Frequency Rolling: A Cooperative Frequency Hopping for Mutually Interfering WPANs

Petar Popovski Hiroyuki Yomo* Sebastien Aprili Ramjee Prasad

Center for TeleInFrastructure (CTIF)
Aalborg University, Aalborg, Denmark
{petarp, yomo, saprili, prasad}@kom.auc.dk

ABSTRACT

A Wireless Personal Area Network (WPAN) provides wireless links among proximate devices, usually carried by an individual. As WPAN gains momentum in ubiquitous usage, the interference that collocated WPANs cause to each other, termed self-interference, becomes one of the major sources that degrade the communication performance of WPAN. This paper introduces the Frequency Rolling (FR), a particular instance of frequency hopping (FH) that enables the collocated WPANs to cooperate and avoid the self-interference. The FR uses as input solely the observed packet error rate (PER) and it does not require any exchange of information among the collocated WPANs. The effect of the FR over a longer time interval is that the WPANs use the complete set of disposable channels in an implicit time-division and cooperative manner. The parameters of the FR are chosen such that a WPAN which uses FR never occupies the channels in the unlicensed spectrum more than what is permitted by the current regulation. We compare the goodput offered by FR to the goodput of the collocated piconets when conventional pseudorandom FH is used. Our simulation results show that FR has superior goodput performance. In addition, the design of FR is made robust towards the errors due to the channel noise. Some guidance for practical fault-tolerant design and future extensions of FR are given. All these features promote the great potential of FR as a coexistence mechanism for unlicensed operation.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communications; C.2.2 [**Network Protocols**]

General Terms

Algorithm, Design, Performance

Keywords

Wireless Personal Area Network (WPAN), Frequency Hopping, Coexistence, Unlicensed spectrum

1. INTRODUCTION

The short-range wireless technologies are focused on providing ad-hoc wireless links among proximate devices. A Wireless Personal Area Network (WPAN) supports connectivity among the devices carried by an individual and provides the user with links to other proximate devices. WPAN is an ad hoc network, established and maintained solely by the self-organizing actions of the participating devices. Bluetooth [1] is a first instance of the WPAN concept and it has been a basis for standardization within the IEEE 802.15 Working Group (WG) [2].

The WPAN technology is aspired to a ubiquitous usage and is therefore set to operate in unlicensed spectrum. Here we consider Bluetooth WPANs that use frequency hopping (FH) over 79 channels in the unlicensed 2.4 GHz ISM band. The unlicensed spectrum is not allocated for exclusive usage and its utilization is regulated [3] by rules that limit the radiated power and set constraints to the occupancy of the individual channels used in hopping. A WPAN interferes with proximate WPANs or other proximate non-WPAN entities that utilize the same spectrum. For example, a serious throughput degradation can occur when a WPAN is settled close to IEEE 802.11 Wireless Local Area Network (WLAN) [4] or microwave oven. As the WPAN proliferation gains momentum, many WPANs will occur to be in close proximity at the spots with high user density, such as airports, conference rooms, etc. Such collocated WPANs cause severe interference to each other [5] [6] [7] [8]. The mutual interference among collocated WPANs is termed *self-interference* and is the main topic of this paper.

The interference experienced by WPAN is itself of ad hoc nature, since it has unpredictable and ephemeral character. Being ad hoc network, the WPAN can not assume existence of infrastructure that manages the spectrum access for the collocated entities. Hence, the WPAN should adapt its spectrum usage to the present interference pattern by applying certain *coexistence mechanism.* One such mechanism is the *adaptive frequency hopping* [9] by which the WPANs attempt to hop over the channels that are likely to provide correct packet reception. Decidedly, the adaptive FH should conform to the regulation rules. The difficulty in adaptive FH lies in knowing which group of channels is less likely to be interfered during a packet transmission. From the per-

---

[*] Hiroyuki Yomo is working for Future Adaptive Communication Environment (FACE) project.

MobiHoc'04, May 24–26, 2004, Roppongi, Japan.

spective of adaptive hopping, we differentiate among three error sources for a WPAN:

1. **Noise** - It is uncorrelated in time and yields uniform error over the whole set of frequencies. As such, the PER cannot be decreased by employing some intelligent FH pattern.

2. **Frequency-static interference** - Used to denote an interference that occurs at group of channels for a time that is considerably longer than the packet duration. In this case the knowledge about the channel is based on the probable correlation in time: If there is higher PER at a group of channels for some period, then it is likely that high PER will be present in the next period. Straightforwardly, the WPAN should remove a bad channel from the hopping set for a certain time. The WLAN is a canonical example of frequency-static interferer for WPAN [10, 11, 9]. The effect of frequency-static interferer can also be produced if there is a frequency-selective fading at some group of channels for a longer time.

