A Dynamic Interference-Avoidance Algorithm for Frequency Hopping Systems

Ivo Stojan

Abstract: **In this paper we investigate an algorithm for the Adaptive Frequency Hopping mechanism that is used by frequency dynamic systems to mitigate interference from other systems. With this algorithm we introduce several improvements in relation to the existing algorithms that are based on the approach of using Packet Error Rate as the means for channel classification. One is the use of a single criterion for channel classification regardless of the dynamics of interfering systems, which adds more flexibility and reduces the risk of erroneous channel classification. The second is the introduction of the concept of channel probing which ensures that channels that are excluded from the hopset are not used until they are clear from interference. The third improvement is the parameterization of the algorithm, which enables the control of the trade off between the main achievements of the algorithm: throughput and quickness of adaptation to changing interference. We show these achievements of the proposed algorithm through simulation.**

Index terms: **frequency hopping systems, interference mitigation algorithms**

I. INTRODUCTION

When Adaptive Frequency Hopping (AFH) is considered as the interference avoidance mechanism for frequency dynamic systems, e.g. Bluetooth, the following factors need to be taken into consideration in the design of the operational algorithm: the channel classification means, the extent of hopset size reduction and the ability to avoid both frequency static and frequency dynamic interference and quickness of adaptation to changing conditions.

In the Bluetooth specification [1] only general principles for channel classification are given. Concrete classification means were proposed in [2]: carrier sensing, RSSI measurement and PER measurement. The simplest and most widely used of these methods is PER measurement. Classification of the channels is performed by measuring the Packet Error Rate (PER) for each channel in the hopset (PER_i); a Bluetooth packet is declared as erroneous if any of the following fails: packet access code, HEC, or CRC.

An important issue when using AFH is the extent of hopset size reduction. By reducing the hopset size to only a small number of channels from the available spectrum, the devices deploying AFH cause a high level on interference on these channels. This may completely disable the operation of neighboring devices using these channels, especially when it comes to lower power devices that do not cause much interference to the disrupting system. The hopset size for the 2.4 GHz ISM frequency band is regulated by [3].

Depending on the dynamics of frequency spectrum usage, an interfering system can be classified as frequency dynamic (FD), such as Bluetooth, or frequency static (FS), such as different variants of WLAN. It is necessary that an AFH algorithm is able to mitigate interference from both interferer types.

We also see the quickness of adaptation to changing interference as an important asset of an AFH algorithm. Some applications such as voice communication would benefit from quick adaptation to changing conditions, while for others, such as file transfer, the achieved throughput is a more important factor.

This paper is organized as follows. In Section 2 related work regarding AFH algorithms is discussed. In Section 3 a new variation of PER based algorithm is presented. Section 4 presents the simulation results, and Section 5 concludes this paper.

II. RELATED WORK

An approach for mitigating FD interference is hopset orthogonalization. The basic idea is to divide the available spectrum into orthogonal hopsets. The approach proposed in [4] predefines five hopsets and each piconet would randomly choose one of these hopsets. This approach was adapted in the form of Adaptive Frequency Rolling (AFR) [6]. In AFR, a piconet rolls its hopset through the entire available spectrum, thus achieving better spectrum dynamics. However, it does not inherently posses the ability to avoid FS interference. The same paper also presents an approach for avoiding both FD and FS interference, called Dynamic Adaptive Frequency Hoping (DAFH). DAFH is based on hopset splitting and doubling. The problem of hopset splitting is that it may result in an over-shrunk hopset which is in collision with FCC regulations regarding etiquette rules in the 2.4 GHz band [3].

Another approach for avoiding both FD and FS interference, called Enhanced Adaptive Frequency Hopping

(EAFH), was presented in [7]. FD interference is mitigated by moving the channels between three channel groups (one-slot, multi-slot and excluded channels) based on how PER_i relates to the mean PER value ($\overline{PER}$). FS interference is mitigated by excluding channels for longer time periods when PER_i is greater than 0.5. We believe that it is not flexible to hard code the threshold that would be used to differentiate FS from FD interference. In case of a larger number of FD interferers, PER_i caused by FD interference may be close to the values for FS interference, especially when channels used in shrunk hopsets coincide. This algorithm also hard limits the minimum hopset size to a value greater than required in Bluetooth specification. The algorithm should do its best to keep the highest possible value, but the hard limit should be as specified in [1]. Also, the relation between $\overline{PER}$ and PER_i that determines the FD threshold for moving channels between the defined groups is not clearly defined or discussed. Finally, we see an issue regarding the reuse of excluded channels without any knowledge of the new channel state.

