SAFH – Smooth Adaptive Frequency Hopping

Sami Ben Cheikh *, Tim Esemann† and Horst Hellbrück†
* Aalto University Helsinki, Finland
School of Electrical Engineering
Email: sami.ben.cheikh@aalto.fi
† Lübeck University of Applied Sciences, Germany,
Department of Electrical Engineering and Computer Science
Email: hellbrueck,esemann@fh-luebeck.de

***Abstract*—Wireless systems based on WLAN (802.11), ZigBee (802.15.4) and Bluetooth (802.15.1) are continuously deployed in new applications covering consumer, industry or medical fields. Especially, Bluetooth is recommended by the Health-Care-Organization for medical applications as frequency hopping is considered as a robust scheme. However dealing with frequency-dynamic sources of interference in the 2.4GHz ISM band is important due to the increase of wireless devices. Adaptive frequency hopping (AFH) suggested by the Bluetooth standard and implemented in many of todays products identifies and avoids using bad channels. It is a good and established coexistence mechanism in the presence of frequency-static sources of interference such as WLANs when the 2.4GHz band is not crowded. However, AFH is facing problems in a crowded 2.4GHz band, especially when the interference is dynamic. We developed a cross-layer algorithm SAFH (Smooth Adaptive Frequency Hopping) that is inspired by entropy maximization and the conventional Bluetooth AFH. SAFH assigns usage probabilities to all channels based on an exponential smoothing filter for frame error rates to estimate and predict the channel conditions. The application layer can adapt SAFH by parameter settings in a cross-layer approach. SAFH achieves low average frame error rate and responds fast to changing channel conditions if required from the application. Simulative Evaluation in the presence of different types of interference (802.11b, 802.15.4 and 802.15.1) shows that our algorithm outperforms conventional frequency hopping and AFH. Additionally, SAFH works smoothly and stable exploiting frequency diversity compared to previous approaches like entropy-maximization based adaptive frequency hopping and Utility Based Adaptive Frequency Hopping (UBAFH).**

I. Introduction

Bluetooth (802.15.1) has been developed ten years ago. Since then the operation conditions of Bluetooth have changed. New standards like 802.15.4 and many proprietary solutions for wireless systems like wireless mouse or remote controls are available in off-the-shelf products. Additionally, standards like 802.11 have been extended by additional physical layers providing higher data rates.

To adapt to these changes Bluetooth standard has been extended and products are modified in new versions. In the first version of Bluetooth frequency hopping has been considered as a robust solution. In the second step in [1] adaptive frequency hopping (AFH) is implemented to reach coexistence with other wireless standards like 802.11.

However, with the increasing density of heterogeneous wireless systems in the 2.4GHz band AFH reaches its limits as shown in [2], [3], [4]. One of the assumptions of AFH is that interference is frequency-static and that the 2.4GHz band is not crowded. These assumptions are not valid for the future anymore with many competing wireless systems evolved in the past. 802.15.4 is not considered in AFH at all. Consequently, there is a next step required to prepare Bluetooth for the future leading towards cognitive radio. Some improvements have been suggested in the past for solving this problem of interference. Entropy-maximization based adaptive frequency hopping [5], [6] is one of the approaches that we will discuss as related work and compare in the evaluation part.

The contributions of this paper are threefold: First we introduce a smoothing estimation and prediction of channel conditions based on exponential filtering for frame error rates. Second based on this estimation SAFH assigns usage probabilities to all channels with a new cross layer scheme. The goal of this approach is (1) smooth operation, (2) lower average FER, (3) fast response to changing channel conditions, and (4) stable operation adaptable by the application layer. The last contribution of this paper is a comprehensive evaluation in a dynamic environment of changing 802.11, 802.15.4 and Bluetooth interference.

The rest of the paper is organized as follows: Section II will discuss related work. Section III introduces SAFH step by step. In Section IV and Section V we will describe the simulation setup and evaluation results. The paper will conclude with a summary and outlook in Section VI.

II. Related Work

In the past several papers were published covering improvement of frequency hopping in presence of frequency-static and dynamic interference. In [7] a partitioning of the original Bluetooth hop band is proposed to reduce the collisions between piconets. The described algorithm Orthogonal Hop Set Partitioning (OHSP) suggests five orthogonal sub-hopsets. Each master chooses randomly and independently one of these sub-hopsets. For best case combination of sub-hopsets the throughput could be improved by 10%, but in worst case combination the algorithm shows even degradation. Therefore on average there is only a marginal improvement.