3. **Frequency-dynamic self-interference** - The disturbance that a collocated WPAN causes at certain channel occurs for a time approximately equal to the packet duration. The above strategy of removing the bad channel for certain time is not applicable: If a WPAN removes a channel where error has been experienced, it is likely that collocated WPAN that caused that error also removes the same channel. This worsens the throughput, since the WPANs have still identical, but smaller hopsets. However, the collocated WPANs are identical entities and we design a hopping method that enables *cooperation* among the collocated WPANs with the objective of avoiding the self-interference.

It is very important to note that what we denote as cooperative hopping *is not related* to the *collaborative techniques* for coexistence. In collaborative techniques, the interfering entities explicitly exchange data to achieve mutual coordination. On the other hand, in cooperative hopping there is no such exchange and the cooperation is *implicit*: A WPAN knows that the packet transmitted at time t_1 and frequency i will, most likely, not be interfered by the collocated WPANs; vice versa, that WPAN offers some assurance to the other WPANs not to use some group of channels at time t_1. Such method of cooperative hopping is the Dynamic Adaptive Frequency Hopping (DAFH), proposed in [12]. DAFH succeeds in avoiding the self-interference and inherently avoids frequency static interference, while making a best effort to keep the occupancy of each channel as low as possible. However, the channel occupancy produced by the DAFH is not in strict agreement with the current regulation for the FH systems, although the authors justify such situation by proposing new quantitative etiquette rule for the unlicensed operation.

In this paper we propose the *Frequency Rolling (FR)*, a method for cooperative adaptive FH which avoids the self-interference among collocated WPANs. The proposal of FR is founded on the fresh look into the current regulation rules for the FH systems in the unlicensed ISM band. In a small time interval, called *rolling interval*, each WPAN picks pseudorandomly a channel from a hopset that contains only a subset of all 79 channels. The hopset is continuously changed, such that observed over a long interval, the WPAN hops over all 79 channels. In each rolling interval the WPAN monitors the PER and if it is larger than a threshold, the WPAN chooses another hopset randomly. This randomization yields a possibility to orthogonalize the hopsets of the collocated WPANs. In fact, the FR achieves that the collocated WPANs use the same hopsets by implicit time-division. The parameters are chosen such that a WPAN that utilizes FR never causes higher occupancy of the channels than what is permitted by the current regulation. Our results show that FR successfully avoids the self-interference and yields high goodput gain. The improvements introduced by FR are valid even when the collocation pattern of the WPANs is subject to dynamic changes.

The paper is organized as follows. Section 2 describes the used system model. Section 3 considers the relevant aspects of the regulations for operation in unlicensed spectrum. Section 4 describes the main concept of the FR as well as the details of the FR parameter design. This is followed by numerical results in Section 5 and discussions in Section 6, and finally the last section concludes the paper.

2. SYSTEM MODEL

The focus in this work is put on the self-interference among the collocated WPANs and we will abstract the internal structure of a WPAN. The abstraction is derived from the actual networking structure in Bluetooth (IEEE 802.15.1) and we briefly outline some facts pertaining to this technology. A basic networking entity in Bluetooth is *piconet*—a star topology with a master and up to seven active slaves. The communication channel in a piconet is slotted with a nominal slot value 625 [μs]. The hop selection in each slot is based on a pseudorandom generator determined by the master. We denote the total set of disposable channels as $C_M = \{0, 1, \ldots, M-1\}$ and $M = 79$. The slaves are time- and hop-synchronized to the master. The master uses polling to schedule the packet transmissions, such that, within the piconet, the communication is collision-free and at each slot only one device transmits.

A WPAN corresponds to a piconet and we will use the terms WPAN and piconet interchangeably. The piconet k is denoted by π_k. In order to develop a precise notion of "collocated", consider two piconets π_1 and π_2. The distance between devices within a piconet is small and the following approximation can be made: If *at least one* device in π_1 is in transmission range of a device in π_2, then *each* device in π_1 is in transmission range of *any* device in π_2, and vice versa. This allows us to consistently define that two piconets π_1 and π_2 are *collocated* if the devices in the piconets are in each other's transmission range. Therefore, a receiving device in piconet π_1 perceives the collocated piconet π_2 as a transmitter with slotted channel. For simplicity, here we assume that each used packet has duration of a single slot, though discussions can be easily extended to the case with multiple-slot packet. The piconets are asynchronous in a sense that their slot-starts are not coinciding (see Fig. 1), making a slot in a piconet overlap with two slots in another piconet. We model the piconet π_k as a transmitter for which the probability that packet occurs in a slot is equal to the traffic load G_k. Such modelling has been commonly used to analyze the coexistence issues in Bluetooth [5] [13] [14].