III. THE PROPOSED ALGORITHM

The algorithm proposed in this paper is also based on the relation between PER_i and $\overline{PER}$ as the channel classification means. We extend this approach with the following:

1. We define a common criterion for channel classification that can be used for both FS and FD interference, which reduces the risk of erroneous channel classification.
2. We introduce the concept of channel probing which ensures that channels that are excluded from the hopset are not used until they are clear from interference.
3. By introducing parameters in the algorithm we enable the control of the trade off between the achieved throughput, the extent of hopset size reduction and the quickness of adaptation to interference.

The state chart of the algorithm for channel i is given in Fig.1. The criterion for channel classification has the following form:

If $PER_i > \overline{PER} * \alpha$, classify the channel as bad.

Parameter α regulates the relation between the extent of hopset size reduction and the achieved decrease of $\overline{PER}$; by choosing a smaller value more channels are classified as bad thus reducing the hopset more but improving $\overline{PER}$ and throughput.

The criterion can be evaluated after each received packet, and is used to move a channel to the bad channel group only when hopset size is larger than the minimal size.

The time during which a channel is classified as bad is determined in the following way. When a channel is classified as bad using the criterion stated above, it will not be used in the hopset for an amount of time that depends on the value of PER_i at the moment of classification:

$$T = MPP * PER_i$$

Maximal Probe Period (MPP) is a parameter that regulates the trade off between the overhead of probing and quickness of response to changing conditions in the spectrum; by choosing a greater value, probing overhead is reduced and throughput increased, but response to change is slower. MPP is expressed in terms of time slot pairs (duration of each Bluetooth time slot is $625\mu s$).

After time T for a certain channel has elapsed, the channel is probed to determine if it can be returned to the good channel group. Probing is performed according to the same classification criterion as described above. However, the channel that is being probed does not affect $\overline{PER}$ or hopset size, i.e. it does not affect the parameters used by the classification criterion until it is moved to the good channel group. If the probation fails, the channel stays in the bad channel group. Time T is calculated again using the PER_i measured during the probation period, and compared to the old value of T. The value that will be used for the next period in the bad channel group is the one which is greater.
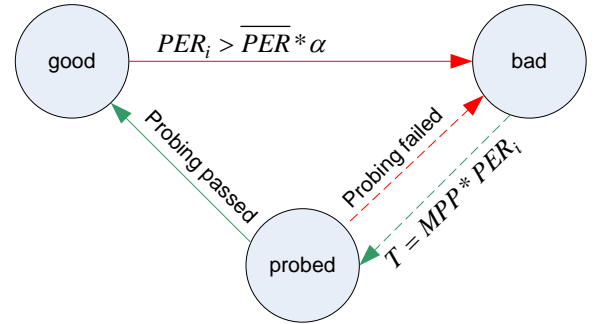

Fig.1. State diagram for the proposed algorithm

IV. SIMULATION

Simulations of the pseudorandom and of the proposed AFH algorithm were performed using MATLAB Simulink and Stateflow. The simulation model consists of four Bluetooth piconets, and two WLAN interferers, all of which can be turned on and of. Each piconet consists of one master and one slave device.

Three variants of the simulation model were created:

*   A model based on basic rate radio (GFSK modulation) and DM1 packet type (2/3 FEC error correction for the payload).
*   A model based on EDR radio (DQPSK modulation for the payload) and 2-DH1 packet type (no payload error correction).
*   Another model based on EDR radio (DQPSK modulation for the payload) and 2-DM1 packet type which is not specified in [1]. This packet type is defined in [5] as an equivalent to DM1, i.e. it is a one slot EDR packet whose payload is DQPSK modulated and which has 2/3 FEC error correction. We use this packet type in order to compare the performance of packet types with and without error correction, in bad channel conditions.

Beside FEC error correction, the simulation model also implements error detection (HEC and CRC), and the ARQ scheme with retransmit filtering.