Popkovski et al. propose a dynamic adaptive frequency hopping (DAFH) algorithm in [8]. According to the observed packet errors each piconet adapts its hopset to a subset of good frequency channels to minimize the interference. The drawback of OHSP and DAFH is that the hopset size is reduced and therefore the frequency diversity is decreased. This results in longer channel occupancy and therefore higher probability of collisions with concurrent radio links degrading their operation.

Stabellini et al. propose a so-called Utility Based Adaptive Frequency Hopping (UBAFH) [9], [10]. UBAFH maps the observed packet error rate (PER) to a probability density function that defines the usage probability of each channel. In [9] the algorithm is evaluated for frequency selective fading channels and in [10] it is additionally evaluated under frequency-dynamic interference. In Section V we will compare our algorithm presented in Section III with UBAFH.

A novel approach called Robust Adaptive Frequency Hopping (RAFH) algorithm in [5], [6] mitigates transmission errors by assigning different usage probabilities for good and bad channels. In this probabilistic approach a constrained entropy maximization problem is solved. RAFH combats frequency-dynamic interference - thats what the authors target for by exploiting frequency diversity. As shown in [5], [6] RAFH outperforms Adaptive Frequency Hopping (AFH) by reducing the packet error rate (PER) fluctuation. RAFH cannot improve the average PER compared to AFH. The way the probability distribution is calculated leads to a convex optimization problem that requires a lot of computation power. Therefore, this approach is not suitable for low cost low power devices. Moreover the work does not consider frequency-static interference. RAFH still exhibits fluctuations and does not converge to the target threshold under stringent requirements.

The performance of our proposed algorithm SAFH is compared in Section V with the AFH, UBAFH and RAFH.

III. Algorithm Description

In our coexistence model, we considered three types of systems: 802.11b networks, 802.15.1 Bluetooth piconets and 802.15.4 networks. 802.11b and 802.15.4 use direct-sequence spread spectrum (DSSS). In 802.11b the bandwidth of channel is 22MHz around the center frequency. 802.11g/n use Orthogonal Frequency Division Multiplexing (OFDM). Typically, channel 1 (2412MHz), 6 (2437MHz) and 11 (2462MHz) are used as non-overlapping channels.

802.15.4 operates in any of 16 channels, each with a bandwidth of 2MHz and a channel separation of 5MHz ((2405+5(k-11))MHz and $k \in \{11..26\}$). Bluetooth uses frequency-hopping spread spectrum (FHSS) with $k = 79$ (1MHz) channels in pseudo random manner ((2402+k)MHz, where $k \in \{0..78\}$). Additionally, there are proprietary protocols that also create unpredictable interferences.

To increase the robustness Bluetooth breaks this random hopping to avoid bad channels which are used by other wireless systems. Therefore, algorithms work in several steps. SAFH as illustrated in block diagram in Figure 1 has 4 steps: channel classification, channel prediction (which is new), mapping and hopset generation with 5 parameters that we will explain in more details in this section. RAFH and UAFH have only three steps: channel classification, mapping and hopset generation; AFH only two: channel classification and hopset generation.

In the first step the algorithm starts with a channel classification by assessing the ISM spectrum to determine the presence of interference. The performance metric is frame error rate (FER) as the average percentage of frames with bit errors or lost frames. In literature the terms frame error rate and packet error rate are used synonymously, we will use FER for the rest of the paper. Calculating the FER by means of time average is recommended by the 802.15 wireless personal area networks Task Group (TG2) [11], since instantaneous measurements would result in overhead in the synchronization between the transmitter and receiver and consequently in more power consumption. Moreover we minimize the risk of incorrect classification due to instantaneous disturbances e.g. other frequency hoppers. The channel status of each of the 79 channels is exchanged between the transmitter and the receiver on a regular basis, because they do not necessarily experience the same channel conditions.