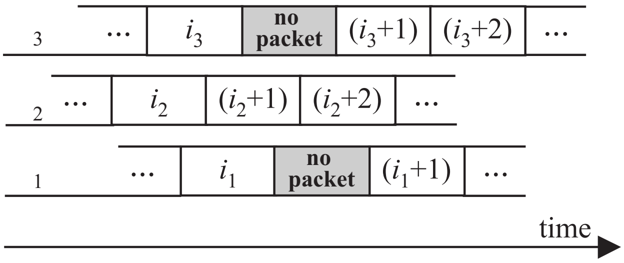

Figure 1: Example of packet overlapping in the case of asynchronous piconets.

In this paper, we consider that piconets are fully loaded ($G_k = 1$)[7] [8].

Piconet is applying per-packet frequency hopping: A frequency is selected at the slot-start and remains constant during the packet transmission. Two collocated piconets interfere with each other if they simultaneously transmit packet at the same frequency. For example, assume that π_1 and π_2 from Fig. 1 are collocated. Then, if packet i_1 in π_1 is transmitted on a same frequency as packet i_2 in π_2, the intended receivers from piconets π_1 and π_2 will experience collision. In this paper, a collision is always considered *destructive*, resulting in a packet error with probability one [5] [6]. IEEE 802.15.2 considers monitoring of the packet errors as a primary method to assess the channel quality, due to its simplicity. Our algorithm also uses the packet error measurements as an input. We do not assume any special mechanism for collision detection—the collided packet is detected to be erroneous by an error detection code. Therefore, the receiver *cannot distinguish* between the error due to collision with other piconet and error due to other channel impairments. We consider a scenario where the errors are induced by collisions due to self-interference and errors due to noise, while we assume absence of frequency-static interferer.

3. REGULATION ISSUES FOR FREQUENCY HOPPING IN UNLICENSED SPECTRUM

A crucial point in our proposal is how to design the cooperative FH by staying in agreement with the regulation rules for the FH systems that operate in the unlicensed spectrum [3]. In its essence, the regulation policy imposes dynamic usage of all frequency channels allocated in the unlicensed band. One of the main goals of such policy is to prevent unfair opportunistic behavior and we make a slight digression to explain this issue.

Define a *reference interfered entity (RIE)* to be wireless entity that uses spectrum B_R and denote by B_P the spectrum of WPAN. B_P is divided into M hopping channels and M_R out of M channels fall into the spectrum B_R. Observe the possible asymmetric situation: RIE uses low power and it cannot interfere the communication in a collocated WPAN, while the communication of RIE is degraded whenever the WPAN hops on one of the M_R channels in B_R. We can, roughly, say that the probability that RIE is interfered by WPAN is $\frac{M_R}{M}$. Now assume WPAN experiences excessive errors when hopping on some channels that are in B_P, but are not belonging to B_R (shortly written as $B_P \setminus B_R$). If the WPAN adapts its hopping to avoid these errors, then it suppresses the usage of some channels in $B_P \setminus B_R$ and hops over only $M_1 < M$ channels. The M_R channels in B_R are retained into the set of M_1 channels. Then, the probability that RIE is interfered by WPAN is $\frac{M_R}{M_1} > \frac{M_R}{M}$. Hence, the WPAN improved its throughput via adapting its hopset, but at the expense of degraded communication in RIE. Let $t_i(T_0)$ denote the occupancy of $i-$th channel in interval T_0, defined as the time within T_0 for which the WPAN uses the $i-$th channel. The communication in RIE is affected by the occupancy that the WPAN causes for the channels from B_R. Therefore, the regulation for unlicensed operation puts upper limits to the occupancy of the individual channels that should be preserved by the FH systems.