Each piconet is assumed to be in connection state, with randomly generated payload and a 100% duty cycle .i.e. a Bluetooth packet is sent in each time slot.

Hopset determination is performed by the master of each piconet, using only its own channel classification. The information about the determined hopset is communicated to the slave using one LMP message for each information exchange. Since the time slot used by this message cannot be used for user data, the exchange of hopset information should be as seldom as possible. We have set the period of this information exchange to 20 time slots. This period was chosen because minimal hopset size is 20, which means that in a period of 20 time slots no channel should be used more than once or twice. Hence the probability that a channel classified as bad is used again in the same period of 20 time slots is relatively low.

WLAN interferers were simulated by using a filtered 22MHz, 100mw white noise signal. The duration of each packet is based on the Gaussian probability density function, with the mean length of a WLAN packet payload being 1024 bytes. Packet arrival is a Poisson process with the average packet rate of 500 pps.

The model uses an AWGN channel with the following ratio of signal energy and noise power spectral density:

$$E_S N_O = 13 dB$$

The value chosen for parameter α was 1.30, i.e. the chosen criterion for channel classification is:

$$PER_i > \overline{PER} * 1.30$$

This value was chosen empirically as a trade off between discarding channels to quickly and keeping too many bad channels in the hopset. The value that should be used depends on packet type and actual channels conditions, which can be seen from simulation results below.

The metrics and the simulation scenarios were selected according to the focus of our simulation. Since the focus is on the efficiency and quickness of interference avoidance while keeping the hopset as large as possible, the chosen metrics are the throughput and hopset size.

The following two simulation scenarios were created for each model variant:

- 2 FS interferers and 1, 2, 3 or 4 piconets - used to determine the performance of the proposed AFH algorithm under FS interference.
- 2 FS interferers and two piconets that are turned on at different times during the simulation - used to determine the performance of the proposed AFH algorithm under FD interference.

The first scenario starts with PFH, and at a certain point all piconets switch to the proposed AFH algorithm. Simulations with no interferers and one piconet that uses PFH were also performed for this scenario and shown on the figures, in order to determine the maximum performance of each packet type for the selected channel. The theoretical maximum for each packet type is defined in [1].

Fig.2 and Fig.3 show the performance of the proposed AFH algorithm for this scenario when DM1 packet is used, for two values of parameter MPP : 1000 and 10000 slot pairs. Since DM1 packet has error correction for both the header and payload, the performance for the selected channel with no interference is close to the theoretical maximum. FS interference occupies 44 of the available 79 channels, so there is a significant performance drop while PFH is used. It is also shown that the FD interference produced by adding piconets is relatively low. When the proposed AFH algorithm is turned on, a significant increase of performance is obtained. It can also be seen that this increase is smaller when probing is performed more often. This is due to a greater number of packets that are lost while probing. When $MPP = 10000$, only channels with low PER_i will be probed after the initial adaptation, until the end of the simulation. Hence, a lower number of packets are lost due to probing and the performance achieved for 1-piconet scenario with FS interference is very close to the 1-piconet scenario when there is no interference.

The performance of the proposed AFH algorithm when 2-DM1 packet is used is shown on Fig.4. The behavior is now similar as for DM1 packet. However, EDR packets are more susceptible to co-channel interference due to more sensitive modulation and the use of root-raised cosine filters instead of the narrower Gaussian filter. Hence, for 2-DM1 packet type the performance drop that occurs with the addition of new piconets is greater than for DM1 packet type.

The performance of the proposed AFH algorithm when 2-DH1 packet is used is shown on Fig.5. Since there is no error correction for the payload and channel conditions are bad, the relative performance fall is greater than for 2-DM1 packet. However, the absolute values are still higher than for 2-DM1.