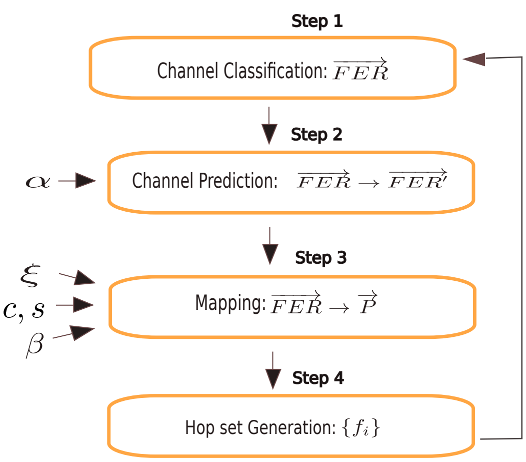

Fig. 1. Block Diagram of the SAFH Algorithm

In the second step of our algorithm – in contrast to all other approaches – exponential smoothing is used to predict the frame error rate of all the channels, (FER'_i, $\forall i \in \{1..N\}$). This popular forecasting approach, first used by Brown [12] to track the location of submarines, computes a moving average, where all data contribute to the smoothed value and decrease exponentially. Equation 1 calculates the predicted frame error rate at time $(t + 1)$ by using the measurement at time t plus an adjustment of the last forecast.

$$\begin{cases} FER'_i(t+1) & = \alpha * FER_i(t) + (1-\alpha) * FER'_i(t) \\ FER'_i(1) & = FER_i(0) \longrightarrow \text{(initial condition)} \end{cases} \quad (1)$$

α is a smoothing factor from the interval $[0 : 1]$ controlling the balance between new and old data. With α approaching 1, the filter gives more weight to recent data and reducing smoothing effect. With α approaching 0, the effect of the current observation is reduced and we return the smoothed past values.

In the third step the different algorithms black list individual channels (AFH) or use them with a reduced probability (RAFH, UBAFH, SAFH) to probe the channel conditions. Therefore, we map the predicted $FER_i'(t+1)$ to a discrete probability distribution meeting the goals of the SAFH algorithm:

Condition 1. *The target average FER must not exceed a threshold ξ given by the application:*

$$\sum_{i=1}^{N}(p_i(t+1) * FER_i(t+1)) \leq \xi \tag{2}$$

Condition 2. *The probability assigned to a channel is a decreasing function of its FER i.e if $FER_i \geq FER_j$ then $p_i \leq p_j$, $i, j \in 1..N$ to reach a low FER.*

A mapping function that fulfills the above conditions is shown in Equation 3:

$$\begin{cases} p_i & = (\beta + c * d_i)/\delta \text{ if } d_i \geqslant 0 \\ p_i & = (\beta + s * d_i)/\delta \text{ if } d_i \leqslant 0 \end{cases} \tag{3}$$

In what follows, we will explain the meaning of the following terms c, s, d_i and β. The first term d_i, is the difference between the FER of channel i and the threshold i.e. $= \xi - FER_i$, as shown Figure 2. A positive value of this metric indicates that we have a good channel, otherwise the channel is considered bad.

Instead of avoiding a bad channel completely, we assign a probability that is a function of its distance from the threshold. We also use two optional parameters, c and s; if not used default values are 1. They define the ratio of rewarding good channels and punishing bad channels using a carrot-stick approach. The larger c the more often good channels are used, which will reduce the probing of the bad channels.

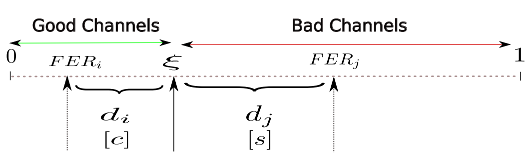

Fig. 2. Mapping of FER into probability mass function

The term δ is a normalizing factor i.e it ensures that $\sum_{i=1}^{N} p_i = 1$; moreover it guarantees that $p_i \leqslant 1$, i.e.

$$\delta = \sum_{i=1}^{N_g}(\beta + c * d_i) + \sum_{i=1}^{N_b}(\beta + s * d_i) \tag{4}$$

N_g and N_b are the numbers of good channels and bad ones respectively. The last term of Equation 3 is β. It is chosen in a way that the constraint on the average FER is fulfilled. This is accomplished by plugging the values of p_i and δ from Equation 3 and Equation 4 respectively in Equation 2, to solve for β. The obtained value of β is a function of the predicted FER, therefore it is a function of time. It is possible to fix β to a value between 0 and $1-\xi$ as shown in Figure 3. However we will not guarantee that the average $\overline{FER}$ will be below the threshold. Moreover, the channels whose FER exceeds $\beta + \xi$ will not be used, since otherwise the probability would be negative.

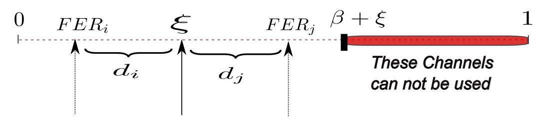

Fig. 3. Mapping of FER when β is constant

We tried different values for β and interestingly the special case when $c = s = 1$ and $\beta = 1-\xi$ resulted in the probabilities $p_i = (1 - FER_i)/\delta$ which represents the same probability mass function used by the UBAFH algorithm described in Section II.

