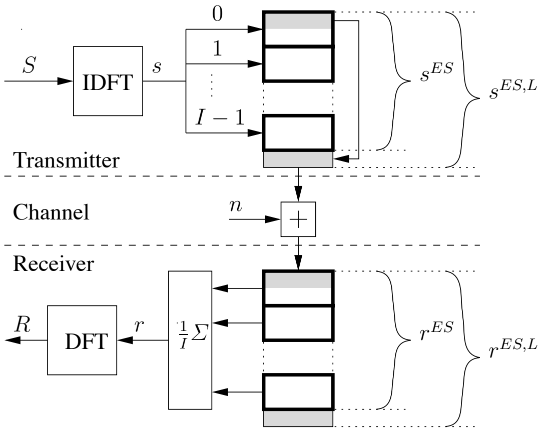


Fig. 1. Base-band equivalent model of coherent ES-OFDM

At the transmitter, a modulator produces S which consists of N complex values (indicated as S_f, $f = 0, 1, ..., N-1$), where each value is a constellation point from a chosen modulation scheme. In this paper we restrict ourselves to BPSK. S is transformed into the time domain through the IDFT giving s.

$$s_t = \frac{1}{\sqrt{N}} \sum_{f=0}^{N-1} S_f e^{j\frac{2\pi f t}{N}}, \quad t = 0, 1, ..., N-1 \tag{1}$$

I copies of s are concatenated giving s^{ES}.

$$s_t^{ES} = s_{\mathrm{MOD}(t,N)}, \quad t = 0, 1, ..., IN-1 \tag{2}$$

where $\mathrm{MOD}(,\mathrm{N})$ indicates the modulo N operator. The last L samples of s ($L \leq N$) act as a cyclic prefix completing $s^{ES,L}$. The values of this extended symbol are shifted out serially and transmitted through the channel. Note that the word 'symbol' is used for representations in both the time and frequency domain. We assume an additive white Gaussian noise (AWGN) channel adding n to $s^{ES,L}$. In the receiver, the first step is to remove the cyclic prefix. The resulting extended symbol is r^{ES}.

$$r_t^{ES} = s_t^{ES} + n_t, \quad t = 0, 1, ..., IN-1 \tag{3}$$

The symbol r^{ES} consists of I blocks of N samples where each block consists of a replica of s and noise. The next step is to average these I blocks.

$$\begin{aligned} r_t &= \frac{1}{I} \sum_{i=0}^{I-1} r_{t+iN}^{ES}, \quad t = 0, 1, ..., N-1 \\ &= s_t + \frac{1}{I} \sum_{i=0}^{I-1} n_{t+iN} \end{aligned} \tag{4}$$

After averaging, the signal is transferred to the frequency domain by the DFT.

$$R_f = S_f + \frac{1}{I} \sum_{i=0}^{I-1} N_{f,i} \tag{5}$$

where

$$N_{f,i} = \mathrm{DFT}(n_{t+iN}), \quad i = 0, 1, ..., I-1 \tag{6}$$

In the next section we will analyze the BER performance when extending symbols.

Extending a symbol with a factor I implies that, at both the receiver and transmitter, the rate at which $\mathrm{(I)DFTs}$ are executed is reduced with the same factor I. At the receiver this reduction is slightly counteracted with a summation operation before the DFT. Note that extended symbols can also be generated by increasing the IDFT size with a factor I and only loading each Ith carrier with information. However, this is computationally inefficient compared to extending symbols as described above.

2.2 Bit Error Rates for coherent ES-OFDM

Extending the symbol at the transmitter and averaging at the receiver effectively does not affect the signal part s but averages the noise (expression 5). In an AWGN channel, the effect of averaging is that the noise power contribution is reduced with a factor I, see [6]. Hence, the SNR is increased with a factor I. Each doubling of the symbol extension factor I improves the SNR performance by approximately 3 dB which makes coherent ES-OFDM being able to operate at low SNR. Using the expression for the BER in case of BPSK in an AWGN channel (see [2]) results in

$$\mathrm{BER} = \frac{1}{2}\mathrm{erfc}(\sqrt{\frac{I}{M}\mathrm{SNR}}) \tag{7}$$

where $M = 1$ for BPSK.

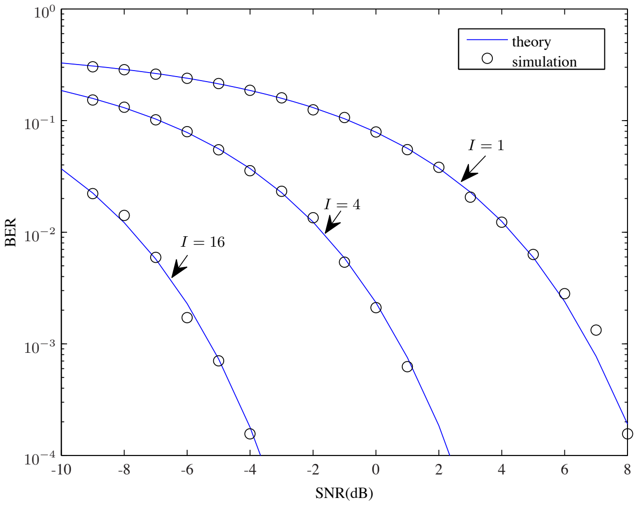


Fig. 2. BERs for coherent ES-OFDM for an AWGN channel.

In Figure 2, the theoretical and simulation results are presented for extension factors I = 1, 4 and 16. The simulation results are in correspondence with theory that each doubling of the factor I improves the BER performance with 3 dB (quadrupling leads to 6 dB improvement). We also see that synchronization has to be obtained at low SNR levels. For example to achieve 10^{-3} BER for $I = 16$, synchronization should be possible at an SNR of approximately -6 dB. Synchronization methods that use the correlation between the cyclic prefix and the 'tail' of the symbol (see [7], [8]) do not deliver the accuracy required; for negative SNR, the error is larger than one sample period. For that reason, we introduce a synchronization algorithm that can cope with negative SNR.

3 Synchronization

The synchronization of coherent ES-OFDM is based on the transmission of pilot symbols, known to both the transmitter and receiver. A prerequisite of the pilot symbol is that its autocorrelation function is the delta function in case of a critically sampled OFDM system. At the transmitter, the pilot symbol is defined as p_t for $t = 0, 1, ..., IN+L$. At the receiver, the pilot symbol is defined as $p_{\mathrm{MOD}(t,IN+L)}$ for $t \in \mathbb{Z}$. We assume that phase and frequency synchronization have been obtained and only time differences remain which are an integral number of sample intervals. The timing difference between transmitter and receiver then equals θ which is an integer number. The received signal is then defined as

$$r_{t,\theta}^{ES,L} = p_{(t+\theta)} + n_t, \quad t = 0, 1, ..., IN + L \tag{8}$$

In case of a critically sampled OFDM receiver and an AWGN channel, p_t and n_t can be considered as realizations of independent stochastic variables P and N, where samples are mutually independent (stochastic variables are indicated with non-italic capitals, realizations with corresponding lower case characters). Consequently, $r_{t,\theta}^{ES,L}$ is a realization of stochastic variable R = P+N. We define σ_{R}, σ_{P} and σ_{N} as the standard deviations of R, P and N respectively. The SNR of the received signal R then equals (see [9])

$$\mathrm{SNR} = \frac{\sigma_{\mathrm{P}}^2}{\sigma_{\mathrm{N}}^2} \tag{9}$$

Since P is an OFDM symbol, we approximate its probability density function by a normal distribution and therefore P and R have a bi-variate normal distribution. The correlation coefficient of this distribution equals

$$\rho = \sqrt{\frac{\mathrm{SNR}}{\mathrm{SNR} + 1}} \tag{10}$$

The correlation function $z_{\tau,\theta}$ of $r_{t,\theta}^{ES,L}$ and p_t is defined as

$$z_{\tau,\theta} = \frac{1}{\sigma_{\mathrm{P}}\sigma_{\mathrm{R}} \cdot (IN + L)} \sum_{t=0}^{IN+L-1} r_{t+\theta}^{ES,L} \cdot p_{t+\tau}^{*} \tag{11}$$

where * indicates the complex conjugate. $z_{\tau,\theta}$ is a set of realizations of stochastic variables $\mathrm{Z}_{\tau,\theta}$ for $\tau = 0, 1, ..., IN + L - 1$.

Basically we are interested in the magnitudes of the complex values $z_{\tau,\theta}$. Note that the factor in front of the summation in expression 11 need not be calculated since we are only interested in the maximum of $|z_{\tau,\theta}|$. The maximum value of $|z_{\tau,\theta}|$ is obtained for $\tau = \theta$. Thus, the position of the maximum value of the magnitudes of the correlation function indicates the time difference between transmitter and receiver. We therefore formulate the following estimator

$$\tilde{\theta} = \arg\max_{\tau}\{|z_{\tau,\theta}|\} \tag{12}$$

4 Performance analysis

To asses the performance of the algorithm, we define $\sigma_{|Z_{\tau,\theta}|}$ as the standard deviation and $\mu_{|Z_{\tau,\theta}|}$ as the expected value of the magnitude of $Z_{\tau,\theta}$. We observe that $\mu_{|Z_{\tau,\theta}|} = \mu_{|Z_{\tau-\theta,0}|}$ and $\sigma_{|Z_{\tau,\theta}|} = \sigma_{|Z_{\tau-\theta,0}|}$. So an analysis of the situation where $\theta = 0$ suffices to indicate the performance for any time difference. For that reason, we will omit the subscript θ in the remainder of this section. The expected value of $|Z_\tau|$, $\mu_{|Z_\tau|}$ will thus have a maximum at $\tau = 0$. Because of the critically sampled OFDM system and the AWGN channel, $\mu_{|Z_\tau|}$ will mostly be zero except for a few values of τ. In case $I = 2$, this is illustrated in Figure 3.

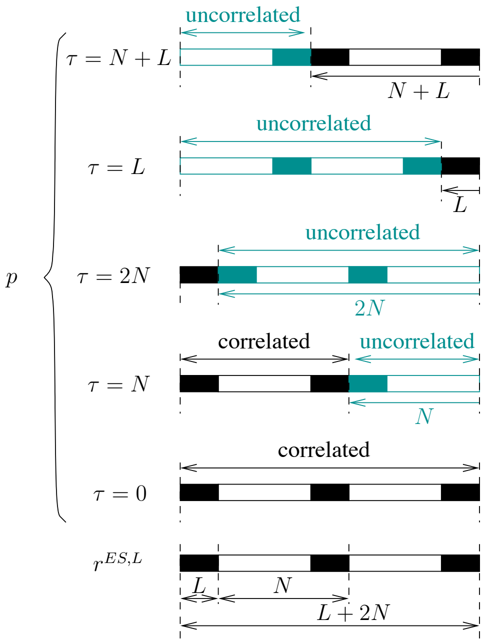


Fig. 3. Example of the construction of a correlation function

In this figure, the signal related part of the received signal $r^{ES,L}$ is shown at the bottom where the cyclic prefix of length L is presented at the left, followed by two symbols. Note that the tails of both symbols are equal to the cyclic prefix. Each sample of $r^{ES,L}$ is multiplied with a sample of p as described in expression 11. In Figure 3, $p_{t+\tau}$ is schematically drawn for those values of τ for which $\mu_{|z_\tau|} \neq 0$. In case $\tau = 0$, all samples of $r_\tau^{ES,L}$ are partly correlated with the corresponding samples of $p_{t+\tau}$. For $\tau = N$, the structure for $\tau = 0$ is cyclically shifted N positions to the left. Consequently, the first $L + N$ samples are still correlated with $r^{ES,L}$ (at the bottom of Figure 3) but the last N samples are uncorrelated. For $\tau = 2N$ the last $2N$ samples are uncorrelated. In case $\tau = L$, the first $2N$ samples are uncorrelated leading to the same expected value of $|z_\tau|$ as for $\tau = 2N$. The correlation peak for $\tau = N + L$ equals the peak for $\tau = N$. In general, $\mu_{|\mathrm{Z}_\tau|}$ has a maximum value for $\tau = 0$ and has smaller peak values for $\tau = iN, \;\; i = 1, 2, ..., I$ and $\tau = jN + L, \;\; j = 0, 1, ..., N - 1.$

Because of the addition of noise and because the summation in expression 11 runs over a finite length, a realization of $|\mathrm{Z}_\tau|$ might have its maximum for other values than $\tau = 0$. This is indicated as an erroneous detection. To evaluate the performance of the synchronization algorithm, we determine the probability of erroneous detection ($\mathrm{P}_{\ \mathrm{E}}$). For convenience, we first determine the probability that the peak is detected correctly (P_D). The peak is detected correctly if $\forall\tau \neq 0, \;\; |z_\tau| < |z_0|$, thus

$$\mathrm{P}_\mathrm{D} = \prod_{\tau=1}^{IN+L-1} \mathrm{P}(|z_\tau| < |z_0|) \tag{13}$$

To determine P_D, we have to determine the probability distribution of $|\mathrm{Z}_\tau|$. We start with the definition of four partial sums x_τ, y_τ, $x\mathrm{o}_\tau$ and $y\mathrm{o}_\tau$. After that, the probability distribution will be determined.

As suggested in Figure 3, the summation in expression 11 is split into two parts; a summation of products of $r^{ES,L}$ and p where there is correlation and a summation of products where there is no correlation. For $\tau = 1, 2, ..., IN + L - 1$, we therefore define x_τ, the first part of the summation, and y_τ, the second part of the summation.

$$\begin{aligned} x_\tau &= \frac{1}{\sigma_\mathrm{P}\sigma_\mathrm{R} \cdot (IN + L - \tau)} \sum_{t=0}^{IN+L-\tau-1} r_t^{ES,L} \cdot p_{t+\tau}^* \\ y_\tau &= \frac{1}{\sigma_\mathrm{P}\sigma_\mathrm{R} \cdot \tau} \sum_{t=IN+L-\tau}^{IN+L-1} r_t^{ES,L} \cdot p_{t+\tau}^* \end{aligned} \tag{14}$$

For $\tau = iN, \;\; i = 1, 2, ..., I$, x_τ is the correlated part and y_τ is the uncorrelated part. For $\tau = jN + L, \;\; j = 0, 1, ..., I - 1$, it is the other way around.

We also split the summation in expression 11 into two parts for the specific case where $r^{ES,L}$ and p are aligned in time

$$\begin{aligned} x\mathrm{o}_\tau &= \frac{1}{\sigma_\mathrm{P}\sigma_\mathrm{R} \cdot (IN + L - \tau)} \sum_{t=0}^{IN+L-\tau-1} r_t^{ES,L} \cdot p_t^* \\ y\mathrm{o}_\tau &= \frac{1}{\sigma_\mathrm{P}\sigma_\mathrm{R} \cdot \tau} \sum_{t=IN+L-\tau}^{IN+L-1} r_t^{ES,L} \cdot p_t^* \end{aligned} \tag{15}$$

In both $x\mathrm{o}_\tau$ and $y\mathrm{o}_\tau$, p and $r^{ES,L}$ are correlated.