For the first scenario we also analyzed the results for the average values of hopset size. The values are given in Table 1. The hopset size that results from the proposed AFH algorithm depends on PER_i values, and on parameters α and MPP . When PER_i values are higher and more dispersed (e.g. for 2-DH1 packet type), channels are discarded faster and replaced harder. When MPP is high, probing is seldom. Since channels are discarded in each time slot, the result is hopset size reduction. When α is low, a smaller number of channels satisfy the criterion for being in the good channel group. The results for the chosen scenario show that for DM1 packet it is possible to choose a lower α in order to achieve better distribution of channels among the piconets. It can also be seen that hopset size reduction is greatest for 2-DH1 packet type: hopset size is 20 even for 1 piconet. This is partly due to the erroneous channel classification in bad channel conditions that is inherent to using PER as the channel classification means. By using PER it is difficult to differentiate if a channel is bad due to interference which is present on a certain channel, or due to noise which is present on all the channels. Using Received Signal Strength Indicator (RSSI) as an additional channel classification means that could resolve this issue is suggested in [2].
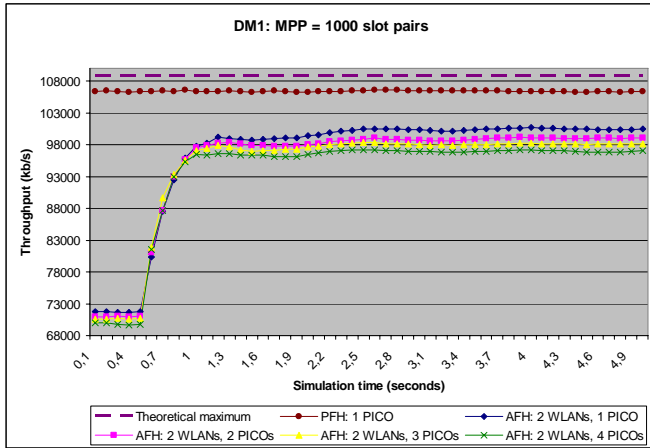
Fig.2. Scenario1: Throughput for DM1 packet and MPP = 1000
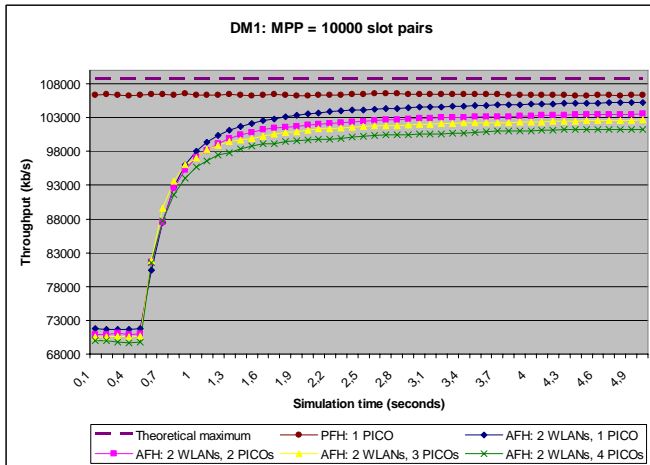

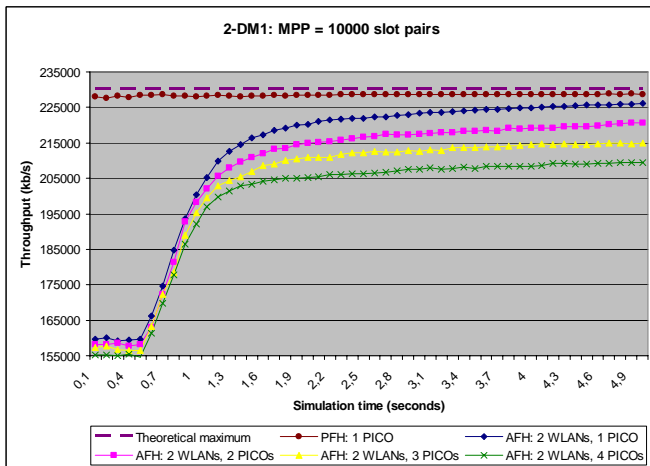Fig.3. Scenario 1: Throughput for DM1 packet and MPP = 10000