We have stated that the proposed frequency rolling (FR) is a method of frequency hopping which enables the collocated WPANs to cooperate in avoiding the mutual interference. In achieving this, the FR has to ensure that the WPAN does not occupy the channels more than permitted by the current regulation. Hence, the starting point in the design of FR is the current regulation [3] for the FH systems in the unlicensed ISM band:

(S1) *Frequency hopping systems in the* $2400-2483.5$ *MHz band shall use at least 15 non-overlapping channels. The average time of occupancy on any channel shall not be greater than 0.4 seconds within a period of 0.4 seconds multiplied by the number of hopping channels employed. Frequency hopping systems which use fewer than 75 hopping frequencies may employ intelligent hopping techniques to avoid interference to other transmissions. Frequency hopping systems may avoid or suppress transmissions on a particular hopping frequency provided that a minimum of 15 non-overlapping channels are used.*

A straightforward way to be in agreement with **S1** is the pseudorandom FH over all allocated channels: For each packet, the frequency is picked out of the 79 channels in pseudorandom fashion. This choice exhibits dynamic channel usage, but there is a graceful degradation of the throughput as the number of collocated self-interfering piconets grows. Alternatively, **S1** allows that each WPAN hops pseudorandomly over a set of at least 15 channels. This permits to achieve a situation in which the collocated WPANs have non-intersecting hopsets and thereby avoid the self-interference. However, this is possible only if the number of collocated WPANs is $N < \lfloor\frac{79}{15}\rfloor = 5$. The goal of FR is to accommodate more than 5 non-interfering WPANs, while keeping the channel occupancy lower or equal to what is permitted by **S1**.

The design of FR embeds the following statement:

(S2) *A frequency hopping system should use the disposable channels in such way that, in* ***any interval*** *of 6 seconds, no channel is used for more than 0.4 seconds.*

We show intuitively that a system that does the frequency hopping conforming to **S2** cannot create higher channel occupancy than permitted by **S1**. The highest occupancy of individual channels in **S1** occurs when the number of used channels is minimal, i.e. 15. The regulation **S1** mandates that the occupancy of each of the 15 channels is 0.4 [sec] in an interval of 6 [sec], while in a longer interval $T_1 > 6$ [sec] the expected occupancy of each channel is $t_i(T_1) = \frac{T_1}{15}$. However, **S2** allows usage of more than 15 channels within T_1, while it mandates that each channel is used not more

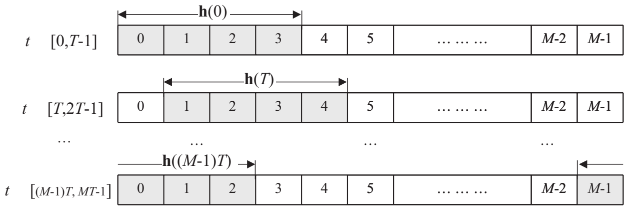

Figure 2: Illustration of the usage of individual channels during rolling with hopset size $H = 4$.

than 0.4 [sec] in each 6-second subinterval of T_1. Consequently, the occupancy of each channel in T_1 is less or equal to $\frac{T_1}{15}$. This claim is proved in the Appendix.

4. FREQUENCY ROLLING

4.1 Basics of Frequency Rolling

In this section we first introduce the main ideas and notions used in the FR. We will use the term *hopset* of piconet π_k to denote a set of H adjacent channels (frequencies) from the total set of channels C_M. The total number of different hopsets is M. For each slot, π_k chooses pseudorandomly and uniformly a channel from its current hopset. Let $\mathbf{h}_k(t_0)$ denote the hopset that π_k starts to use in slot t_0. If $\mathbf{h}_k(t_0)$ is represented by the following H adjacent channels:

$$\{m, (m+1), \ldots, (m + (H-1))\}$$

where the additions are modulo M, then we write

$$\mathbf{h}_k(t_0) = [f_k^s(t_0), f_k^e(t_0)] \tag{1}$$

where $f_k^s(t_0) = m$ (starting channel of the hopset) and $f_k^e(t_0) = (m+(H-1)) \text{mod } M$ (ending channel of the hopset). In *nominal* frequency rolling, π_k changes the hopset $\mathbf{h}_k(t_0)$ after a *rolling period* of T slots, by shifting the hopset one channel to the right. That is, for each slot $t \in [t_0, t_0 + T)$ [1], the hopset of π_k is $\mathbf{h}_k(t_0)$, while at slot $(t_0 + T)$, the hopset becomes:

$$\mathbf{h}_k(t_0 + T) = [f_k^s(t_0 + T), f_k^e(t_0 + T)]$$

where

$$\begin{aligned} f_k^s(t_0 + T) &= (f_k^s(t_0) + 1) \text{mod } M \\ f_k^e(t_0 + T) &= (f_k^e(t_0) + 1) \text{mod } M \end{aligned} \tag{2}$$

From the representation on Fig. 2, the hopset is *rolling* in time over the set of all frequency channels, with speed of 1 frequency per T slots.