To determine P_D (expression 13), the probality distribution of $|\mathrm{Z}_\tau|$ has to be determined for each τ. We distinguish 3 disjunct sets of values for τ. For τ-set 1, $\tau = iN$, $i = 1, 2, ..., I$. For τ-set 2, $\tau = jN + L$, $j = 0, 1, ..., I - 1$ and τ-set 3 consists of all other values of τ. We will analyse the probability distributions of $|\mathrm{Z}_\tau|$ for the three τ-sets separately, followed by an overall analysis.

4.1 Analysis of τ-set 1

For $\tau = iN, \;\; i = 1, 2, ..., I, x_{iN}$ equals $x\mathrm{o}_{iN}$ because the summations over the correlated parts lead to identical results. The difference between z_{iN} and z_0 is caused by the summation over the uncorrelated part; y_{iN} and $y\mathrm{o}_{iN}$. Relying on the Central Limit Theorem, $|y_{iN}|$ and $|y\mathrm{o}_{iN}|$ can be considered as realizations of normally distributed Gaussian processes (see [10]): $\mathrm{U}_{iN} \sim \mathcal{N}(\mu_{|Y_{iN}|}, \sigma^2_{|Y_{iN}|})$ and $\mathrm{C}_{iN} \sim \mathcal{N}(\mu_{|Y\mathrm{o}_{iN}|}, \sigma^2_{|Y\mathrm{o}_{iN}|})$ respectively. A detection is correct if $|y_{iN}| < |y\mathrm{o}_{iN}|$. The probability of correct detection is then $\mathrm{P}(\mathrm{U}_{iN} < \mathrm{C}_{iN})$ or $\mathrm{P}(\mathrm{T}_{iN} < 0)$, $\mathrm{T}_{iN} = \mathrm{U}_{iN} - \mathrm{C}_{iN}$. T_{iN} has a $\mathcal{N}(\mu_{\mathrm{T}_{iN}}, \sigma^2_{\mathrm{T}_{iN}})$ distribution, where

$$\mu_{\mathrm{T}_{iN}} = \mu_{|Y_{iN}|} - \mu_{|Y\mathrm{o}_{iN}|} = 0 - \rho = -\rho \tag{16}$$

$$\begin{aligned} \sigma^2_{\mathrm{T}_{iN}} &= \sigma^2_{|Y_{iN}|} + \sigma^2_{|Y\mathrm{o}_{iN}|} \\ &= \frac{1}{2iN} + \frac{1+\rho^2}{2iN} = \frac{\rho^2+2}{2iN} \end{aligned} \tag{17}$$

The probability of correct detection for $\tau = iN, \;\; i = 1, 2, ..., I$, then becomes

$$\mathrm{P}(|z_\tau| < |z_0|) = \mathrm{P}(\mathrm{T}_{iN} < 0) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{-\mu_{\mathrm{T}_{iN}}}{\sigma_{\mathrm{T}_{iN}}\sqrt{2}}\right)\right) \tag{18}$$

where erf is the error function.

4.2 Analysis of τ-set 2

For $\tau = jN + L, \;\; j = 0, 1, ..., I-1$, $\mathrm{P}(|z_{jN+L}| < |z_0|) = \mathrm{P}(|z_{(I-j)N}| < |z_0|)$. So, the contributions to expression 13 for $\tau = jN + L, \;\; j = 0, 1, I-1$ are equal to the contributions for $\tau = iN, \;\; i = 1, 2, ..., I$.

4.3 Analysis of τ-set 3

For all other values of τ, the probability distributions of $|\mathrm{Z}_\tau|$ are equal to a Rayleigh distribution and an estimate of the probability of detection is based on [10]

$$\mathrm{P}(|z_\tau| < |z_0|) = 1 - e^{-\rho^2(IN+L)} \tag{19}$$

4.4 Overall analysis

The overall probability of detection as given in expression 13, is then approximated by

$$\begin{aligned} \mathrm{P_D} = &\left(\prod_{i=1}^{I}\left(\frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{-\mu_{\mathrm{T}_{iN}}}{\sigma_{\mathrm{T}_{iN}}\sqrt{2}}\right)\right)\right)^2\right) \\ &\cdot\left(1 - e^{-\rho^2(IN+L)}\right)^{(N-2)I} \end{aligned} \tag{20}$$

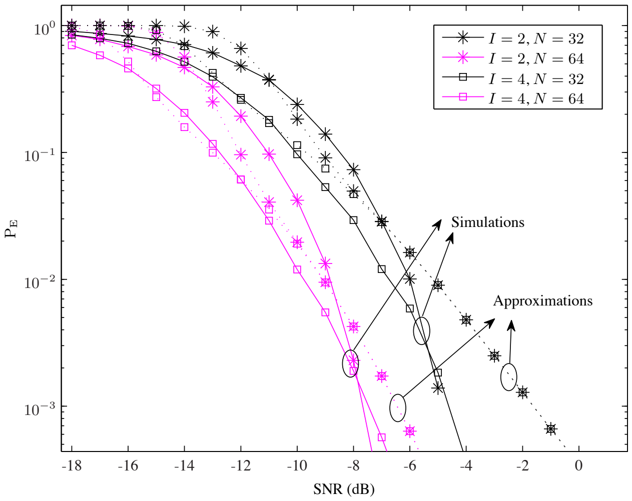


Fig. 4. BERs for coherent ES-OFDM for an AWGN channel.

The probability of error ($\mathrm{P_E}$ = 1-$\mathrm{P_D}$) is given in Figure 4 for $I = 2$ and $I = 4$. For both cases, $\mathrm{P_E}$ is given for $N = 32$ and $N = 64$. The approximations are given by dashed lines, whereas simulation results are given by solid lines.

As can be seen from Figure 4, the simulation results match reasonably well with the approximations. We explain the differences between simulations and approximations by realizing that we assume that extended OFDM symbols and the values of the correlation function z_τ have a Gaussian probability distribution but in practice they have not. Especially for a small number of carriers, this assumption is violated. This is confirmed by the observation that the approximations for N = 64 give a better match with the simulations than the approximations for N = 32. Furthermore, if we concentrate on situations where $\mathrm{P_E} < 10^{-3}$, increasing the number of carriers has more effect than increasing the symbol extension factor I. For these low $\mathrm{P_E}$ values, the second part within expression 20 has limited influence. We therefore concentrate on the first part. The influence of I and N is effectuated through $\sigma_{T_{iN}}$. Increasing N will increase each element of the product in expression 20 whereas increasing I will only add one element to the product, resulting in smaller increase of $\mathrm{P_D}$.

We conclude that the proposed synchronization algorithm can cope with low SNR scenarios. However, for a fixed number of carriers, the symbol extension factor I cannot be increased infinitely since synchronization performance does not scale with I.

5 Conclusion

By extending symbols, OFDM can be used to achieve acceptable BERs at low SNR. In case of coherent ES-OFDM, the SNR can be lowered by 3 dB each time the symbol length is doubled (and inherently, the data rate is halved). Acceptable BERs can be achieved far below the noise floor.

In this paper, an algorithm is presented which estimates the time difference between transmitter and receiver under the assumption that phase and frequency synchronization have been obtained. It makes use of (extended) pilot symbols and can achieve accurate estimates at negative SNRs. Both theoretical approximations as well as simulation results are presented. For example, for an extension factor 4 ($I = 4$), the probability that the time difference is not correctly estimated is less than 10^{-3} in case of 64 carriers and SNR = -6 dB.

The algorithm has been analyzed for an AWGN channel. Future work will be to use more realistic channel models. Furthermore, implementation aspects of the algorithm will have to be investigated.

References

1. A. Bahai, B. Saltzber, and M. Ergen, *Multi-carrier Digital Communication: Theory and Applications*, 2nd ed. Springer, 2004.
2. S. Haykin, *Communication Systems*, 4th ed. John Wiley & Sons, Inc., 2001.
3. N. Maeda, H. Atarashi, S. Abeta, and M. Sawahashi, "Throughput comparison between vsf-ofcdm and ofdm considering effect of sectorization in forward link broadband packet wireless access," *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, vol. 1, pp. 47–51 vol.1, 2002.
4. B. Gaffney, A. Fagan, and S. Rickard, "Upper bound on the probability of error for repetition mb-ofdm in the rayleigh fading channel," *Ultra-Wideband, 2005. ICU 2005. 2005 IEEE International Conference on*, pp. 4 pp.–, Sept. 2005.
5. L. Medina and H. Kobayashi, "Proposal of ofdm system with data repetition," *Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd*, vol. 1, pp. 352–357 vol.1, 2000.
6. W. L. Davenport, W. B. Root, *An Introduction to the Theory of Randam Signals and Noise*. John Wiley & Sons, 1987.
7. J. van de Beek, M. Sandell, and P. Borjesson, "Ml estimation of time and frequency offset in ofdm systems," *Signal Processing, IEEE Transactions on*, vol. 45, no. 7, pp. 1800 –1805, jul 1997.
8. J.-J. van de Beek, P. Borjesson, M.-L. Boucheret, D. Landstrom, J. Arenas, P. Odling, C. Ostberg, M. Wahlqvist, and S. Wilson, "A time and frequency synchronization scheme for multiuser ofdm," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 11, pp. 1900–1914, Nov 1999.
9. J. Stuart, A. & Ord, *Kendall's advanced theory of statistics*, 5th ed. Charles Griffin & company, 1987, vol. 1.
10. K. Milne, "Theoretical performance of a complex cross-correlator with gaussian signals," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 1, pp. 81 –88, feb 1993.

Dynamic Master selection in wireless networks

Maurits de Graaf

Thales Nederland B.V.
Bestevaer 46, 1271 ZA Huizen, Netherlands,
Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
maurits.degraaf@nl.thalesgroup.com

Abstract. Mobile wireless networks need to maximize their network lifetime (defined as the time until the first node runs out of energy). In the broadcast network lifetime problem, all nodes are sending broadcast traffic, and one asks for an assignment of transmit powers to nodes, and for sets of relay nodes so that the network lifetime is maximized. The selection of a dynamic relay set consisting of a single node (the 'master'), can be regarded as a special case, providing lower bounds to the optimal lifetime in the general setting. This paper provides a first analysis of a 'dynamic master selection' algorithm.

1 Introduction

Mobile wireless networks are often battery powered which makes it important to maximize the network lifetime. Here, the network lifetime is defined as the time until the first node runs out of energy. The broadcast network lifetime problem asks for settings of transmit powers and (node-dependent) sets of relay nodes, that maximize the network lifetime, while all nodes originate broadcast traffic. Literature in this area considers the lifetime maximization in mobile ad-hoc networks (MANETs). Often, the complexity is reduced by assuming transmissions originate from a single source ([3], [5] and [7]). The related problem of minimizing the total energy consumption for broadcast traffic has also been widely studied, because it provides a crude upper bound to the lifetime of the network. In [4] and [1] it is shown that minimizing the total transmit power is NP-hard. Another way to reduce the complexity is to allow transmissions from multiple sources but ask for a node independent *set* of relay nodes to maximize the network lifetime. This leads to lower bounds for the general network lifetime problem. This paper presents a first analysis of a special case, where we ask for a *single* relay node (the master), which is allowed to change over time.

2 General model and notation

We assume all nodes can reach each other when transmitting at maximum power. For a set $V \subseteq \mathbb{R}^d$ of potential master nodes, a power assignment is a function $p : V \to \mathbb{R}$. To each ordered pair (u, v) of transceivers we assign a transmit power threshold, denoted by $c(u, v)$, with the following meaning: a signal transmitted by transceiver u can be received by v only when the transmit power is at least $c(u, v)$. We assume that $c(u, v)$ are known, and that these are symmetric. For a node $m \in V$, let p_m denote the power assignment $p_m : V \to \mathbb{R}$ defined as:

$$p_m(v) = \begin{cases} c(v, m) & \text{for } v \neq m, \\ \max_{v \in V} c(v, m) & \text{for } v = m. \end{cases} \tag{1}$$

Each vertex is equipped with battery supply b_v, which is reduced by amount $\lambda p_m(v)$ for each message transmission by v with transmit power $p_m(v)$. Similarly, b_v is reduced by amount $\mu r(v)$ for each reception. Let $T_1, T_2, T_3, \ldots$ denote the time periods. Let node i transmit $a_i(T_j)$ times during time period T_j. We assume that the $a_i(T)$ are constant for all T_i, $(i = 1, \ldots,)$, and define $a_i = a_i(T)$. We call a series of transmissions were each node i transmits a_i times a *round*. Suppose node m is master. With these assumptions, we obtain after one round:

$$b_v = \begin{cases} b_m - \lambda p_m(m) \sum_{v \in V} a_v - \mu r(m) \sum_{v \neq m} a_v & \text{for } v = m, \\ b_v - \lambda a_v p_m(v) - \mu r(v) \sum_{v \in V} a_v & \text{for } v \neq m. \end{cases}$$

In [2] we analyzed the case where a master m is chosen which is kept for the whole lifetime of the network. This paper is concerned with the following problem: given a graph $G = (V, E, c, b, a)$, $c : E \to \mathbb{R}$ denotes the transmit power thresholds, and $b : V \to \mathbb{R}$ denotes the initial battery levels $b_v, v \in V$, and the relative frequencies $a_1, \ldots, a_n$, one asks for times $x_v \geq 0$ for each node v to be master in such a way that $L(G, x) = \sum_{v \in V} x_v$ is maximized under the condition that the remaining battery capacity of each node is positive during the lifetime of the network. In this paper, we assume $\lambda = 1$ (by scaling), $V \subseteq \mathbb{R}^d$, E corresponds to a complete graph, $c(u, v) = \|u - v\|^2$. We also assume $\mu = 0$, which is consistent with many long-range radio systems, where transmit power dominates the signal processing power.[1] We call $x = (x_1, \ldots, x_n) \in \mathbb{N}_+^n$ *feasible* if for all $m \in V$,

$$b_m - \lambda \sum_{v \neq m} a_v x_v p_v(m) - \lambda x_m p_m(m) \sum_{v \in V} a_v \geq 0. \tag{2}$$

[1] The analysis presented above is straightforwardly extendable to the case $\mu \neq 0$.

The terms $\lambda \sum_{v \neq m} a_v p_v(m)$ and $\lambda x_m p_m(m) \sum_{v \in V} a_v$ in (2) indicate the reduction in battery capacity of node m during the periods when nodes $v \neq m$ are master, and when m is master, respectively.