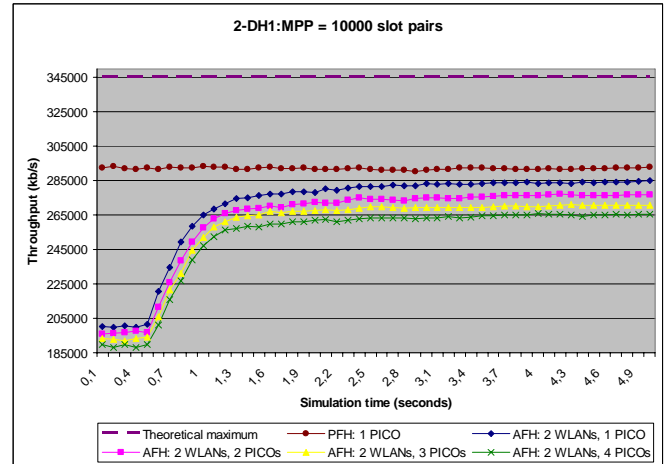Fig.4. Scenario1: Throughput for 2-DM1 packet and MPP = 10000

Fig.5. Scenario 1: Throughput for 2-DH1 packet and MPP = 10000

TABLE 1.
HOPSET SIZE

Packet Type	MPP	Piconets	Hopset Size
		1	49
	1000	2	47
		3	46
DM1		4	44
		1	45
	10000	2	34
		3	31
		4	27
		1	43
2-DM1	10000	2	22
		3	21
		4	21
		1	20
2-DH1	10000	2	20
		3	20
		4	20

Since minimum hopset size is 20 and the number of channels free from FS interference is 35, the division of these channels is difficult even for 2 piconets, regardless of the choice of values for parameters α and *MPP*. Hence, FD interference mitigation is not obvious from simulations performed for scenario 1. We use the following scenario for a better illustration of how the proposed AFH algorithm mitigates FD interference.

The scenario starts with two WLAN interferers and one piconet that uses PFH. At a certain point in the simulation, the proposed AFH algorithm is turned on for the active piconet, which then selects a hopset outside of the channel groups occupied by the FS interferers. After some time, another piconet is turned on and starts hopping using PFH. At a certain point the proposed AFH algorithm is turned on for the second piconet. It starts discarding the channels under FS interference, and using the channels that are already used by the first piconet. Both piconets continue to hop until the end of the simulation using the proposed AFH algorithm.

Simulations were performed using 3 values of *MPP* : 500, 1000 and 10000 time slot pairs. Simulation results for different *MPP* values are shown on Fig.6 to Fig.8.
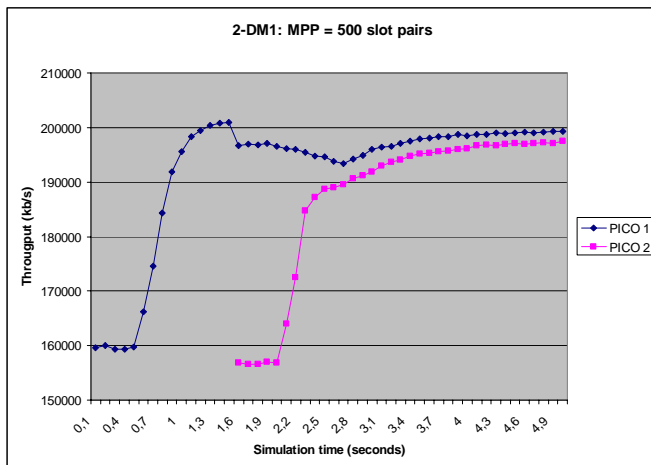
Fig.6. Scenario 2: Throughput for 2-DM1 packet and MPP = 500
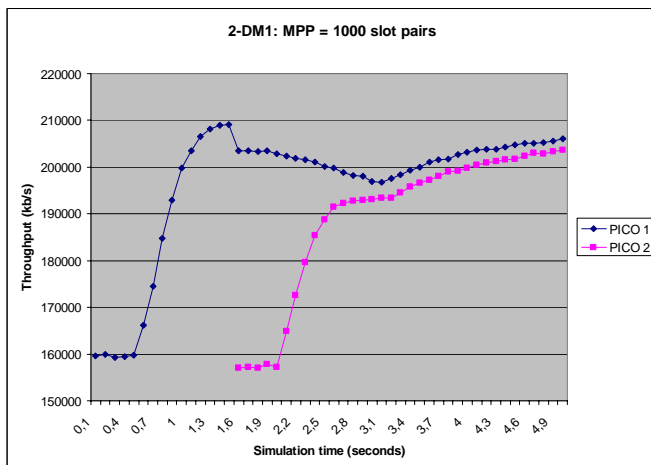