In the last step of the algorithm the probability distribution is mapped to a hopset using the inversion methods [13]. The fact that the parameters α, ξ, c, s, β can be changed on the fly can be beneficial for performance adaption requested by higher layers.

IV. Simulation Setup

To evaluate the performance of the algorithm we choose a setup based on the RAFH evaluation in [5]. Thereby, our results are comparable to previous work. The setup consists of one Bluetooth piconet under test running various algorithms including our SAFH. Three 802.11b (with the following traffic loads also referred to as duty cycle Ch1 28 %, Ch6 6.3 %, Ch11 30 %) and additionally three 802.15.4 (Ch15 0.8%, Ch20 0.3%, Ch25 0.8%) networks are considered as frequency-static interference and 12 Bluetooth piconets are considered as frequency-dynamic interference. Frequency bands of 802.11b and 802.15.4 interference do not overlap in our setup. The traffic load of the interfering systems is a very important factor, since collision between devices is caused by overlaps in time and in frequency. Being packet based 802.11b and 802.15.4 devices create bursts of activity at a poisson rate. We configured the corresponding transmitters with specific mean packet rate and mean packet length to calculate the traffic load.

The piconet under test uses single slot voice packets (HV1) that are protected by 1/3 FEC. In HV1 the radio hops into a new channel each second slot, while HV2 and HV3 hop every fourth and sixth slot respectively (resulting in the following traffic loads HV1 50%, HV2 25% and HV3 17%).

We implement the scenario in the well-established MATLAB/SIMULINK software package. Additive white Gaussian noise (AWGN) is used to model the noise at the Bluetooth receiver. The fading is flat i.e. frequency non-selective. The propagation model consists of two parts (1) line-of-sight propagation for the first 8 meters and (2) a propagation exponent of 3.3 for distances above 8 meters.

The default setup studied in this paper assumes that all the 802.11b and 802.15.4 networks are switched on. The hopset length T of the Bluetooth under test consists of 1000 hops and the 12 Bluetooth interferers use different hopping patterns. The default simulation run lasts $30T$, where T is the time needed to hop through all the channels in the hopset

V. Results and Discussions

In this section we present and discuss the results obtained from the setup introduced in previous section. In the scenario under test we require a reliable Bluetooth voice communication which needs FER between 5% to 10% [14].

In Figure 4 we illustrate the performance of SAFH under the default topology (frequency-dynamic and static interference) by modifying some of the algorithm parameters. The algorithm UBAFH [9] is a special case of our algorithm, when $c = s = 1$ and $\beta = 1 - \xi$ as mentioned in Section III. With these parameters the probability distribution is a function of the FER only. Therefore, it does not attempt to meet the constraint on the average FER. For other fixed values of β, we noticed better performance for smaller values than for larger ones. This will result in reduced frequency diversity, since smaller β results in more channels not being used as illustrated previously in Figure 3.

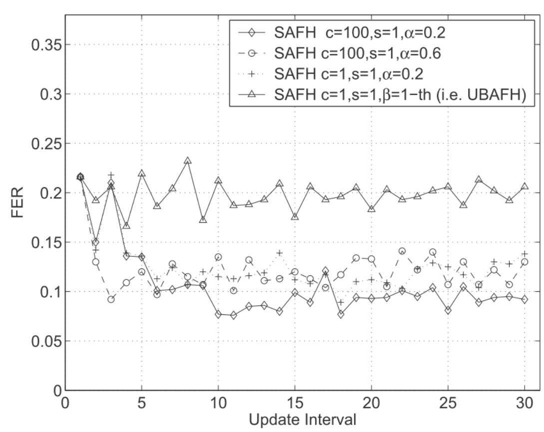

Fig. 4. Performance of SAFH using various Parameters

Figure 4 illustrates the faster convergence of the SAFH algorithm for large values of α. This is due to the characteristic of an exponential smoothing filter with number of time periods $\propto \frac{1}{\alpha}$ needed for convergence. Furthermore it can be seen that a larger value of c yields in a lower average FER.

We optimize the parameter set for values of $c = 100$, $s = 1$ and $\alpha = 0.2$ that provide a good trade-off between FER and fast convergence. With this α the filter suppresses the effect of the frequency dynamic interferers (12 Bluetooth) successfully.