Now (2) can be rephrased as: $Ax \leq b$, where $b : V \rightarrow \mathbb{R}^+$, and where A is an $n \times n$-matrix where the entry corresponding to (v, m) is defined by:

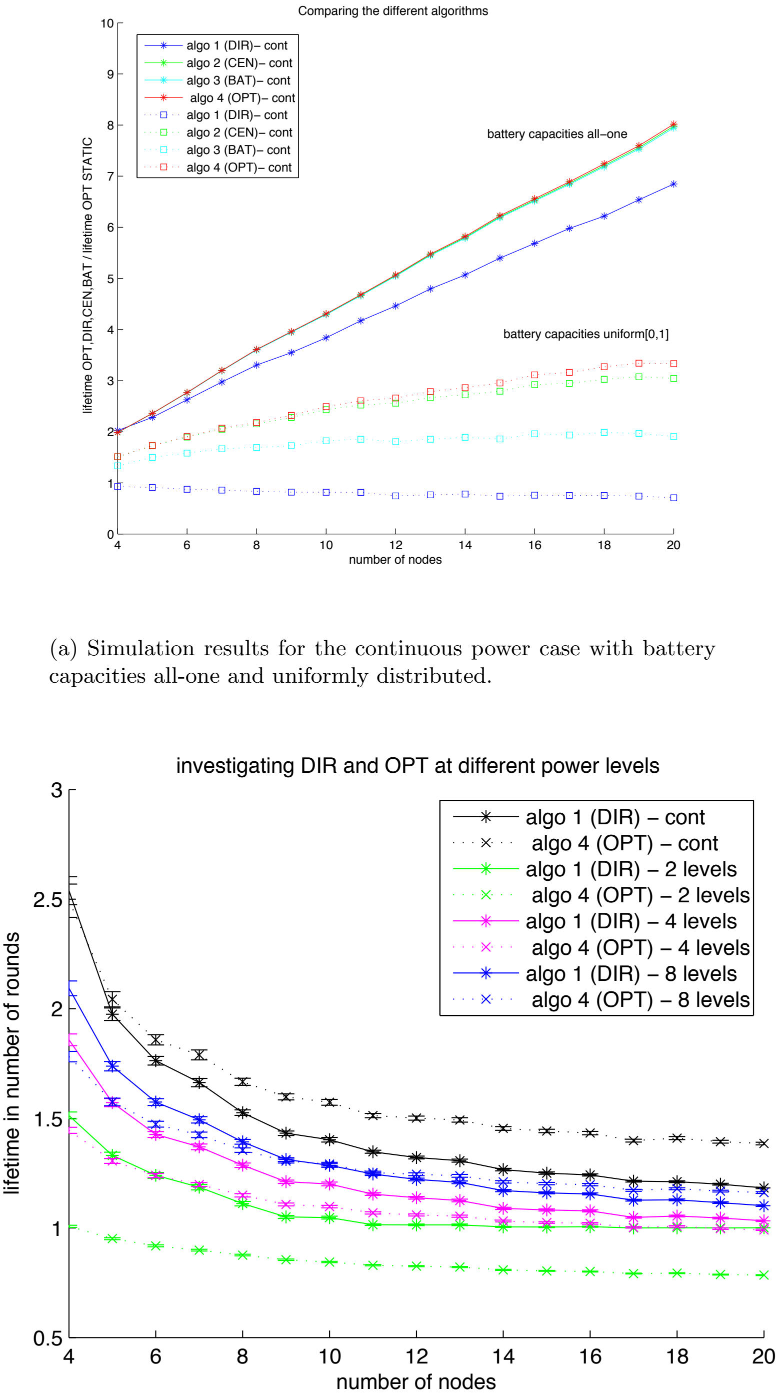
$$A(v,m) = \begin{cases} p_m(m) \sum_{v \in V} a_v & \text{for } v = m, \\ a_v p_v(m) & \text{for } v \neq m. \end{cases} \tag{3}$$

In Section 3 of this paper we compare dynamic master selection algorithms for the continuous power case. In Section 4 we address the impact of supporting only a discrete set of transmit power levels. Section 5 presents the conclusions.

3 The continuous power case

The network lifetime in number of rounds was evaluated for n, ranging from 4 to 20. The nodes were uniformly distributed in a two dimensional disk of unit diameter. For each algorithm, the average network lifetime was evaluated over 1000 simulations. The relative message transmission frequencies were $a_v = 1$ for $v \in V$. The following algorithms were compared: *Optimal Master Selection (OPT).* Choose $x \geq 0$, so that $L(G, x)$ is maximized, under condition (2). *Central Master Selection (CEN).* Choose x, by periodically selecting performing the optimal static master node selection, according to [2]. *Maximum Battery Master Selection (BAT).* Choose x by periodically selecting a master node in such a way that (at the update time t) b_m is maximal among b_v for $v \in V$. *Direct Transmission (DIR).* There is no master: all nodes reach all other nodes via a single hop transmission. We include it for reference purposes.

In Figure 1(a), we compare the ratio of lifetime for the algorithm to the lifetime of the optimal static algorithm (as in [2]). Two cases are displayed: all-one battery capacities: $b_v = 1$ for all $v \in V$, and $b_v \cong U(0, 1), v \in V$. The simulations show that dynamic master selection extends the lifetime significantly compared to static master selection. In order of decreasing lifetime the algorithms are : OPT, CEN, BAT and DIR. OPT and CEN are close, and we expect that CEN and OPT are equal when considering infinitesimal time periods. The improvement depends strongly on the initial battery capacities: for uniformly [0,1] battery capacities this factor is about 3 (for 15 nodes or more), for the all-one battery capacities -where the total amount of energy in the network is, on average, doubled- this



(a) Simulation results for the continuous power case with battery capacities all-one and uniformly distributed.

(b) Comparing DIR and OPT for continuous and 2 and 8 discrete power case with all-one battery capacities.

Fig. 1. Simulation results for dynamic master selection.

factor amounts to at least 6. In this case OPT,CEN and BAT are very close. For the case of uniform [0,1] battery capacities even *static* master selection is better for the network lifetime than direct routing (shown by the blue squared dotted line dropping below one for increasing number of nodes). As the dynamic master selection is a highly specific case of ad-hoc multihop routing, this indicates that introducing multihop routing functionality is beneficial for the network lifetime, provided the transmit power levels are continuously adjustable. Work is in progress to support these simulation results with mathematical analysis.

4 Restricting the number of power levels

In practice, often only a discrete set of transmit power levels is supported in hardware and software. In the extreme case only one constant power level is supported. In contrast to the previous section it is immediately clear that in the constant power case DIR outperforms multihop routing, due to the fact that multihop routing reduces the battery by a constant at each transmission for (at least) 2 nodes. In Figure 1(b) we investigate how many power levels need to be supported before OPT outperforms DIR. Simulations with $U[0,1]$-distributed battery capacities (not displayed) show OPT outperforms DIR already for 2 power levels. However, the figure shows that, with all-one battery capacities, 2 power levels is not enough. For 8 power levels OPT outperforms DIR for 10 nodes or more. However, with 4 or less power levels, DIR outperforms OPT.

As a special case of the fixed number of power levels, we address the constant power case. Here, the matrix A as defined in (3) equals $A = (n-1)pI_n + pE_n$, where I_n denotes the identity matrix and E_n the all-one matrix. Clearly direct transmission leads to a lifetime, in rounds $L = \min\{b_i/p\}$. For the OPT we obtain:

Theorem 1 *Let $G = (V, c, b)$ be given, and $n \geq 2$. Then the network lifetime for algorithm OPT is*

$$L(G) = \min_{v \in V}\{b_v, \frac{\sum_{v \in V} b_v}{p(2n-1)}\} \tag{4}$$

Proof. W.l.o.g. $V = \{1, \ldots, n\}$, $p = 1$ and $b_1 \leq \ldots \leq b_n$. By LP duality $\max\{1^T x | Ax \leq b, x \geq 0\} = \min\{y^T b, yA \geq 1, y \geq 0\}$, where y^T denotes the transpose of a vector, and 1 denotes the all-one vector. Considering $y = (2n-1)^{-1}1^T$, it follows that $\sum x_i \leq (2n-1)^{-1} \sum_{v \in V} b_v$. To see the other upper bound, consider $y = [1, 0, \ldots, 0]$, which implies that $nx_1 +$

$\sum_{i=2}^{n} x_i \leq b_1$, whence also $\sum_{v \in V} x_v \leq b_1$. To see that the upper bounds are attainable, first assume $b_1 \geq \sum_{i=1}^{n} b_i/(2n-1)$. Next consider x as given by $x_i = (b_i - \frac{\sum b_i}{2n-1})/(n-1)$. By assumption x is feasible. Moreover: $\sum x_i = \sum b_i/(2n-1)$ by simple substitution. To see that the lower bound b_1 is attainable, assume ((2)) does not hold, so $b_1 < \sum_{i=1}^{n} b_i/(2n-1)$. Choose $x_1 = 0$, and repeat this procedure until we are back in the situation under (a). With the corresponding assignment also the lifetime b_1 is realized.

5 Conclusions and future work

When the transmit power can be regarded as a *continuous variable*, we find that dynamic master selection algorithms extend the network lifetime significantly compared to static master selection. In order of decreasing lifetime the algorithms are : OPT, CEN, BAT and DIR. The improvement depends strongly on the initial battery capacities. Work is in progress to support these simulation results with mathematical analysis as in [2]. For *discrete power levels*, dynamic master selection can only improve upon direct routing, when there are at least two power levels. Our results suggest that 8 power levels are sufficient for multihop routing to have longer network lifetime than direct transmission, except for small networks.

References

1. Cagalj, M., Hubaux, J., Enz, C.: Minimum-energy broadcast in all-wireless networks, NP-completeness and distribution issues. In: Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, pp 172–182 (2002)
2. Maurits de Graaf, Jan-Kees van Ommeren: Increasing network lifetime by battery-aware master selection in radio networks, Proceedings of the 3rd ERCIM workshop on eMobility, May 2009, Netherlands, pp. 3-14.
3. Kang, I., Poovendran, R.: Maximizing Network Lifetime of Broadcasting over Wireless Stationary Ad Hoc networks, Mobile Networks and Applications, 10, 879–89, (2005)
4. Liang, W.: Constructing minimum-energy broadcast trees in wireless ad hoc networks, In: Proceedings of the International Symposium on Mobile Ad Hoc Networking and Com-puting (MobiHoc), pp. 112–122 (2002)
5. Pow, C.P., Goh, L.W.: On the construction of energy-efficient maximum residual battery capacity broadcast trees in static ad-hoc wireless networks, Computer Communications, 29, 93–103 (2005)
6. Lloyd, E., Liu, R., Marathe, M., Ramanathan, R., Ravi, S.: Algorithmic Aspects of Topology Control problems for ad-hoc networks, Mobile Networks and applications, 10, Issue 1-2 , 19–34 (2005)
7. Park, J., Sahni, S.: Maximum lifetime broadcasting in wireless networks. In: 3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005:1-8 (2005)

A Proposal for Modelling Piggybacking on Beacons in VANETs

Klein Wolterink, W., Heijenk, G, and Karagiannis, G.

Department of Computer Science, University of Twente, The Netherlands
{w.kleinwolterink, geert.heijenk, g.karagiannis}@utwente.nl

Abstract. Piggybacking on beacons is a forwarding technique that is regularly used in vehicular ad-hoc network (VANET) research as a means to disseminate data. With this technique data is attached to and transmitted along with scheduled beacons, without changing the timing of the beacons.

The performance of piggybacking largely depends on network parameters such as the network density, the beaconing frequency, etc. It is our goal to model the performance of piggybacking as a function of such parameters. In this paper we present our methodology to achieve this goal, and show some first conclusions w.r.t. which network parameters should be taken into account in our model.

Keywords: beaconing, dissemination, piggybacking, VANET

1 Introduction

Vehicular networking can be considered as one of the most important enabling technologies for Intelligent Transportation Systems (ITS). Vehicular networking is concerned with communication between vehicles and infrastructural devices, supporting a multitude of traffic applications.

Traffic applications can typically be categorized either as safety applications or efficiency applications. A typical example of the former is the 'Emergency electronic brake lights' use case [1] in which a vehicle sends out a high-priority warning to all nearby vehicles. It is critical that such a warning is disseminated fast (< 100 ms) to all relevant (i.e., nearby) vehicles – messages will therefore be disseminated with an increased priority. Non-delivery of a message can cause less safe situations. The distances involved are limited and can be covered by at most a few transmission hops. In contrast to this, traffic efficiency messages are typically targeted at a larger geographical region and may have a lifetime of tens of seconds. Non-delivery of a message can cause less efficient behaviour (e.g., increased travel time) but will not cause dangerous situations. A typical example is the 'Decentralized floating car data' use case, see [1].

The issue of disseminating safety messages has so far received a lot of attention, leading to a large number of (mainly) flooding-based solutions. When applied to disseminating efficiency messages however these solutions are far from optimal. For this reason attention has been shifting to more delay-tolerant dissemination strategies for the delivery of efficiency messages. One such strategy is disseminating messages by attaching them to network-level beacons. We refer to this technique as *dissemination by beaconing* or *piggybacking*.

With piggybacking forwarding of packets is only done by attaching them to scheduled beacons. Since the scheduling of beacons is a far from trivial problem [2], it seems preferable that the piggybacking process should not influence the beacon scheduling, thus keeping the timing of the beacons unchanged. Forwarding by piggybacking is therefore relatively slow when for instance compared to a flooding strategy. As was already noted in [3] the speed with which information is disseminated depends amongst others on the beaconing frequency, the network density, and the transmission distance.

A main *expected* advantage of piggybacking is that the impact piggybacking has on the network load should be considerably less when compared to other forwarding strategies:

- Since the packets are attached to already scheduled beacons, network- and security overhead for every transmitted packet can be saved. Together this overhead may be more than 200 bytes [4][5].
- Additionally one access to the wireless medium per packet is avoided, thus reducing contention and the risk of network collisions.

Based on our own experiences in [3] and reported results by others (see the discussion on related work in [10]) our ongoing research focuses on piggybacking.

Consider the following scenario: there exists a source node S and a destination node D, the latter which is located d meters from S. S transmits a packet at $\tau = 0$ which is forwarded by means of piggybacking to D. Our problem statement then becomes: What is the probability that D will have received the packet within τ seconds for a given set of network parameters? Examples of network parameters are the network density, the transmission power, the beaconing frequency, etc.

It is our goal to create a model that is able to predict the probability that destination node D will have received the packet within τ seconds, as a function of the set of network parameters. In this paper we present our methodology to create such a model. We also show some results of a first analysis we have performed on the impact of a number of network parameters on piggybacking in a static network.

The outline of this paper is as follows. In Section 2 we introduce our methodology. In Section 3 we present the results of a simulation study on piggybacking in a static network. We conclude the paper in Section 4.

2 Methodology

Piggybacking may be implemented in a number of ways, and it is impossible to create a single model that is able to capture the behavioural details of every possible implementation. In Section 2.1 we describe how we model the behaviour of a piggyback protocol in a generic manner, and list the assumptions that our model contains. In Section 2.2 we present our methodology.

2.1 Forwarding model

Any dissemination protocol in an intermittently connected VANET must employ two different forwarding strategies, depending on the state of the network [3] [6] [7] [8]:

1. As long as a node is able to forward the data to a node that is closer to the destination the data will actively be forwarded in that direction.
2. When an intermediate forwarder is not able to forward the data to a node that is closer to the destination, then the *store-carry-forward* mechanism is used: the data is locally stored by a node that moves in the direction of the destination. This node will carry the data until it finds a node that is closer to the destination, and will then forward the data to this node, at which point the first strategy is again applied.