All piconet members must possess information which frequency is used in each slot t. Analogously to Bluetooth, we can assume that the piconet master determines a *Generating Sequence*, denoted $GS(t)$, which is a sequence of pseudorandom numbers from the set $[0, H-1]$. The master shares $GS(t)$ with all piconet slaves. The channel used in slot t is obtained as:

$$f(t) = (GS(t) + GO) \text{ mod } M, \tag{3}$$

where $GO \in C_M$ is the *Generating Offset*, a value that is also shared by all piconet members. The initial GO is conveyed by the master. After a rolling period expires, each piconet member updates:

$$GO = (GO + 1) \text{ mod } M \tag{4}$$

We need to find what are the admissible values for the rolling period T and the hopset size H, such that a piconet follows **S2** under a nominal rolling. We set $H_{\min} = 2$ due to the following reasons. The discussion above implies that, once GO and $GS(t)$ are initialized in the piconet and if nominal rolling is applied, there is no need to convey any update information on hopping within the piconet. However, as it will be seen further, a piconet occasionally interrupts the nominal rolling by choosing another hopset before the current rolling period expires. In such case, the piconet master must convey to the slave the correct information about the GO ($GS(t)$ is not affected by this action) and the instant at which that GO is updated. On the other hand, if $H = 1$ and two collocated piconets are using the same channel, no information can be sent within each of the piconets unless using some ALOHA-like mechanism to resolve the contention. If $H > 1$ the pseudorandom frequency selection inherently resolves the contention. The maximal considered value is $H_{\max} = 13$, by which we can accommodate $\lfloor \frac{79}{13} \rfloor = 6$ collocated piconets. Note that if the target is to accommodate e.g. 5 piconets, there are more straightforward methods implied by **S1**. For example, we can define in advance 5 non-intersecting hopsets: 4 hopsets, each with 15 frequencies, and one hopset with 19 frequencies. Then, when a piconet detects excessive PER during the utilization of a hopset, it attempts to pick randomly another, non-interfered, hopset from the other 4. However, if there are more than 5 collocated piconets, this strategy could lead to severe throughput degradation.

Further, we require maximization of the rolling period T due to two reasons. First, the PER estimation, needed as an input to the FR, is more reliable (see Section 4.4). Second, if T is shorter, it becomes more difficult to reliably disseminate the information about hopset change (see Section 4.5).

We now proceed to find maximal rolling period T, provided that $2 \leq H \leq 13$. With slot duration $t_S = 625$ [μsec] and a slight abuse of the notation, we can write T [slots]$= T \cdot t_S$ [sec]. Let π_k employ the hopset

$$\mathbf{h}_k(0) = [0, H-1] \tag{5}$$

at slot 0, as in Fig. 2. Due to the pseudorandom selection from the hopset, the occupancy of each $m \in \mathbf{h}(t)$ for interval $T_1 = [t, t+T)$ is $t_m(T_1) = \frac{T}{H}$. The frequency rolling is done over the whole set of $M = 79$ channels. We first take $T = 0.4$ [sec]$=$ 640 [slots]. The special case of nominal rolling with $H = 3$ is depicted on Fig. 3. At most 16 different hopsets can occur during any observation interval $T_0 = 6$ [sec]. Since in our case $H < 16$ and $M > 16$, a single channel can occur in at most H different hopsets in any T_0. Hence, for nominal rolling, the maximal expected channel occupancy for any $m \in C_M$ in any T_0 is $t_m(T_0) = H \cdot \frac{T}{H} = T = 0.4$ [sec], which is compatible with **S2**. Note that this holds for *any* $H < 16$ when $T = 0.4$ [sec].

Let now be $T > 0.4$ [sec], i.e. $T = 0.4\alpha$ with $\alpha > 1$. In any observation interval $T_0 = 6$ [sec] there can be at most

$$H^{\alpha} = \left\lceil \frac{T_0}{T} \right\rceil = \left\lceil \frac{15}{\alpha} \right\rceil \tag{6}$$

different hopsets occurring. For each T_0, the same channel

---

[1] Since t takes only integer values, with this notation we have $[t_0, t_0 + T) = [t_0, t_0 + T - 1]$

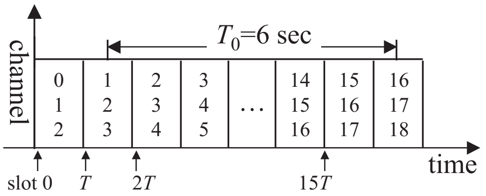

Figure 3: The channel occupancy in observation interval $T_0 = 6$ [sec] with rolling period $T = 0.4$ [sec] and hopset size $H = 4$.