Figure 5 shows a comparison of the performance of SAFH, RAFH and AFH. The performance of RAFH and AFH are comparable to previous results in [5] due to the same implementation and similar setup. We can clearly see how SAFH outperforms the other algorithms with respect to frame error rate (FER). SAFH reaches an average FER of 10% compared to AFH (15%) and RAFH (18%) In Figure 5 we can also see the fluctuations of AFH which is the result of resetting bad channels every other interval for probing. RAFH also fluctuates since it updates the usage probabilities of the channels whenever it is unable to meet the threshold. In contrast, SAFH is stable and operates smoothly compared to RAFH and AFH.

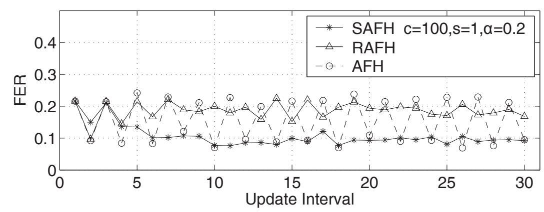

Fig. 5. Performance of SAFH versus RAFH and AFH

The stringent requirement on the FER threshold of less than 10% was the reason that RAFH calls its update routine at every interval. Moreover RAFH uses only the FER of the most recent interval, thus it does not take advantage of the history of the channels. Therefore, SAFH with exponential smoothing filter reaches a notable improvement compared to the other approaches. The second reason that affects the performance of RAFH is the fact that probabilities are assigned to the channels based on the FER of the current interval. We tackled this issue in a simple and efficient way by first predicting the FER of the upcoming interval. Based on the forecast FER we assign the probabilities to the channels. The results shown in Figure 5 clearly indicate the gain in performance. Moreover we do not have to solve a convex optimization problem like RAFH, thus reducing the computational complexity compared to RAFH.

Table I presents the average FER of all the algorithms, when Bluetooth piconets are not present i.e. the only interferes are 802.11b and 802.15.4 networks (frequency-static interference).

TABLE I
Average FER with Frequency-Static Interferences

SAFH c=100 s=1 $\alpha = 0.2$	**SAFH** c=1 s=1 $\alpha = 0.2$	**SAFH** c=100 s=1 $\alpha = 0.6$	**AFH**	**RAFH**	**UBAFH**
0.100	0.095	0.075	0.100	0.115	0.145

SAFH in the first column (parameters: $c = 100, s = 1$ and $\alpha = 0.2$) is the worst candidate but reaches at least the average FER of AFH and performs better than RAFH and UBAFH. Other parameter sets of SAFH (in column two and three) perform better. The results in Figure 5 and Table I show that the optimal parameter set is depending on the type of interference. The optimal combination of the parameters depending on current spectrum occupancy will be investigated in more details in our future work.

Figure 6 illustrates the channel usage of SAFH, RAFH and AFH in the default setup. The x-axis in the figure ranges from 0 to 78, where 0 corresponds to channel 2402MHz in the ISM band, which represents the first Bluetooth channel in the spectrum. The 802.11b networks centered at channel 1, 6 and 11 are identified successfully by all the algorithms. The main performance difference is how often one algorithm uses bad channels compared to the others. Figure 6 shows that SAFH uses bad channels less than all the other algorithms. This was in fact a key requirements that we set while designing our algorithm to achieve better performance and Figure 6 demonstrates that we have achieved this goal.

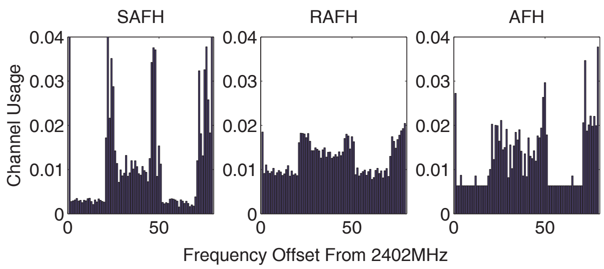

Fig. 6. Channel Usage of SAFH, RAF and AFH

The last evaluation criterion is the responsiveness of SAFH to sudden change of interferences. To measure this parameter we switched one 802.11b transmission off at time $10T$ and then back on at time $20T$. Remember that 802.11b is the main source of interference in our setup. In Figure 7 at time $T = 11$ turning 802.11b off results in a slightly lower FER. When we switch the device back on at time $21T$ an FER peak ($T = 21$, $FER = 14\%$) occurs that drops down within less than $3T$ as the algorithm converges.