The above two-step approach also holds for piggybacking. Thus, the time it takes to piggyback data from the source to the destination depends on:

1. $d_{forwarding}$ – The forwarding delay: the time it takes to actively forward data a certain distance.
2. $d_{carrying}$ – The carrying delay: the time it takes for a node to carry the data to a node that is closer to the destination.

In our model we treat these two delays independently. Work on calculating $d_{carrying}$ can be found in [9]. Our work initially focuses on $d_{forwarding}$; later on we will combine the two delays into a single model.

We now state two assumptions that are meant to ease the modelling of $d_{forwarding}$ for piggybacking, while we argue that it will not diminish the applicability of our model. The first assumption is that both source and destination are situated on the same stretch of road. The second assumption is that this stretch of road is straight.

The first assumption is rather unrealistic of course, but since the node topology in a VANET is by definition limited to the road network every possible route between a source and a destination can be broken down into a limited set of stretches of road. For each of these stretches our assumption holds.

Although the second assumption is incorrect as well, we do not expect it to have a significant impact on the model. In non-urban situations curves in a road are rather gradual, while inside an urban area sharp curves often imply a new stretch of road, or can be modelled as such.

2.2 Methodology

As has already been stated, our work focuses on the forwarding delay. Our methodology to calculate $d_{forwarding}$ consists of the following steps:

1. First we simulate the performance of a piggybacking protocol for a range of network parameters.

2. We then analyse the impact each network parameter has on the performance of the piggybacking protocol.
3. Based on this analysis we model the performance of the piggybacking protocol based on those network parameters that were found to have a significant impact.

To ease the modelling of the impact of the network parameters we initially ignore mobility. Later on we will repeat the three steps with mobility taken into account. We expect the effect of mobility on the forwarding delay to be limited, since the speeds with which vehicles move are typically not significant w.r.t. the transmission ranges and beacon frequencies involved.

Currently we have performed the first two steps for a static network. We discuss the results of these steps in Section 3.

3 Simulation Study of a Static Network

In this section we describe the set-up and discuss the results of a simulation study that investigated the performance of piggybacking in a static network. This study was part of the first two steps of our methodology, see Section 2.2. The goal of this study was to answer the following research questions:

1. Which network parameters should be taken into account to express the probability that a packet has been piggybacked a certain distance within a certain time interval?
2. How significant is the impact of each network parameter on the performance of the piggybacking (specified below), and in what way do the parameters affect performance?
3. Can the effect of some network parameters be combined into a single parameter?

In Section 3.1 we describe the set-up of the experiment, the parameters that have been varied during simulation, and the performance metrics that have been measured. In Section 3.2 we discuss the results.

Due to space limitations the description of the experiment and the discussion of the results are limited and incomplete – for a complete discussion of our simulation study see [10].

3.1 Experimental Set-up

The piggyback protocol that we have used in our experiments is relatively simple, such that we are better able to judge the impact of the network parameters. It is similar to the protocol we have used earlier [3] which despite its simplicity proved to be quite effective. Once we have fully modelled the performance of this protocol as a function of network parameters, we expect that it should take considerably less effort to model more involved protocols.

It is assumed that nodes know their own geographical position. The forwarding rules for every node are as follows. At $\tau = 0$ the source node piggybacks a service data unit (SDU) that has a geographical destination region attached to it. When a node receives a beacon that contains an SDU, it will encapsulate this SDU in its next scheduled beacon if by that time all of the following (still) hold:

1. The node is not the source of the SDU.
2. The node has not received the SDU from another node that is located closer to the destination region.
3. The node has not included the SDU in a previous beacon.

Once the SDU reaches a node that is inside the destination region the forwarding stops.

Different scenarios have been created by varying the following parameters: the distance over which a packet must be piggybacked (the *dissemination distance*), the average distance between nodes (the *inter-node distance*), the transmission power, the transmission bit rate, the beacon frequency, the size of the beacon window (not explained here as it was found to have no significant impact on performance), the size of a beacon, and the size of the SDU.

For each experiment the following performance metrics have been measured:

1. $d_{forwarding}$ – The forwarding delay: the time it takes to forward the SDU to the destination region
2. P_{pr} – The probability that the SDU reaches the destination region. It may be that the SDU is lost during piggybacking because of transmission failures.
3. $P_{pr}(\tau)$ – The measured probability whether the SDU has reached the destination region within τ seconds. I.e., if we have measured that 80% of all SDUs reached the destination within 10 s, we state that $P_{pr}(10) = 0.8$.

3.2 Results Analysis

The three network parameters that have the biggest impact on performance are the dissemination distance, the inter-node distance, the transmission power, the transmission bit rate, and the beacon frequency.

An increase in the dissemination distance has the obvious effect of increasing the network delay. An increase of the dissemination distance gives a linear increase in $d_{forwarding}$ and a linear decrease in P_{pr}. If the dissemination distance and the inter-node distance are changed but their reciprocal ratio remains the same, $P_{pr}(\tau)$ (and thus the other two performance metrics as well) will remain largely unchanged. E.g., if the dissemination distance is doubled and the inter-node distance is halved, you will get the same results.

The reception probability for a single hop is mainly determined by the transmission power and the transmission bit rate used. If different combinations of these two network parameters result in the same single hop packet reception probability, $d_{forwarding}$, P_{pr}, and $P_{pr}(\tau)$ for the multi-hop case are also the same for these combinations. We can therefore combine the network parameters transmission power and transmission bit rate into a single network parameter: the single hop reception probability.

The main effect of increasing the beacon frequency is an exponential decay in $d_{forwarding}$. The main effect of increasing the mean inter-node distance is a linear increase of the network delay and a decrease of the packet reception probability. The main effect of increasing the beacon size is a linear increase in the network delay.

Increasing the transmission power, transmission bit rate, the beacon frequency, and the beacon size will lead to an increased network load, as will a decrease of the inter-beacon distance. As the network load is increased a point will be reached where increasing the load further leads to an increase in the amount of unsuccessful transmissions. An increase of the network load beyond this point will lead to an increase in $d_{forwarding}$ and a decrease of P_{pr}. E.g., as the beacon frequency is increased beyond this point, P_{pr} decreases linearly.

The effect of the size of the SDU on performance is significant but negligible compared to the effect of the network parameters mentioned previously. The size of the beacon window has no impact on performance.

4 Conclusions & Future Work

Piggybacking is a method to disseminate data by attaching it to already scheduled beacons. It is our goal to model the performance of piggybacking as a function of relevant network parameters. Our main performance metric, and the outcome of our model, is the probability that a destination node has received the data within τ seconds.

In this paper we have presented our methodology to create such a model and the specific steps involved. We have also shown how the delay of piggybacking data can be broken down into two parts: the forwarding delay when actively forwarding data from node to node, and the carrying delay when data is carried by a node to nodes that are closer to the destination. Our research focuses on the forwarding delay.

In Section 3 we have discussed some results of a simulation study on piggybacking inside a static network. The following network parameters should at least be taken into account when modelling piggybacking: the dissemination distance, the average distance between nodes, the transmission power, the transmission bit rate, and the beacon frequency. Although the beacon size and the size of the SDU also impact performance significantly, their effect is an order of magnitude less.

Our next steps will be to model the piggyback performance for the static case. We will then simulate, analyse and model the impact of mobility on performance. Finally we will combine our model to calculate $d_{forwarding}$ with (an) existing model(s) to calculate $d_{carrying}$, into a model that is able to calculate the complete delay to piggyback information from a source node to a destination region.

References

1. "Intelligent transport systems (its) – vehicular communications – basic set of applications – definitions," European Telecommunications Standards Institute, Technical Report 102 638, 2009.
2. M. van Eenennaam, W. Klein Wolterink, G. Karagiannis, and G. Heijenk, "Exploring the solution space of beaconing in vanets," in *Vehicular Networking Conference (VNC), 2009 IEEE.* IEEE, pp. 1–8.
3. W. Klein Wolterink, G. Heijenk, and G. Karagiannis, "Dissemination protocols to support cooperative adaptive cruise control (cacc) merging," in *International Conference on ITS Telecommunications (to appear)*, 2011.
4. M. Raya and J. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.
5. C. Project, "D31 european its communication architecture – overall framework – proof of concept implementation," Information Society Technologies, Tech. Rep., 2008.
6. C. Sommer, R. German, and F. Dressler, "Adaptive beaconing for delay-sensitive and congestion-aware traffic information systems," *University of Erlangen, Dept. of Computer Science, Technical Report CS-2010-01*, 2010.
7. L. Wischhof, A. Ebner, and H. Rohling, "Information dissemination in self-organizing intervehicle networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 1, pp. 90–101, 2005.
8. O. Tonguz, N. Wisitpongphan, and F. Bai, "Dv-cast: a distributed vehicular broadcast protocol for vehicular ad hoc networks," *Wireless Communications, IEEE*, vol. 17, no. 2, pp. 47–57, 2010.
9. S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 6, pp. 3341–3356, 2008.
10. W. Klein Wolterink, G. Heijenk, and G. Karagiannis, "Information dissemination in vanets by piggybacking on beacons an analysis of the impact of network parameters," in *IEEE Vehicular Networking Conference (VNC) 2011 – To appear.* IEEE, 2011.

Efficient sharing of dynamic WSNs

Dennis J.A. Bijwaard[2] and Paul J.M. Havinga[1,2]

[1] Pervasive Systems, University of Twente, P.O. Box 217, 7500 AE Enschede
D.Bijwaard@utwente.nl and P.J.M.Havinga@utwente.nl
[2] Ambient Systems, Colosseum 15d, 7521 PV Enschede

Abstract. The Ambient middleware supports real-time monitoring and remote maintenance across the Internet via wired and mobile wireless network access technologies. Additionally, the middleware offers easy integration with third-party applications. Ambient Studio utilizes the middleware for remote WSN configuration and monitoring. The ConnectBox utilizes it to monitor and maintain WSNs remotely. This paper describes the Ambient middleware and compares its efficiency with the existing messaging protocols used for instant messaging and web services.

1 Introduction

The Ambient middleware enables remote monitoring and maintenance of WSNs, and makes it easy to use sensor readings in customer applications. The GPRS-enabled ConnectBox allows deployment of sensor networks in moving vehicles like trucks. This enables real-time monitoring while goods are in transit. When there are temporary connection outages, the ConnectBox buffers the sensor messages and flushes them when the connection is re-established

The Ambient network [2] is self-organizing and consists of two main layers: an infrastructure layer with a Gateway and MicroRouters that relay messages across multiple hops, and a layer of SmartPoints that move through the network and in/out of networks.

This paper describes the Ambient middleware and compares its messaging efficiency.

2 Ambient middleware

The Ambient middleware enables customers to easily integrate their applications and to enable remote monitoring and maintenance. The interaction between the different components is depicted in Figure 1. Note that AmbientStudio and the ConnectBox share the same Ambient middleware (AmbientMW).

One or more Gateways can be connected via RS232 using the AmbientMW in a ConnectBox device or AmbientStudio on a PC. The ConnectBox device is an embedded Linux device that offers Ethernet connectivity towards the wireless nodes from XML applications, AmbientStudio, or other AmbientMW instantiations.

The AmbientMW offers the ConnectAPI to ease integration with third-party applications using asynchronous XML messages over a TCP/IP connection (optionally encrypted with SSL). The XML messages are the same Device Driver Interface (DDI) that are used between the nodes in the WSN, but fully parsed so they can be easily used in an application utilizing its XML schema (which enables code generation in for instance Java and C#). When required, a pass filter can be configured to reduce the type of DDI messages that are forwarded over ConnectAPI.

To offer flexibility, the ConnectAPI can be started as client and server: The server allows multiple local or remote applications to connect. The client allows connecting to a remote host, automatic re-connects, and automatic logging of messages while disconnected and flushing when the connection is re-established.

The AmbientMW also offers AmbiLink to ease remote monitoring and maintenance of sensor networks using asynchronous binary messages over optionally SSL encrypted TCP/IP connections. Similar to the ConnectAPI, both AmbiLink client and server can be started with the AmbientMW. This offers the flexibility to monitor and maintain multiple ConnectBoxes with one or more AmbientStudio instances without loosing messages when client connections are disrupted. Additionally to DDI messages, also management messages can be sent over both ConnectAPI and AmbiLink for configuring, opening and closing, serial ports and remote connections. New message types can easily be added, for instance, for fetching historical data or changing DDI message filters. Another message type could be introduced for file exchanging (for instance firmware) with the WSN, such that the WSN can use its own pace and protocol for exchanging it with the involved node(s). AmbiLink also supports merging sensor information from all connected nodes via multiple ConnectBox or AmbientStudio instances. It can then provide the merged data to multiple applications using the ConnectAPI. In both AmbiLink and ConnectAPI, message destinations can be unicast, multicast, and broadcast using wildcards in the destination of messages.

For both AmbiLink and ConnectAPI, conversion between DDI and respectively their binary and XML counterpart was automated. Logging and flushing is implemented in the middelware for both protocols in order to cache messages that cannot be sent by the client during connection outage. Server logging and flushing is not implemented, since sensor messages are usually towards a server and there is no guarantee that a client will ever reconnect to the server. To reduce message loss, the TCP connections were set up such that small messages were sent without delay and the last sent message is logged until it is possible to sent the next message. This removed the need for a special acknowledgement scheme on top of TCP (which already has its own acknowledgements), since a new message cannot be sent unless the previous one was successfully sent.

3 Messaging efficiency

In this section the efficiency of ConnectAPI and AmbiLink is compared with existing messaging protocols.

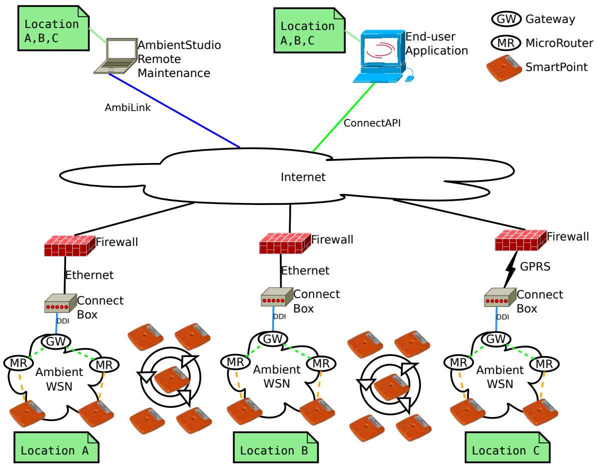


Fig. 1. Ambient connect framework

3.1 Comparing existing methods

Existing methods for messaging over the Internet are the email protocol Simple Message Transfer Protocol [12] (SMTP) for sending/receiving email, and Instant Messaging (IM) protocols like Internet Relay Chat [10] (IRC), Protocol for SYnchronous Conferencing [1] (PSYC), Session Initiation Protocol (SIP)/SIP for IM and Presence Leveraging Extensions [7] (SIMPLE) and Extensible Messaging and Presence Protocol [14] (XMPP). Also web-services like Simple Object Access Protocol [13] (SOAP) and Representational State Transfer [16] (REST), and peer to peer (P2P) messaging like P2P SIP can be used for message exchange over the internet. These messaging protocols can be used over a variety of transport protocols like TCP and UDP, and can use security protocols like Internet Protocol Security [11] (IPsec), Secure Socket Layer [5] (SSL) and Transport Layer Security [4] (TLS). Most of the protocols can also provide nomadicity (i.e. reconnection after connection loss), Mobile IP (MIP) can be used to provide seamless connectivity when switching networks. Unfortunately, MIP is not deployed in current networks and would therefore at least require a home agent and driver software on each involved computer to function.