can be a member of at most $\min(H^\alpha, H)$ hopsets. Hence, in this case we have a lower bound on H, which is $H > H^\alpha$. If $H \leq H^\alpha$, then $\min(H^\alpha, H) = H$, which violates **S2**, since the obtained average channel occupancy would be $T > 0.4$ [sec] within 6 [sec]. Let $H > H^\alpha$. Then, for some m and T_0, the occupancy is:

$$t_m(T_0) = H^\alpha \cdot \frac{T}{H} = \left\lceil \frac{15}{\alpha} \right\rceil \cdot \frac{0.4\alpha}{H} \leq 0.4 \text{ [sec]} \qquad (7)$$

and it follows that

$$\left\lceil \frac{15}{\alpha} \right\rceil \leq \frac{H}{\alpha} \qquad (8)$$

which yields $H \geq 15$. Since we are interested in $2 \leq H \leq 13$, in the sequel we will fix the value of the rolling period to be $T = 0.4$ [sec].

4.2 The Random Jump Mechanism

We now proceed to describe the case when the nominal rolling is interrupted if the WPAN detects excessive PER. The FR algorithm is in fact executed only by the piconet master, but we assume that the master can detect any erroneous packet received within a piconet and thereby assess the PER calculated for all packets sent in the piconet. Indeed, every packet exchange involves the master and, if the packet is sent from the master to the slave, the master detects error by receiving NACK or implicitly, by not receiving answer from the slave. The master of piconet π_k, denoted μ_k, has a counter of erroneous packets, denoted C_k. When a new hopset $\mathbf{h}_k(t_0)$ is used, μ_k resets $C_k = 0$ at slot t_0. Let τ_k denote the *triggering threshold* of μ_k. Then μ_k is *triggered* if it detects $C_k = \tau_k$ in some slot t_1, where $t_0 < t_1 < t_0 + T$. After being triggered, μ_k decides to prematurely change the hopset at slot t_2, where $t_1 < t_2 < t_0 + T$. The new hopset is $\mathbf{h}_k(t_2) = [f_k^s(t_2), f_k^e(t_2)]$, where $f_k^s(t_2)$ is determined by a *random jump* J_k as:

$$f_k^s(t_2) = (f_k^s(t_0) + J_k) \mod M \qquad (9)$$

The random jump is a pseudorandom integer $J_k > 1$ generated by the master upon triggering (the lower and upper bound on J_k will be determined subsequently). The value J_k changes the generation offset in (3). Therefore, all slaves in π_k should receive the values J_k and t_2 from μ_k, in order to be able to correctly estimate the frequency used in π_k for any $t > t_2$. In slot t_2, all devices in the piconet set:

$$GO = (GO + J_k) \mod M \qquad (10)$$

Let the piconet π_l become collocated to π_k and let their hopsets be denoted by $\mathbf{h}_k(t_0^k)$ and $\mathbf{h}_l(t_0^l)$, respectively. Since $\mathbf{h}_k(t_0^k) \cap \mathbf{h}_l(t_0^l) \neq \emptyset$, the piconets experience self-interference. The two key FR parameters, H and T, are identical for all piconets. The triggering threshold τ_k and τ_l are not necessarily equal and let us assume that $\tau_k < \tau_l$. For the sake of clarity, let π_k and π_l be slot-synchronous, but not synchronized in the rolling, i.e. the nominal change of the hopset does not occur at the same slot for both piconets. Having $\tau_k < \tau_l$, the master μ_k is triggered first at some slot $t_1^k > t_0^k$ and changes to new hopset at slot $t_2^k > t_1^k$ by using a jump J_k. Assume that J_k is picked such that $f_k^s(t_2^k) > f_l^e(t_0^l) + 1$. Although not explicitly mentioned, it is understood that the comparison of the frequencies is modulo M (see Fig. 2). This implies $\mathbf{h}_k(t_2^k) \cap \mathbf{h}_l(t_0^l) = \emptyset$. Since the piconets are rolling at the same pace, the piconet π_l can never "reach" the piconet π_k. Therefore π_l and π_k will completely avoid the self-interference. Note that, if $f_k^s(t_2^k) = f_l^e(t_0^l) + 1$, the self-interference is not avoided: the piconet π_l is first to change its hopset at $t_0^l + T$, which results in

$$\mathbf{h}_k(t_2^k + 1) \cap \mathbf{h}_l(t_0^l + T) = f_k^s(t_2^k) \qquad (11)$$

4.3 Bounds for the Random Jump

The master μ_k picks the random jump J_k uniformly from an interval $[J_{\min}, J_{\max}]$. This subsection considers the choice of values $J_{\min}$ and $J_{\max}$.