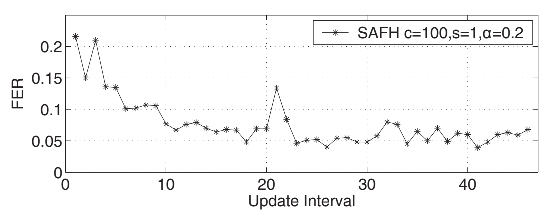

Fig. 7. Response of SAFH to change in Interference; Switch off 1 WLAN at $T = 11$ and again on at $T = 21$

VI. Conclusion and Future Work

In this paper we motivated the need for a new adaptive cross layer frequency hopping algorithm. The goal was to reach smooth operation in the presence of frequency-static and dynamic interference. We propose a Smooth Adaptive Frequency Hopping (SAFH) algorithm that can be adapted from application layer by parameter like setting a maximum frame error rate (FER) threshold. SAFH has two basic building blocks: (1) Smooth estimation of FER and prediction of channel conditions based on exponential filter and (2) efficient assignment of usage probabilities to all channels. The simulative evaluation of the proposed SAFH scheme compared to adaptive frequency hopping, utility based adaptive frequency hopping and robust adaptive frequency hopping results in a smooth operation, lower average frame error rate, fast response to changing channel conditions due to dynamic interference, and stable operation.

For the future we will implement the algorithm in a transceiver and provide measurement results. Additionally, we will also include overhead calculation for the exchange of the channel states. We further evaluate the parameter sets of SAFH to give advice for the optimum settings for various frequency static and dynamic interference.

Acknowledgments

This work has been supported by the Federal Ministry of Education and Research of Germany: Förderkennzeichen 17N3809, SoFT.

References

[1] *Specification of the Bluetooth System Version 4.0 [Vol 0]*, Bluetooth SIG Std., 2009.
[2] T. Esemann and H. Hellbrück, "Limitations of frequency hopping in 2.4 GHz ISM-band for medical applications due to interference," in *Proceedings of the Workshop Consumer eHealth Platforms, Services and Applications (CeHPSA)*. IEEE CCNC 2011 Conference Proceedings, 2011, accepted for publication.
[3] N. Golmie, N. Chevrollier, and O. Rebala, "Bluetooth and WLAN coexistence: Challenges and solutions," *IEEE Wireless Communications Magazine*, vol. 10, pp. 22–29, 2003.
[4] A. Mathew, N. Chandrababu, K. Elleithy, S. Rizv, and L. Almazaydeh, "Interference of 802.11b WLAN and Bluetooth: Analysis and performance evaluation," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 2, pp. 140–150, 2010.
[5] K.-J. Park, T. R. Park, C. D. Schmitz, and L. Sha, "Entropy-maximization based adaptive frequency hopping for wireless medical telemetry systems," in *Proc. 1st ACM International Workshop on Medical-grade Wireless Networks WiMD*, 2009.
[6] K. J. Park, T. R. Park, C. D. Schmitz, and L. Sha, "Design of robust adaptive frequency hopping for wireless medical telemetry systems," *IET Commun.*, vol. 4, no. 2, pp. 178–1191, 2010.
[7] Z. Jiang, V. C. Leung, and V. W. Wong, "Reducing collisions between Bluetooth piconets by orthogonal hop set partitioning," in *Proc. IEEE Radio and Wireless Conference RAWCON*, 2003.
[8] P. Popovski, H. Yomo, R. Prasad, and S. Member, "Dynamic adaptive frequency hopping for mutually interfering wireless personal area networks," *ACM Mobihoc*, vol. 4, pp. 991–1003, 2004.
[9] L. Stabellini, L. Shi, A. A. Rifai, J. Espino, and V. Magoula, "A new probabilistic approach for adaptive frequency hopping," in *20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC*. IEEE, 2009, pp. 2147–2151.
[10] S. Luca, "Toward reliable wireless sensor networks : Energy-aware distributed interference management for unlicensed bands," Ph.D. dissertation, KTH, Communication Systems, CoS, 2010.
[11] *Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands*, IEEE Std. 802.15.2, 2003.
[12] R. G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*. Courier Dover Publications, 2004.
[13] L. Devroye, *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.
[14] T. Bourk, "Techniques mitigate interference between 802.11 and Bluetooth," "http://www.design-reuse.com/articles/4288/techniques-mitigate-interference-between-802-11-and-bluetooth.html", online; accessed 12-Mars-2011.