Criteria for comparing the existing methods:

- Availability: are the required elements widely deployed, or are can they be easily deployed? Availability is positive when the protocol is generally supported in the endpoints and intermediate routers, negative when is is hardly supported on the endpoints and routers. For instance MIP and multicast are not widely deployed, application-level protocols can often be easily deployed.
- Impact: The impact is high when the routers along the path must be equipped to support the protocol (denoted as ”dr” for dedicated router), or when the

firewall must be updated to support incoming traffic (denoted as ”df”). The impact is also high when dedicated clients (denoted as ”dc”) or a dedicated server (denoted as ”ds”) is required. The impact is less when a library can be used for clients (denoted as ”lc”), and servers (denoted as ”ls”). Using for instance XML messages, usually requires a library for parsing it.

- Latency, i.e. are messages forwarded in real-time, or are there inherent delays? For instance request-based mechanisms like web-services require higher bandwidth and processing time and double that with the required return messages.
- Reliability: is message loss prevented, or is there a mechanism to prevent losing messages?
- Reachability: can the Wireless Sensor Networks (WSN) be reached remotely when there is an Internet connection? For instance (company) firewalls often block all incoming ports and are not keen on clear-text protocols, a default Network Address Translation [15] (NAT) router blocks all incoming connections unless configured with specific forwarding rules.
- Bandwidth: can the protocol work across a limited bandwidth link such as General packet radio service (GPRS)? For instance verbose messaging like SOAP could add much overhead and other associated costs across a wireless link such as GPRS.
- Security: can others inject or obtain messages, or disrupt the service? Can the protocol easily be encrypted?

The web-service protocols eXtensible Markup Language (XML)-RPC and its successor SOAP use XML documents for messaging. REST can use both text, XML and other representations (for a request an URL could suffice). These web-services all use the request/response model of HTTP. JSON-RPC uses a compact representation and is one of the few web-service protocols that can also be used bi-directionally over a socket, i.e. it allows requests, responses and notifications to be sent asynchronously in each direction over the same connection. When behind a firewall the other protocols require either opening a firewall port, tunnelling or polling on a reachable server to receive messages (SOAP could also be used over SMTP with associated high latency, but then it would not act as a web-service). Using HTTP Secure (HTTPS) for security increases the latency of the first message, since the connection needs not only to be set up for each request but also the security association. The reliability of web-services is generally ok. Multiple libraries are available for all protocols, however there is no cross-platform C++ library available for JSON-RPC (JsonRpc-Cpp is GPLv3 licensed which requires opening all linked source when releasing). Table 2 compares the popular web-service protocols.

For messaging over the Internet, a great number of protocols exist. Only a limited number of these protocols are suitable for integration in applications (i.e. are an open standard [8]). Most of these protocols are not designed for reliability, but reachability is good for all of them since they all provide one or more ways to traverse through firewalls. The messages in these protocols are quite large because they are text-based, especially SIMPLE and XMPP. Table 3 compares the popular open messaging protocols.

3.2 Comparing Ambient middleware

The AmbiLink users binary DDI and ConnectAPI uses DDI in XML format for messaging, for both messaging is asynchronous, meaning that no response is required like in web services. When an AmbiLink or ConnectAPI client is behind a firewall, it can still reach its related server on the Internet without having to reconfigure the firewall. Both AmbiLink and ConnectAPI can be secured with SSL with the added delay of setting up the security association. The reliability of the Ambient middleware is ok, it logs and flushes messages when the connection is temporarily unavailable. AmbiLink only works as part of the Ambient middleware, ConnectAPI can be used from any program that can send XML documents over a socket. Table 4 compares the Ambient middleware protocols.

Table 5 compares the number of messages and bandwidth for a number of protocols in more detail[3]. Typical HTTP header size is 256 bytes, the size of XML and JSON documents are comparable when XML attributes are used instead of tags (else XML is about 30% larger), a typical size of such a message is 1024 bytes. A typical SOAP envelope adds 172 bytes. Typical AmbiLink binary sensor messages are approximately 250 bytes long, typical ConnectAPI messages are approximately 900 bytes long. ConnectAPI messages make heavy use of XML attributes instead of tags, which make them comparable in size to JSON messages.

The table clearly shows that the asynchronous messaging of JSON-RPC, AmbiLink and ConnectAPI saves the return-trip messaging as well as the HTTP headers. Depending on the setup of server and client, the HTTP keep-alive can keep the TCP connection open for a long time. However, usually the keep-alive timeout is less than a minute, which means more connection setups (and associated higher latency) for low-frequency messaging over HTTP. Note that the typical SOAP messages are around 1500 bytes, so a slight increase would require an additional TCP packet.

3.3 Bandwidth optimizations

The aim is to use the Ambient middleware protocols across low bandwidth links like GPRS, in which the download bandwidth varies between 9 and 52 kbit/s, and upload is usually limited to 18 kbit/s. It is envisaged that also large sites may want to use GPRS to be independent of Ethernet infrastructure which could be owned or managed by another party or simply be unavailable in a storage area. For instance 1000 nodes with 3 sensors (e.g. temperature, humidity and

[3] TCP uses 3-way handshake for setup and teardown, the set-up ACK can already contain part of the message, HTTP1.1 can use keep-alive which reduces the number of required TCP connects, TCP message header is 24 bytes, The latency of messages doubles when there is an explicit response for each message. The table assumes that each TCP message is acknowledged, where it practice the acknowledgement can be for a number of them (depending on the rate of transmission). IP header is 24 bytes

Table 1. AmbiLink versus ConnectAPI features

Protocol	Usage	Transport	Security	Format	Filter	Destination	Merged WSNs
ConnectAPI	3^{rd}-party applications	TCP/IP	SSL option	XML	header fields	Broadcast to all applications	Using multiple clients
AmbiLink	Monitoring & maintenance	TCP/IP	SSL option	binary	per WSN	Routing to AmbiLink instance(s)	At client or server

Table 2. Comparison of web service protocols

Protocol	Availability	Impact	Latency	Reachability	Bandwidth	Security
XML-RPC	+	ls+lc	medium	issues	medium	HTTPS
SOAP	+	ls+lc	medium	issues	high[9]	HTTPS
REST	+	ls+lc	medium	issues	depends	HTTPS
JSON-RPC	+/-	ls+lc	low	two-way	low/medium	SSL/TLS

Table 3. Comparison of open messaging protocols

Protocol	Availability	Impact	Latency	Reliability	Bandwidth	Security
SMTP	+	ds+dc+lc	high	+/-	medium	-
IRC	+	ds+dc+lc	low	+/-	medium	SSL
PSYC	-	ds+dc	low	+/-	medium	TLS/SSL
SIMPLE	+/-	ds+dc+lc	medium	+/-	high	TLS
XMPP	+	ds+dc+lc	medium	+/-	high	TLS

Table 4. Comparison of Ambient middleware protocols

Protocol	Availability	Impact	Latency	Reliability	Bandwidth	Security
AmbiLink	+	ds+dc	low	+	low	SSL
ConnectAPI	+	ds+dc+lc	low	+	medium	SSL

Table 5. Comparison of Bandwidth (in bytes) & latency for N message exchanges, and bandwidth for N=10

	TCP/IP	HTTP	request	response	typical	bandwidth
	headers	headers	messages	messages	message size	N=10 and
Protocol	48 bytes	256 bytes				1 TCP connect
XML-RPC	5+4N..9N	N*2	N*XML	N*XML	XML=1024	27760
SOAP	5+4N..9N	N*2	N*(envelope+XML)	N*(envelope+XML)	envelope=172	36790
REST	5+4N..9N	N*2	N*(URL|XML|other)	N*(XML|OK|other)	URL|OK=100	19000
JSON-RPC	5+2N..5+4N	0	N*JSON	N*(optional JSON)	JSON=900	10200..20160
AmbiLink	5+2N	0	N*AmbiLink	0	AmbiLink=250	3700
ConnectAPI	5+2N	0	N*ConnectAPI	0	ConnectAPI=900	10200

tilt) sending a message every 5 minutes yields an average rate of 10 messages per second[4].

For 10 AmbiLink messages per second[5] that would yield a bandwidth of 2500 bytes/s = 20 kbit/s. So, also AmbiLink could certainly use compression for bigger sensor networks over GPRS. A simple gzip[3] on a binary message gives a compression factor of 1.6 on AmbiLink messages. Compressing a group of messages, e.g. 4 at a time gives compression rate of 4, 25 at a time gives a compression rate of 8. So it would make sense to compress a group messages (e.g. all messages to be send in a second) when possible, this would also reduce the overhead on the TCP/IP level, but will increase the message latency. AmbiLink messages could also be reduced in size by shortening them or using a generic compressing on string values in these messages that are now sent as UTF-8. Huffman coding [6] would be a candidate for this, an alternative would be a look-up table for commonly used attribute names.

Sending 10 ConnectAPI messages per second would require a bandwidth of 70 kbit/s. Compression of these XML messages would thus be required for using ConnectAPI across GPRS with bigger networks. Compression with gzip of a temperature message achieves a compression factor of 1.8. Compressing a group of 4 messages yields a compression factor of 3, compressing a group of 25 messages yields a compression factor of 18. Some more can be saved by stripping redundant information from the ConnectAPI messages and shortening the XML tag and attribute names. A large part of these attribute and tag names come from the DDI descriptors, so shortening them in these descriptors will reduce the bandwidth.

4 Conclusion

The Ambient middleware utilizes the DDI framework that allows any resource in the system to be configured and accessed remotely. This paper compared the efficiency of the middleware messaging with existing methods and describes how it can be further improved.

References

1. Psyc instant messaging. `http://about.psyc.eu/`, Last visited March 2011.
2. Ambient systems. `http://ambient-systems.net`, Last visited August 2011.
3. P. Deutsch. GZIP file format specification version 4.3. RFC 1952, Internet Engineering Task Force, May 1996.

[4] note that these message rates are only required when full sensing history is required, else it is more practical to configure alarms in the SmartPoint on specific sensing conditions

[5] The amount of messaging depends on the number of SmartPoints, its reporting period or alarm thresholds, and scale of the network (current maximum is 64 infrastructure nodes)

4. T. Dierks. The transport layer security (tls) protocol version 1.2. RFC 5246, Internet Engineering Task Force, Aug. 2008.
5. A. O. Freier, P. Karlton, and P. C. Kocher. The ssl protocol version 3.0. `http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt`.
6. D. A. Huffman. A method for the construction of minimum redundancy codes. In *Proc. IRE 40*, pages 1098–1101, 1952.
7. IETF. The simple working group charter. `http://datatracker.ietf.org/wg/simple/charter/`.
8. ITU-T. Open standard. `http://www.itu.int/en/ITU-T/ipr/Pages/open.aspx`.
9. M. B. Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko. Comparison of performance of web services, ws-security, rmi, and rmi-ssl. *Journal of Systems and Software*, 79(5):689 – 700, 2006. Quality Software.
10. W. Kantrowitz. Network questionnaires. RFC 459, Internet Engineering Task Force, Feb. 1973.
11. S. Kent and K. Seo. Security architecture for the internet protocol. RFC 4301, Internet Engineering Task Force, Dec. 2005.
12. J. Klensin. Simple mail transfer protocol. RFC 5321, Internet Engineering Task Force, Oct. 2008.
13. N. Mitra and Y. Lafon. Soap specificiations. `http://www.w3.org/TR/soap/`.
14. P. Saint-Andre. Extensible messaging and presence protocol (XMPP): core. RFC 3920, IETF, Oct. 2004.
15. P. Srisuresh and M. Holdrege. IP network address translator (NAT) terminology and considerations. RFC 2663, Internet Engineering Task Force, Aug. 1999.
16. S. Tilkov. Introduction to rest. `http://www.infoq.com/articles/rest-introduction`.

Acoustic Scoring and Locating System for Rockets, Artillery & Mortars

Łukasz Stano, Jelmer Wind, Hans-Elias de Bree

Microflown AVISA
PO Box 2205, 6802 CE Arnhem, The Netherlands
stano@microflown.com

Abstract. Acoustic sensors can be used to detect, classify and locate battlefield threats such as mortars, rifles and various vehicles. Sound pressure microphones are commonly used for this purpose, but this article focuses on Acoustic Vector Sensors (AVS's). These sensors consist of three orthogonal particle velocity sensors in combination with a sound pressure microphone. These sensors make it possible to measure the direction-of-arrival of a sound wave instantaneously. The use of multiple sensors leads to very robust source localization and classification. This paper presents a system which consists of multiple Unattended Ground Sensors (UGS's). Applications, with an emphasis on Acoustic Scoring and Locating System for Rockets, Artillery & Mortars (RAM-LOC), are discussed.

Keywords: Battlefield acoustics, CRAM, situational awareness

1 Introduction

Conventional sound pressure microphones measure only scalar value of the sound field, such that measurements at a single point do not yield information on the direction of arrival. Hence, spatially distributed microphone arrays must be used. DOA is calculated based on phase differences between the sound pressure at different locations. This technique has some drawbacks: large system size, limited bandwidth and accuracy loss due to wind and temperature changes.

The acoustic vector sensor (AVS) is a 4 channel sensor which consists of an omnidirectional sound pressure microphone and the three orthogonal acoustic particle velocity sensors, each of which is sensitive in one direction. The ratios between these signals are used to determine the direction of the source. This principle makes it possible to determine the direction of a wide range of sources, from helicopters to hand weapons. Since the sensor is only a few millimeters across, it can be mounted on virtually any platform. This means acoustic vector sensors remove two large disadvantages of the existing acoustic system, making it a uniquely versatile technology. Regarding the benefits of AVS the Acoustic Scoring and Locating System for Rockets, Artillery & Mortars is being under development. The main purpose of this project is to increase situational awareness on the battlefield or on the training range.

This article is built up as follows. Section 2 considers the principles of operation of Microflown sensors. Section 3 considers possible applications. In section 4 Unattended Ground Sensor for boarder control and RAM target practice is described. Section 5 considers source localization on experimental data and in section 6 possible improvements are discussed. Conclusions are drawn in section 7.

2 The Microflown Sensor

Invented in 1994, the Microflown sensor is the world's only true acoustic particle velocity sensor designed to operate in air. As can be seen in figure 1 (left), the sensor consists of two wires which are heated to 200°C above the ambient temperature during its operation. As air flows across the sensor, the upstream wire cools down and gives off some heat to the passing air. Hence, the downstream wire cools down less due to the warmer air. This difference in temperature is measured electrically, making it possible to measure the acoustic particle velocity directly. The heating of the wires requires about 70mW.