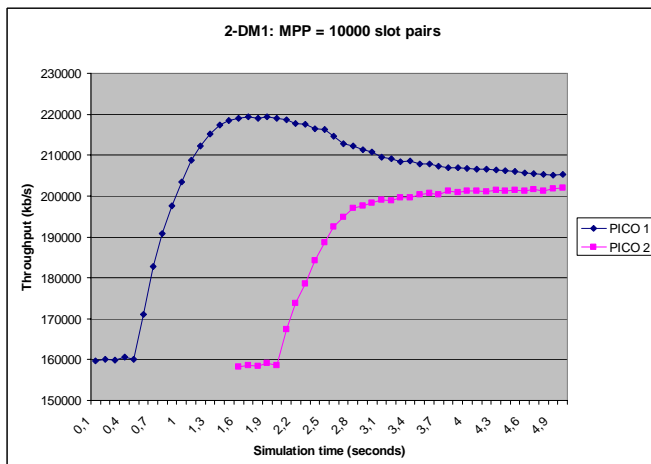Fig.7. Scenario 2: Throughput for 2-DM1 packet and MPP = 1000

Fig.8. Scenario 2: Throughput for 2-DM1 packet and MPP = 10000

It can be seen that for MPP = 500, after the proposed AFH algorithm has been turned on for both piconets, the distribution of the channels between the two piconets is quick. For MPP = 1000 the distribution of the channels between the two piconets is slower, but the throughput values are higher due to less frequent probing. For MPP = 10000, little probing is performed until the end of the simulation. Hence, each piconet hops according to the first adaptation and the

distribution of channels among the two piconets is not performed: the first piconet occupies all 35 channels that are free from FS interference, and the second piconet occupies the minimum hopset (20 channels) in this 35 channel group.

V. CONCLUSION

In this paper we have proposed a dynamic and parameterized AFH algorithm that is based on a common approach for mitigating both FS and FD interference. Using two simulation scenarios we have shown that the algorithm is effective for both types of interference. We have also shown that its performance can be controlled using the defined parameters; the trade off between the achieved throughput, the quickness of adaptation and the extent of hopset size reduction, can be regulated through these parameters.

REFERENCES

[1] Bluetooth Special Interest Group: "Specification of the Bluetooth System 2.1 + EDR, July, 2007.
[2] IEEE Std 802.15.2, "Coexistence of wireless personal area networks with other wireless devices operating in unlicensed frequency bands," Aug. 2003.
[3] FCC, "Operation within the Bands 902–928 MHz, 2400–2483.5 MHz, and 5725–5850 MHz," Pt. 15: Radio Frequency Devices, Feb. 2006.
[4] Z. Jiang, V. Leung, and V. Wong, "Reducing collisions between Bluetooth piconets by orthogonal hop set partitioning," in Proc. IEEE Radio and Wireless Conf. (RAWCON '03), August 2003.
[5] L.-J. Chen, T. Sun, and Y.-C. Chen: "Improving Bluetooth EDR data throughput using FEC and interleaving," in Proceedings of the 2nd International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '06), vol. 4325 of Lecture Notes In Computer Science, pp. 725-736, Hong Kong, December 2006.
[6] Petar Popovski, Hiroyuki Yomo, Ramjee Prasad: Strategies for adaptive frequency hopping in the unlicensed bands, Wireless Communications, IEEE [see also IEEE Personal Communications] Volume 13, Issue 6, Dec. 2006 Page(s):60 – 67.
[7] Alex C.-C. Hsu, David S.L. Wei, C.-C. Jay Kuo, N. Shiratori, Chung-Ju Chang: Enhanced Adaptive Frequency Hopping for Wireless Personal Area Networks in a Coexistence Environment, GLOBECOM 2007, 26-30 Nov. 2007, on page(s): 668-672.

Ivo Stojan is with Ericsson Nikola Tesla d.d., Croatia, from 2000. He has been working on research projects in cooperation with the Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split. He has participated in research and prototyping of products and services based on wireless local and personal area networking technologies such as marine wireless sensor networks, smart home systems, and wireless medical telemetry. He received his B.S.E.E. and M.S. degrees at the University of Split.