$J_{\min}$ is determined such as to avoid the overlapping between the hopsets of two collocated piconets. Consider two collocated π_k and π_l and let $\tau_k < \tau_l$, so that μ_k is triggered first. The minimal value that guarantees avoidance of self-interference can be derived if we consider the case when

$$f_k^e(t_0^k) = f_l^s(t_0^l) \Leftrightarrow f_k^s(t_0^k) = f_l^s(t_0^l) - H + 1 \qquad (12)$$

Let μ_k be triggered in t_1^k and let π_k employ the new hopset in $t_2^k > t_1^k$. Since we require that $f_k^s(t_2^k) > f_l^e(t_0^l)+1$, then we should have $J_{\min} = 2H$. However, if in the same overlapping situation, the piconet π_l is triggered first, then the value $J_{\min} = 2$ would be enough. The intention to keep $J_{\min}$ as small as possible is motivated by the fact that, when there are more than 2 collocated piconets, it is beneficial to have more possible jump values available, so that the hopsets of the piconets can be densely packed over the set of disposable channels. As a compromise, we adopt the average value $J_{\min} = H + 1$.

The value $J_{\max}$ is determined such as to ensure that π_k stays in agreement with **S2**. Therefore, the FR imposes the following condition to the piconet:

(S3) *After a random jump, the piconet π_k continues with a nominal rolling, without random jump, for at least 6 seconds. This corresponds to 15 rolling periods with $T = 0.4$* [sec].

We now state and proof the following theorem:

THEOREM 1. *If the frequency rolling includes the condition **S3** and $J_{\max} = M - 15$, then statement **S2** is satisfied.*

PROOF. Without loss of generality, let us assume that $H = 3$. As in Fig. 4, let π_k have $f_k^s(0) = 0$, such that it uses the hopset $\mathbf{h}_k(0) = [0, 2]$ in the rolling interval $[0, T)$. Let π_k apply the new hopset, obtained by a random jump, in slot $t_1 = 15T + T_1$ where $T_1 < T$. From **S3** it follows that the rolling is nominal at least 6 [sec] before and at least 6 [sec] after the jump. Hence, there is no random jump both in any $[iT, (i+1)T)$ and any $[t_1 + iT, t_1 + (i+1)T)$, where $i = 0, 1, \ldots 14$.

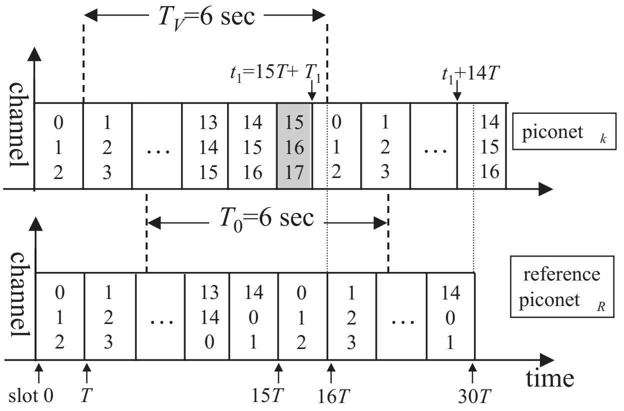

Figure 4: Comparison of the channel occupancy in the observed piconet π_k and a reference piconet π_R which uses only 15 channels. The condition (S3) is satisfied for each interval T_0.

We first show that if FR applies **S3**, but $J_{\max} > M - 15$, then the WPAN violates **S2**. Let π_k apply maximal jump at t_1. If $J_{\max} = M - 14$, then the hopset applied in t_1 would be $\mathbf{h}_k(t_1) = [1, 3]$ instead of $[0, 2]$, which is depicted on Fig. 4. Consider what would be the expected occupancy of channel 3 within the 6-second interval T_V that starts at slot T:

$$3 \cdot \frac{T}{3} + \frac{T - T_1}{3} = T + \frac{T - T_1}{3} > 0.4 \text{ [sec]} \tag{13}$$

which violates **S2**.