From 1994 to 2004, the Microflown sensor was a hot topic in the scientific community, leading to hundreds of scientific papers. From around 2004, the sensor has become widely accepted, primarily in the automotive industry. The technology is currently being used to improve the interior sound quality of the products of almost all major car manufacturers.

Microflown Technologies introduced the first true acoustic vector sensor in 2002 (see figure 1, right). It consists of 3 Microflown sensors and a co-located pressure microphone. The three Microflown elements form the acoustic particle velocity vector. Together with the pressure microphone, the direction of the source can be identified.

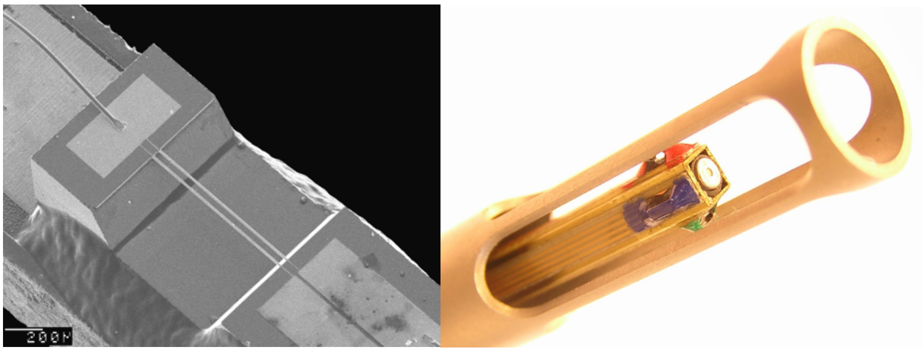


Fig. 1. Left: The Microflown Sensor. Right: An Acoustic Vector Sensor.

3 Applications

This section considers the wide range of defense and security applications of Acoustic Vector Sensors (AVS). Three of the platforms on which the sensors can be placed are discussed: the unattended ground sensor, the unmanned aerial vehicle (UAV) and the ground vehicle. Also, a number of different acoustic signatures are considered.

The main focus of this article lies on the RAM-SCORE system, which is currently under development at Microflown AVISA. This system consists of a network of unattended ground sensors (see figure 2, left) to determine the location of the launches and impacts of Rockets, Artillery and Mortars (RAM). It is used in target practice on military training ranges.

This information is useful for safety officers because it returns up-to-date information about the time and location of shots that are being fired and alert the operator if any rockets or projectiles that land outside the target area. The system also makes it possible give accurate feedback to the practicing infantry or artillery troops. A similar product, called RAM-LOC, is a network of unattended ground sensors for wide area surveillance in counterinsurgency warfare such as in Afghanistan.

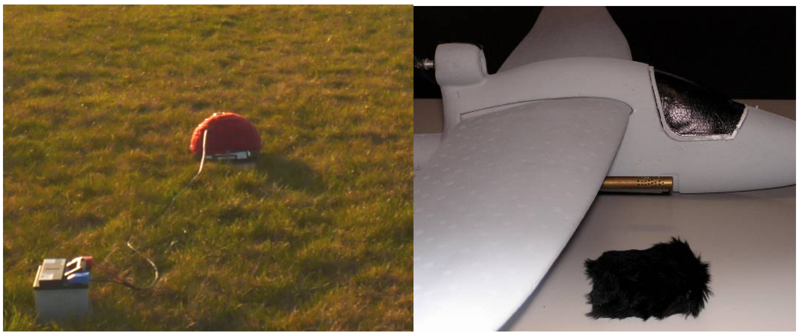


Fig. 2. Left: Unattended Ground Sensor. Right: Microflown UAV Demonstrator.

The second platform is the miniature unmanned aerial vehicle (mini UAV) (see figure 2, right). These remote controlled aircrafts, with a wingspan of 1-1.5m are widely used for video surveillance activities by the police and the armed forces. The camera has a narrow field of view, which means the operator is unaware of most things happening around the UAV. The acoustic vector sensor makes it possible to detect and locate any weapon noise, from pistols to mortar fire.

The sensor can also be placed on the ground vehicle platform. Inside an armored vehicle such as the Fennek reconnaissance vehicle (see figure 3), the operators are essentially deaf to external noise. Acoustic Vector Sensors make it possible to alert the camera operator to any threat and, if the operator needs it, rotate the camera towards it. A similar system is being considered for the air target artillery, where threat is not a rifle or mortar, but a helicopter. Although this requires different signal processing, most of the hardware is the same in both applications.

This section considered a number of applications of acoustic vector sensors for defense and security. Three different platforms and various acoustic signatures have been considered. From this point on, this article focuses on the RAM-SCORE system.

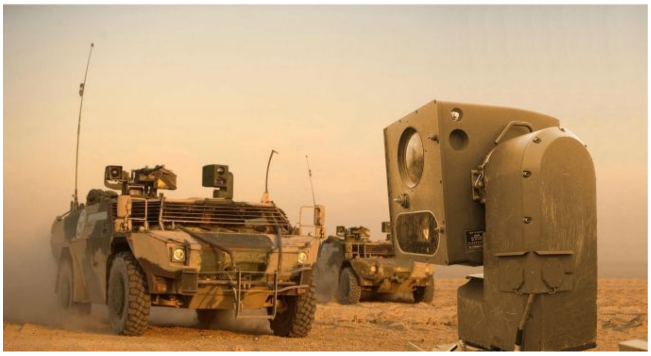


Fig. 3. Ground-vehicle-based sensor (technology demonstrator).

4 The Unattended Ground Sensor

The Unattended Ground Sensor (UGS) (see figure 6) is designed to be a self-powered device which records measurement data, classifies events and sends the time and direction of the event to a main station. A network of sensors is used to determine the location of the event based on time differences and directions.

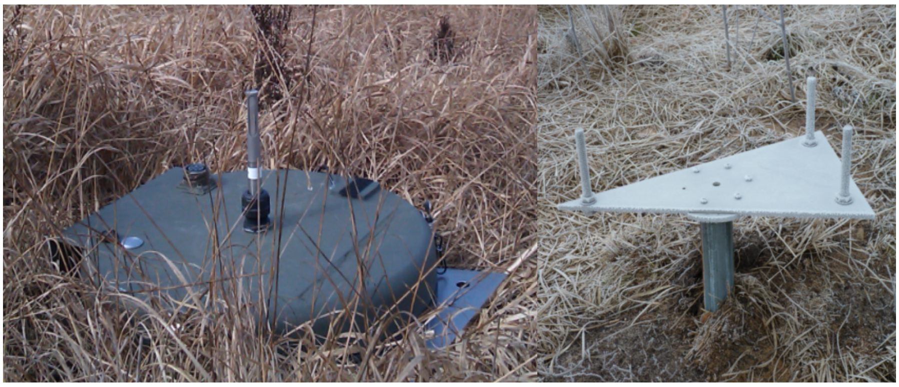


Fig. 4. Left: Unattended Ground Sensor without its windshield. Right: Sensor post.

Each UGS has a diameter of 0.33m (13'') and a height of 0.25m (10''). It contains a computer running Windows, a pre-amplifier, an AD converter and a GPS unit. A 12V/62Ah battery is used to power the system. The 686MHz band is used to transfer data at distances of 10km and more.

To determine the location of the events, the location and orientation of the sensors must be known accurately. For this purpose, the sensors are placed on posts (see figure 4, right). The location of these posts has been measured up to 1m and the orientation is known up to ½ degree.

5 Source Localization

The location of the launches and impacts is determined at the main station. The time and direction of the events is received from all of the sensors and brought together to identify a source location. The details of the source localization algorithm are not discussed because of company confidentiality. As becomes clear in this section, both time differences and directions are used.

Experimental results are collected during military practice shootout at the ASK shooting range (Artillerie SchietKamp). 81mm mortars are launched from known locations and the sensors are placed at 2, 2.5 and 5km from it. The localization results are depicted in figure 5 (left). The colored lines depict the direction of the sound wave, as measured by the sensor. Each blue curve indicates the set of possible launch locations, based on the time difference of one pair of sensors. The intersection of two or more of these curves is the launch position, determined from the time differences. The accuracy can be improved by combining the time differences and the directions. By zooming in to the picture (see figure 5, right), it can be seen that the error is less than 50 meters.

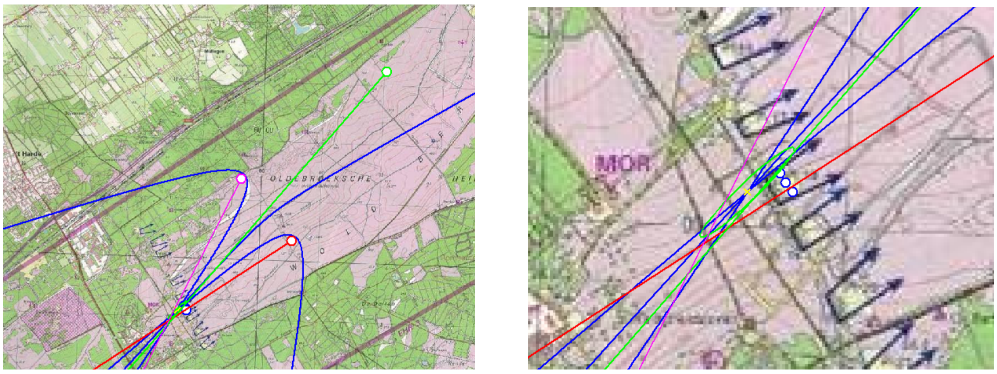


Fig. 5. Computing a launch position at the ASK (3 sensors).

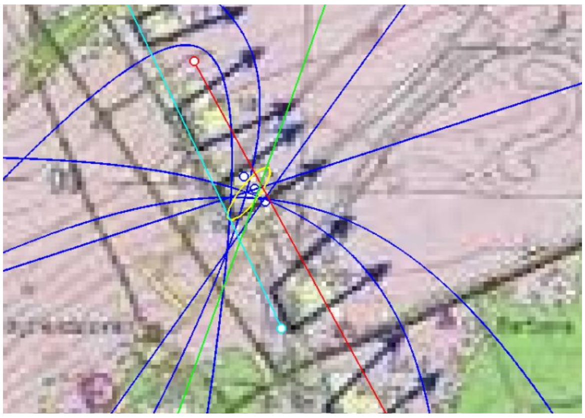


Fig. 6. Computing a launch position at the ASK (4 sensors).

Better results are achieved if 4 sensors are used instead of 3, and if the sensors are placed around the launch position. Figure 6 depicts the computed source position using 4 sensors at 500m,500m, 2 km and 2.5km from the launch position. The error is within 5 meters.

6 Future Developments

The main current effort lies in the development of embedded signal processing hardware to reduce the systems size and battery usage while at the same time, making it more robust, lighter and smaller. The embedded system will be ruggedized to be suitable for wide area surveillance in counterinsurgency warfare: the RAM-LOC system.

Another important development lies in the classification and localization of non-impulsive acoustic sources such as ground vehicles and aircraft. The Acoustic Vector Sensor (AVS) has equally large advantages for these types of sources: it can determine a direction of the source for each frequency bin of a Discrete Fourier Transform (DFT). This makes it possible to draw a time-frequency-angle plot. Here, the horizontal and vertical axes represent time and frequency respectively, the brightness indicates the sound level and the color indicates the direction.

Figure 7 (left) is the time-frequency-angle of a passing rotary wing aircraft. It can be seen that in the lower frequency range, the signals are contaminated by noise from the east. Only the fundamental frequency of the main rotor at 100Hz shows that the aircraft is actually in the west. Above 200Hz, the noise is absent and the source is consistently to the east. Figure 7 (right) depicts the direction of the source based on this higher frequency range, as well as a curve which corresponds to a source flying in a line at a constant velocity. It can be seen that this approximation is accurate up to a few degrees.

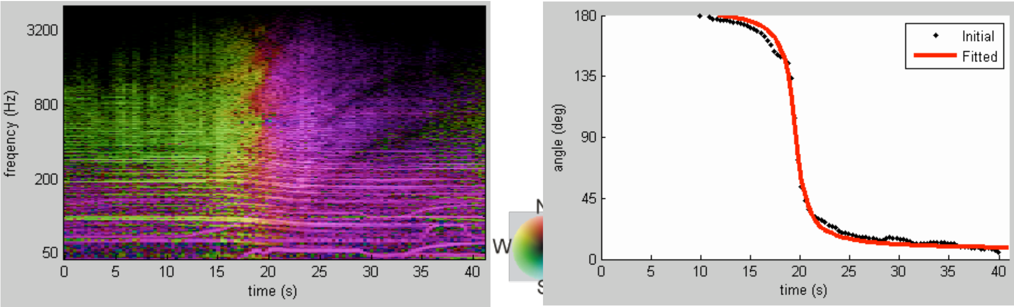


Fig. 7. Left: Time-frequency-angle plot of a rotary wing aircraft. Right: Direction in horizontal plane.

7 Summary

This article considers defense and security applications of the Acoustic Vector Sensor (AVS) with specific attention to the RAM-SCORE system which uses unattended ground sensors (UGS) to determine the locations of launches and impacts at military training grounds and promising results have been presented. Two other platforms have been discussed: the mini-UAV and the ground vehicle. In both cases, the operator can be made aware of any threats and a camera can be pointed at it automatically.

A promising future application of AVS-based signal processing is the use of time-frequency-angle plots to characterize multiple sources simultaneously.

Analysis of Mobile WSNs over IP

Dennis J.A. Bijwaard[1], Paul J.M. Havinga[2,3], and Henk Eertink[4]

[1] Inertia Technology, Offenbachlaan 2, 7522JT Enschede,
dennis@inertia-technology.com
[2] Pervasive Systems, University of Twente, P.O. Box 217, 7500 AE Enschede
P.J.M.Havinga@utwente.nl
[3] Ambient Systems, Colosseum 15d, 7521 PV Enschede,
[4] Novay, Brouwerijstraat 1, 7523 XC Enschede, Henk.Eertink@novay.nl

Abstract. Movement of wireless sensor and actuator networks and of nodes between WSANs are becoming more commonplace. However, enabling remote usage of sensory data in multiple applications, remote configuration and actuation is still a big challenge. The purpose of this paper is to analyse and describe which mobility support can best be used in different scenarios. This paper describes logistic and person monitoring scenarios, where different types of movements take place. These mobility types and their implications are categorized and analysed. Different degrees of support for these mobility types are analysed in the context of the mobility scenarios.

1 Introduction

This paper analyses the different movements that can take place in and across Wireless Sensor and Actuator Networks (WSANs) and of attached devices that provide connection to one or more IP applications. These IP applications can use sensor information from the WSAN as well as configure and actuate the elements of individual nodes. The purpose of this analysis is to gain insight in the different types of mobility and to determine which setup works best in different usage scenarios. A lot of research has been done on mobility within WSANs (e.g. in [1, 8, 9], this paper focusses on mobility issues of: nodes that move between WSANs, WSANs that move in each other's range, and moving IP applications that use the sensor information.

This paper is organized as follows. In section 2 the WSAN types are described where mobility is a concern. In section 3 these WSAN types are used in scenarios where both mobility and shared use by IP applications take place. In section 4 the types of mobility related to WSANs and IP applications are further detailed and the consequences of these mobility types are analysed. In section 5 the level of support for these mobility types required by the scenarios is further analysed. The article concludes how mobility of WSANs can best be handled.

2 Considered mobile WSAN types

In this paper we distinguish the following WSANs types (based on [6]) where mobility and sharing of sensor data can be a concern:

- **Body sensor network (BSN)**: BSNs are sensor networks consisting of few wireless sensor nodes on or around a living being's body integrated with a more powerful device such as a smart phone. Monitoring of vital signs, tracking, and data collection have been the main objectives of these sensor networks. Interaction with sensor-enabled objects [3], such as a dumbbell or ball, is an interesting upcoming usage area. BSNs are small scale, heterogeneous (in terms of different types of sensors) and require single-hop communication. Due to the fact that various types of personal information can be collected by these networks, both security and privacy are major concerns. Reliable data processing and timely feedback are of high importance. Applications using the sensor data can run on the mobile phone or on a server on the Internet (e.g. via connectivity provided by General packet radio service (GPRS)).
- **Structure sensor network (SSN)**: SSNs consist of medium to large numbers of wireless nodes usually attached to buildings (e.g., office), structures (e.g., bridges), infrastructure (e.g., rails) or deployed in specific venues (industrial sites). SSNs may be deployed both indoors and outdoors. Wireless nodes can also be attached to objects moving inside the structure and between structures. SSNs require protection mechanisms against both physical and electronic attacks. They may be both single and multi hop (depending on their scale) and are often heterogeneous (in terms of both sensor nodes functionality and type of sensors).
- **Vehicle sensor network (VSN)**: The sensor data from within a moving vehicle (e.g. a car, boat, train, plane) can also be transferred wirelessly (e.g. via GPRS) to a central server, and be monitored remotely and/or merged with data from other sensor networks.

3 Mobility scenarios

Four scenarios have been defined where different types mobility take place when nodes, complete WSANs or IP applications using the sensor data are moving. Two scenarios are described where a truck with monitored goods moves between distribution centres and two where a monitored person moves around. For both trucks and monitored persons, an IP application can run on the Internet or be directly attached to the WSAN while using information from another IP application running on the Internet. Both a smartphone and router can be the IP gateway (IPG) for WSANs and applications.

3.1 Moving vehicle sensor network

In this scenario, goods in a distribution center are tagged [4] with a sensor node that travels with it when it moves with a truck to another distribution center.

The trucks have a VSN deployed and the distribution centres have an SSN deployed, see figure 1. All sensor data, including Global Positioning System (GPS) location, are provided to the monitoring application. The VSN in Truck 1 may loose its connection to the monitoring application when travelling through low-coverage areas (for instance tunnels) and the IPG will roam to other GPRS network providers when going abroad. The monitoring application would typically offer realtime insight in the conditions of the goods, both when in storage and during transit. Based on condition deterioration, the truck could be re-routed to a closer-by destination.

3.2 Moving vehicle application

In this scenario, truck 2 in figure 1 will have a GPRS connection to the Internet, and the vehicle application may loose connection to the monitoring application when travelling through low-coverage areas and the IPG will roam to other network providers when going abroad. An example vehicle application could monitor the condition of goods in the truck, and compare the measurements with the inventory list to see if nothing is lost, misplaced or spoiled. Via the monitoring application, the vehicle application could check historic conditions of the goods, and location of missing goods or replacements.

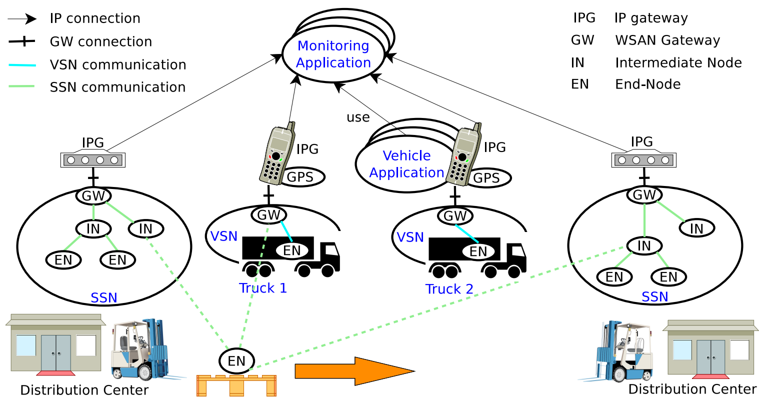


Fig. 1. Monitoring moving goods in logistics

3.3 Moving body sensor network

In this scenario, a man with BSN 2 and smartphone moves between two houses with WiFi coverage and deployed SSN. The man uses objects that have sensor nodes attached that are compatible with the BSN. The BSN is used by a group application running remotely on the Internet (for example monitoring health status and location and may use other monitoring applications), see figure 2. The smartphone will use the cheapest available Internet connection for communication to the Internet, such as WiFi.

3.4 Moving personal application

In this scenario a woman with BSN 1 and smartphone moves between two houses with WiFi coverage and deployed SSN and uses sensor information from these SSN nodes. The BSN is used by a personal application running on the smartphone that the she carries, see see figure 2. The smartphone will use the cheapest available Internet connection for obtaining measurements from a monitoring application. This monitoring application provides real-time sensor information from buildings based on GPS location.

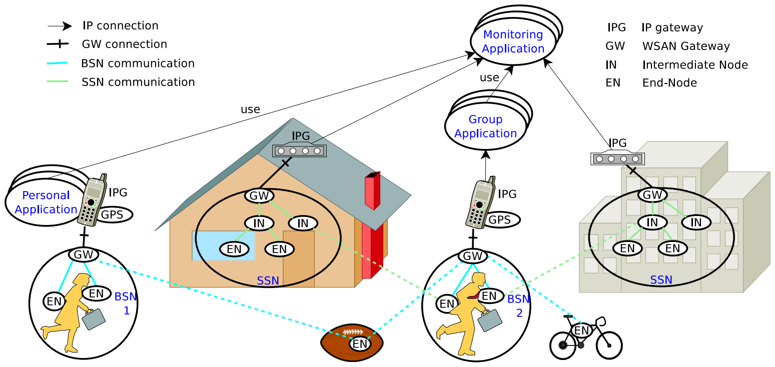


Fig. 2. Moving BSN and personal applications

4 Analysis of mobility types

Since WSAN nodes and its gateway can be attached to different moving objects, multiple types of mobility can occur within and across WSANs. Additionally, a device that hosts an IP application using the sensor data can move. A wireless node can be an end-node that is usually equipped with sensors and/or actuators, or an intermediate node that can extend the coverage area of the WSAN.

This paper makes a distinction between the following WSAN nodes: the **gateway** that makes it available to applications, **intermediate nodes** that extend the coverage of the WSAN gateway, and **end-nodes** that can connect to the intermediate nodes or gateway. Although the paper assumes that the end-nodes do not change to intermediate nodes (like in the Ambient WSAN [2]), most of the mobility types described also apply when they do (such as with the Collection Tree Protocol [5] (CTP)). In the CTP, an end-node can join the WSAN via another end-node, turning the latter into an intermediate node.

The **wireless resources** used by a WSAN are characterised by one or more radio channels and the type of transmission. Examples transmission types are: probabilistic as in Carrier Sense Multiple Access (CSMA), and/or using timeslots as in Time Division Multiple Access (TDMA) and/or frequency hopping.

We distinguish the following types of mobility related to WSANs:

- A moving IPG. Network Mobility takes place when the IPG starts using another wireless or wired network technology or starts using a different network provider on the same network technology. The implication of this change is that the Internet Protocol (IP) address of the IPG changes which will break connections when there is no transparent mobility support (like Mobile IP (MIP)) is in place. For short-lived connections like via HTTP, this connection break will result in a time-out. Movement can also make the IPG unreachable when there is no network coverage, or when it moves into a private or protected network. The moving IPG affects:
  - an attached WSAN. The IPG provides the WSAN with Internet connectivity for applications that want to use info from, configure or actuate nodes in the WSAN. Examples are moving BSNs and VSNs. The implication of movement can be (un)reachability and (dis)connection of IP applications.
  - an attached IP application. An IP application can use sensor data from nearby or remote WSANs via TCP/IP. The IPG movement can break existing connections from the IP application to the WSAN and make others possible.
- A moving WSAN, i.e. a WSAN gateway that may have associated nodes. When the WSAN moves in range of another WSAN, matching wireless resources may require changing these resources in one of the WSANs to avoid bandwidth degradation and possible collisions. Nodes may jump from one to the other WSAN. When moving out of range of another WSAN, the nodes that move with it can stay associated or will associate when they were not yet, non-moving nodes will associate to the non-moving WSAN. When the WSAN moves in range of an intermediate or end-node, that node may decide to join the WSAN when the wireless resources are compatible. When the WSAN moves out of range of an associated intermediate or end-node, the association will be lost.
- A moving intermediate node (with or without connected nodes)
  - within a WSAN, for instance an intermediate node attached to a forklift can extend the radio coverage of the WSAN in the direction it moves and allows end-nodes to communicate. When this intermediate node moves in range of a WSAN gateway or other intermediate node, it has the option to join that WSAN, when it moves out of range it will loose the connection when it was associated. When the intermediate node moves in range of an end-node, the end-node may join when the intermediate node is itself joined. When the intermediate node moves out of range of an

end-node, the end-node will loose its association when it was associated via the intermediate node.
  - across WSANs, for instance an intermediate node attached to a forklift moving between the coverage areas of different WSANs, and picks up goods with attached end-node(s). The intermediate node will join the other WSAN when it is out of range of the other one, an can choose the WSAN when it is in range of both. When it comes in range of another node, that node can choose to join it, when it goes out of range of a node that node will loose its association unless there is alternative intermediate node or gateway in range.
- A moving end-node
  - within a WSAN, the node may have to communicate via different intermediate nodes depending on their radio coverage. When an end-node moves in range of a WSAN or connected intermediate node it can join it. When it moves out of range of a WSAN, it will be disassociated. When it moves out of range of an intermediate node, it will be disassociated unless there is an alternative intermediate node or gateway in range.
  - across WSANs, for instance an end-node that is placed with goods transported between WSAN-enabled distribution centres (see section 3). When an end-node moves in range of a WSAN or intermediate node, it can join it. When it moves out of range of a WSAN it will be disassociated. When it moves out of range of an intermediate node, it will be disassociated when there is no alternative in range.

4.1 Remarks on WSAN mobility types

- Clearly there are a number of options for connected nodes when another WSAN comes in reach, how they deal with this can vary per WSAN type. In section 5 we analyse this further for the given scenarios.
- When different WSAN protocols or wireless resources are used, nodes can not use these links. The gateway may still need to re-allocate resources when the other WSAN operates on the same channel. Section 5 shows different levels of support for overlapping WSANs.
- Without mobility support, complete WSANs and IP applications will disconnect when the IPG changes IP address.
- WSAN nodes can potentially listen to messages in each of the WSAN they become part of, so they can also transfer information from one WSAN to another. Section 5 describes how data protection can be provided.

5 Analysing the mobility scenarios

In this section the mobility scenarios from section 3 are analysed in the light of the different mobility types described in section 4 and the level of mobility support that can be offered.

Important factors for this analysis are:

- **encryption keys**: cryptographic credentials can be used to authenticate a node in a network and to encrypt the traffic, examples of these credentials are keys and passwords.
- **interference**: networks that use the same wireless resources can potentially interfere with each other.
- **awareness**: when a WSAN is aware of the presence of another WSAN it can adapt itself accordingly. Examples of WSAN adaptation are: channel change, synchronisation and distribution of timeslots between WSANs, turning off the gateway, changing mode of operation (for instance change from gateway to intermediate node).
- **mobility**: what do nodes need to do to switch to another network? Clearly this depends greatly on the WSAN type, for instance in the the Ambient WSAN [2], the IPv6 over Low power Wireless Personal Access Networks [7] (6LoWPAN) network, a unique node-id, equal wireless resources and optionally a symmetric key are required for communcation with the WSAN.

5.1 Moving vehicle sensor network

The following levels of mobility support can be offered when the VSN (partly) overlaps with a SSN (depending on the compatibility and awareness in the WSANs):

1. **unaware WSANs**: WSANs that are unaware of each other can potentially disrupt each other when they use the same wireless resources. In this case, the intermediate and end-nodes of both networks may choose to connect to the other sensor network when wireless protocol and encryption keys are compatible.
2. **robust unaware WSANs**: When the WSANs are robust against foreign protocol messaging, they will only suffer a decrease in available bandwidth when they are partly overlapping while using the same wireless resources. When using the same protocol and wireless resources and encryption key, there is no way to stop nodes connecting to the other WSAN and vice-versa.
3. **aware gateway**: VSN gateway turns off and the VSN sensors report to the SSN of the distribution center. This is only an option when the wireless protocol and encryption keys are compatible, and the same wireless resources can be used.
4. **intelligent aware gateway**: the VSN gateway turns into intermediate node. This is an option when compatible wireless protocol and encryption techniques are available, when the same wireless resources can be used, and when an additional intermediate node can be accommodated in the SSN). This ensures better coverage for the sensors inside the truck, but puts more load on the SSN, especially when there are multiple VSN-enabled trucks.
5. **intelligent aware sensor nodes**: the VSN sensor nodes in the truck communicate both to the VSN and the SSN in parallel and may report differently to both networks regarding for instance privacy rules and needs. The communication towards the nodes may become harder, since they will be busy on

the other WSAN part of the time. This will also involve a more complicated scheduler, and multiple WSAN protocol stacks when they are incompatible.

6. **intelligent application**: sensor information is merged elsewhere (i.e. the VSN data is merged remotely with that of both SSNs, for instance in a back-office monitoring application). This requires an indication that the VSN gateway is in range of the SSN gateway to correlate the data, for instance using GPS location. Additionally, the gateways should not interfere too much, for instance use separate channels, different encryption keys, and be robust against foreign protocol messaging.

Additionally, the IPG can change its IP address when it starts using another network technology or another network provider, and no transparent network mobility like MIP is in place. Additionally, Internet connectivity can be temporarily unavailable. This will change the IP address of the IPG or make it unavailable and therefore break existing connections from the WSAN to applications.

Given the required resources in terms of bandwidth and code size on the WSAN nodes and gateway to support mobility awareness and intelligence, and scalability issues with multiple trucks, the following options are the most viable:

- the robust unaware WSANs combined with an intelligent application.
- the aware gateway combined with an intelligent application.

In both cases the different networks can be protected with separate encryption keys. The end-nodes would then require encryption keys for all WSANs they need to operate in, and/or can encrypt its payload such that it can only be decrypted in a specific application.

5.2 Moving vehicle application

When the IPG to which the vehicle application is attached moves, the IPG may connect to different GPRS networks and optionally other wireless network technologies like WiFi. It can also mean temporary unavailability of IP connectivity. The implication of this change of network attachment is often that the IP address of the IPG changes or becomes unavailable, which will break existing connections from the vehicle application or VSN to other IP applications on the Internet.

5.3 Moving body sensor network

The following mobility support options can be considered when a BSN attached to a smartphone moves in range of an SSN and WiFi access point (data protection is an important privacy aspect in BSNs):

1. **WiFi usage**: when the smartphone moves in range of the WiFi access point it can use that for sending BSN messages to the group application instead of the more costly GPRS. The implication is that the IP address of the

smartphone changes and the old connection breaks when no transparent IP mobility like MIP is in place. When multi-homing is supported, the GPRS connection could be kept open while using WiFi. When moving out of WiFi range, GPRS will be used again breaking the WiFi connection to the application.

2. **secured object use**: since objects can potentially listen, store and forward information, communication of more sensitive BSN sensor data should be encrypted.
3. **robust and separate uplink**: BSN and SSN are robust for each other's messaging and use a different uplink. The BSN should use encryption for privacy-sensitive messages and its uplink should use encryption towards the application.
4. **compatible WSANs**: when BSN and SSN are compatible, end nodes may use any intermediate node or gateway to send their information upstream. The information could be encrypted such that only a specific application can decrypt it, for instance by using the public key of the application for encrypting the message payload. Still the destined application for BSN messages should be known to the SSN gateway, since it is most probably different from the application that uses the SSN data.
5. **intelligent BSN end-nodes**: end-nodes that can communicate both with the SSN and incompatible BSN. This can also be used to sent messages with encrypted payload upstream. Here, the BSN message destination also needs to be conveyed to the SSN gateway.

WiFi usage and secured object use are a must for lowering communication costs and enhancing privacy. A robust and separate uplink for the BSN and SSN is the most viable option. Sending BSN messages via a SSN is troublesome, since it needs to be encrypted and somehow addressed to the IP application.

5.4 Moving personal application

The following options can be considered for a moving application (on a smartphone) that uses its attached BSN and nearby SSN data:

1. **Intranet access to SSN gateway**: the SSN gateway could offer direct IP access to sensor data to nearby applications. Access may be possible in the associated Intranet when the smartphone would be allowed in this network, direct access via the Internet is less likely because of firewalls and private networks that are usually in place. Because of local access, the SSN needs to advertise itself in some manner to be discovered by the smartphone application.
2. **public SSN server**: the SSN sends its sensor data to a publicly reachable server on the Internet from which applications can fetch it when they have the proper credentials. Retrieval could for example be based on the current GPS coordinates of the smartphone.

3. **Direct access to SSN nodes**: Intercepting sensor information from the SSN in an BSN end node is not really feasible, since SSN nodes direct their readings only towards the gateway and sleep most of the time to save energy and bandwidth (so requests could take very long). It would also require a compatible protocol, the same wireless resources and encryption keys.

The first two options are both viable. Direct access to SSN nodes is not really an option.

5.5 Conclusions for WSAN mobility scenarios

The following conclusions can be drawn for the WSAN mobility scenarios:

- Support for moving end-nodes between VSNs and SSNs is feasible when all WSANs are controlled by one party. When multiple parties are involved these WSANs are likely to use different encryption keys (or even different protocols). For more flexibility, the end-nodes could be equipped with multiple keys so that they can operate in all WSANs that they have keys for. The downside of this is that the network keys could potentially be obtained from each end-node, so therefore the encryption should work such that the encryption key only makes it possible to send something towards the gateway, not to decrypt everything that is sent inside the network. This can be accomplished by encrypting with the public key of the receiving gateway or the application. When using multiple applications, the gateway (or middleware connected with it) is the best option. Traffic from the gateway to applications can then be encrypted separately.
- It is better to merge BSN and SSN data at the application layer, since obtaining sensor information directly from the SSN proves troublesome and sending private BSN information via the SSN requires usage of SSN protocol and encryption and addressing towards the application.
- Encryption needs to be in place when BSN nodes send privacy-related information, else foreign objects can store and forward them.
- When nodes of different WSAN types move in each other's range, mobility is easier solved at the application layer, unless the WSANs are compatible. In the latter case, the moving WSAN can better turn off its gateway.
- WSAN protocols should be robust against foreign protocols, in order coexist with other WSANs in the same area.

6 Conclusions

This paper analysed scenarios in which different WSAN and application movements take place. Moving end-nodes between different WSANs are easily supported when the networks are compatible. When the encryption or protocol is different in the used WSANs, the end-nodes will need to support all of these encryption types. When compatible WSANs move in each other's range, the moving WSAN can better turn off its gateway and let the end-node directly communicate with the other WSAN. Irrespective of the WSAN type, data of overlapping WSANs can best be merged at the IP application layer instead of via each other. In order to support coexistence of WSANs in the same area, WSAN protocols should be robust against foreign protocol messaging. A way to automatically adapt the used wireless resources to be different from the other WSAN is advisable.

When privacy is required, as it is often the case in body sensor networks, encryption can best be accomplished by encrypting with the public key of the receiving gateway (or middleware), which can in turn sent it encrypted to one or more applications.

References

1. M. Ali, T. Suleman, and Z. Uzmi. MMAC: a mobility-adaptive, collision-free mac protocol for wireless sensor networks. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pages 401 – 407, april 2005.
2. Ambient systems. ambient-systems.net, Last visited August 2011.
3. S. Bosch, R. S. Marin-Perianu, P. J. M. Havinga, M. Marin-Perianu, A. Horst, and A. Vasilescu. Automatic recognition of object use based on wireless motion sensors. In *International Symposium on Wearable Computers 2010, Seoul, South Korea*, pages 143–150, USA, October 2010. IEEE Computer Society.
4. L. Evers, M. J. J. Bijl, M. Marin-Perianu, R. S. Marin-Perianu, and P. J. M. Havinga. Wireless sensor networks and beyond: A case study on transport and logistics. Technical Report TR-CTIT-05-26, Centre for Telematics and Information Technology University of Twente, Enschede, June 2005.
5. O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14, New York, NY, USA, 2009. ACM.
6. N. Meratnia, B. J. v. d. Zwaag, H. W. v. Dijk, D. Bijwaard, and P. J. Havinga. Sensor networks in the low lands. *Sensors*, 10(9):8504–8525, 2010.
7. G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 packets over IEEE 802.15.4 networks. RFC 4944, Internet Engineering Task Force, Sept. 2007.
8. H. Pham and S. Jha. An adaptive mobility-aware mac protocol for sensor networks (MS-MAC). In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 558 – 560, oct. 2004.
9. D. Zhang, Q. Li, X. Zhang, and X. Wang. DE-ASS: An adaptive mac algorithm based on mobility evaluation for wireless sensor networks. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1 –5, sept. 2010.

Exploring Patterns of Activities of Daily Living in the Home Environment

Thijs M. Tönis [1], Harm op den Akker [1], Simone T. Boerema[1], Freek van Polen[2], Hermie J. Hermens[1,3]

[1] Roessingh Research and Development, Enschede, The Netherlands
[2] Almende, Rotterdam, The Netherlands
[3] University of Twente, Enschede, The Netherlands

Background

Senior citizens tend to live longer and longer independently. Judging whether a senior person is still capable of living on his own is often based on the occurrence of incidents, with all consequences thereof. In the specific case of early dementia, the symptoms are not immediately apparent and the occurrence and severity of incidents progress gradually over time. In this case, the children or grandchildren are burdened by the question whether or not the elderly adult can still live safely and independently in his or her own home. This decision is only based on input obtained through incidental visits. We believe that the capability of independent living can only be objectively judged, by a health professional, if long term objective information on the elderly person's daily activities of living (ADL) is available.

The progress in the field of miniaturised wireless sensors makes it possible to obtain real-time, objective information on a user's activities in his home environment in a way that is unobtrusive and respects the user's privacy, to ultimately be able to automatically, and unobtrusively detect slow changes in the behaviour of elderly people living in their own homes. A system, comparable to [1] is needed, one that is able to recognize a broad spectrum of ADL (e.g. cooking, eating, and sleeping) from the output of various types of miniaturised wireless sensors.

Goal

In this work we describe the first step in the development of the system aforementioned: the setup of an experiment in a house fully equipped with miniaturised wireless sensors, capable of detecting ADL performed by a single user. The goal of the experiment was to capture real-time sensor data and simultaneously obtain the gold standard on the subject's current activities in the house. From an enormous and redundant sensor-set we want to identify the minimal set of sensors needed to accurately recognise ADL.

Methods

Five subjects have been living solitary in a fully sensor-equipped house, for five days each, as part of the experimental validation of the wireless sensor network platform developed within the ALwEN project [2]. Figure 1 shows the floor plan of our experimental setup in a normal house, including the installed equipment. The various coloured squares in the figure indicate the placement of the various types of sensors that were used. 1) Pressure sensors were placed under chairs, couches and the bed, to detect the use of this furniture. 2) Magnetic switch sensors were placed on doors, kitchen cupboards and kitchen appliances to detect the state of the doors (open or closed). 3) As the kitchen and stairs don't have doors, their use was detected by light gates, placed at the kitchen pass through and up- and downstairs. 4) Passive infrared sensors were placed in each of the rooms to detect the subject's movement through the house. 5) Temperature and humidity sensors were placed in the kitchen and bathroom to sense environmental changes, indicating cooking and showering. 6) All electrical appliances (e.g. the television, washing machine, and vacuum cleaner) and all the lights were equipped with AC current sensors to detect electrical currents flowing through them, indicating their use. 7) Additionally the washing machine, iron and vacuum cleaner were equipped with a 3D-accelerometer, to detect its usage. 8) Finally, the subject was wearing a 3D-accelerometer and a heart-rate sensor.

A gold standard is needed to define the truly performed ADL. To this end, seven cameras were placed, covering most of the living spaces, excluding the toilet and bathroom because of privacy. Video images will be annotated by two independent researchers, making it possible to retrospectively identify all ADL performed by the subject. From this we can validate the activities recognised based on the sensor dataset using clustering techniques. The large corpus of sensor data also gives the opportunity to study the minimal sensor set needed to detect ADL with an acceptable accuracy.

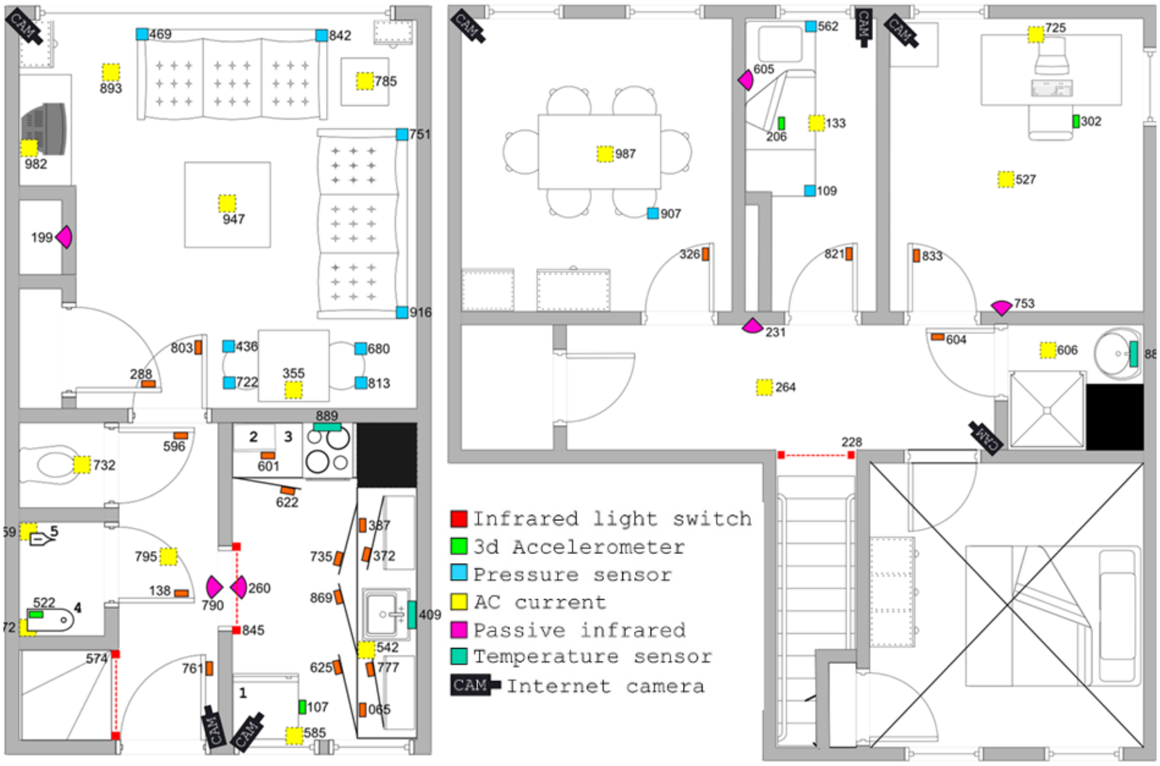


Figure 1: Floor plans of the ground floor (left) and upper floor (right) of the experiment house, including sensors and cameras.

Results and Discussion

The experiment took place from March 2011 till June 2011. In this period we have created a large corpus of sensor data from five healthy subjects living in the equipped home environment. In total we gathered sensor and video data of 36 days (576 hours). All 63 sensors send each minute their state of the previous minute, which was stored in the database. Although no results based on the dataset are available yet, we have shown that wireless sensor technology has great potential to be applied in home monitoring environments. In our opinion the use of a gold standard (camera system) is essential during this early stage of research towards an ADL monitoring system. We believe a monitoring system like this should be wireless in order to be easily applicable in any home environment. The wireless nature of the system makes it vulnerable to data loss. Data analysis will show whether our redundant sensor setup is sufficiently resistant against these errors.

The next step will be to create the gold standard ADL data set by hand-annotating the video data on all relevant events, ranging from “door open”, “television on”, to more complex compound events such as “cooking” or “cleaning”. Using the gold standard data we can experiment with machine learning techniques that have already proven their merit in earlier studies such as Support Vector Machines [3] in order to find the best method of training classifiers for the various events and evaluate them properly. From the recognized activities, “activity-patterns” (such as daily or weekly patterns) must be extracted, and compared to some definition of a healthy lifestyle for a particular user. The final step of automated extraction would be to automatically detect slow changes in the recognized activity patterns in order to be used in the detection of early stage dementia.

References

[1] Intille, S. S., Larson, K., Tapia, E. M., Beaudin, J., Kaushik, P., Nawyn, J., & Rockinson, R. (2006). Using a Live-In Laboratory for Ubiquitous Computing Research. *Pervasive Computing*, *3968*, 349-365.

[2] ALwEN project website. www.alwen.nl

[3] Zhao, L., Wang, X., & Sukthankar, G. (2010). Recognizing household activities from human motion data using active learning and feature selection. *Technology and Disability*, *22*(1-2), 17-26.