Now we set $J_{\max} = M - 15$ and let π_R be a reference piconet that applies frequency rolling, but its complete channel set contains only 15 frequencies $\{0, 1, 2, \ldots 14\}$. The piconet π_R also starts to use the set $[0, 2]$ at slot 0, it always does nominal rolling, without any random jump (see Fig. 4) and π_R satisfies **S2**. Let π_k make a maximal jump at t_1, such that $\mathbf{h}_k(t_1) = [0, 2]$. Consider arbitrary 6-second interval T_0 that starts in slot $t_2, 0 < t_2 < t_1$. The usage of each of the channels $15, 16, 17$ in π_k is less than 0.4 seconds in T_0. Observe the usage of any channel $i \in \{0, 1, 2, \ldots 14\}$ within any T_0 for both π_k and π_R. No channel i can be occupied more in π_k than in π_R, since π_k uses additional channels in T_0 as compared to π_R. Therefore, π_k is in agreement with **S2** if $J_{\max} = M - 15$.

If the jump at t_1 is $J < J_{\max}$, then, using again Fig. 4, we can see that a occupancy of each channel in π_k decreases because π_k uses more different channels within T_0. Finally, due to **S3**, the next random jump in π_k can be applied after the slot $t_1 + 15T$. This means that **S2** is satisfied for T_0 that starts in slot t_1, while we can apply the previous considerations for the intervals T_0 that start later than t_1 if there is random jump in $t_1 + 15T$. □

4.4 Choice of the Triggering Threshold

If we are certain that a packet error in piconet π_k can be caused only by self-interference with a collocated piconet, then a single packet error implies existence of a collocated piconet and the master μ_k should set $\tau_k = 1$. However, if there are other error sources, primarily noise, then the FR does not help to avoid the interference, while the value $\tau_k = 1$ causes false triggering. The false triggering only increases the overhead that is generated when the master conveys the hopset information to the slaves. A master should not be triggered in the absence of a collocated piconet and therefore the threshold should be chosen to be above the noise level. Let $\hat{p}_n$ be the estimated maximal probability of packet error induced by channel noise. If π_k is fully loaded, then T packets are sent during the rolling period of T slots. To be above the noise level, the threshold should be chosen:

$$\tau_{\min} = \lceil \hat{p}_n \cdot T \rceil \tag{14}$$

such that it is likely that π_k keeps on with a nominal rolling when the noise is the single source of error. For example, if $\hat{p}_n = 0.01$, then $\tau_{\min} = 7 > 0.01 \cdot 640$. Recall that we have chosen the rolling interval T to have the maximal possible value. The higher value of T implies higher value of $\tau_{\min}$, which further implies more reliable assessment of the PER by the master and thus reduction of the noise-induced triggerings.

If π_k is not fully loaded and has traffic load $G_k < 1$, then the threshold choice (14) implies that μ_k can tolerate PER that is additionally higher than the minimal admissible value due to the noise. To make it equivalent with the fully loaded case, the threshold should be chosen:

$$\tau_{\min} = \lceil \hat{p}_n \cdot G_k \cdot T \rceil \tag{15}$$

Since we need to have the master triggered when there is collocated piconet, we briefly investigate the PER when there is a single collocated piconet. Let us neglect the noise and let the collocated piconets π_k and π_l use m common channels. The probability that a transmission in π_k is not interfered by π_l when π_k uses a channel from the common channels is:

$$\left(1 - \frac{G_l}{H}\right)^2 \tag{16}$$

since each packet in π_k overlaps with two packets in π_l (Fig. 1). The value G_l is the traffic load of π_l. The probability of successful transmission in π_k is:

$$p_s(m) = \frac{H - m}{H} + \frac{m}{H}\left(1 - \frac{G_l}{H}\right)^2 \tag{17}$$

since the packet in π_k is always transmitted successfully whenever a channel that is not common with π_l is chosen. Finally, the probability of packet error in π_k when π_l is collocated and has m common channels with π_k is:

$$p_\epsilon(m) = 1 - p_s(m) = \frac{2mG_l}{H^2} - \frac{mG_l^2}{H^3} \tag{18}$$

The interference to π_k is maximized if π_l is fully loaded $G_l = 1$. Fixing $G_l = 1$, the minimal value in (18) occurs when H is maximized and π_k and π_l have only one common channel $m = 1$. If $H = 13$, the probability of packet error is 0.0114. If in this case the estimated noise has $\hat{p}_n > 0.0114$, then the FR would have high threshold and therefore allow two collocated piconets to interfere at the single common channel.

Although the goal of each piconet is to minimize the tolerable PER, the master μ_k does not always set $\tau_k = \tau_{\min}$. In fact, the value τ_k is chosen uniformly from an interval $[\tau_{\min}, \tau_{\max}]$, where $\frac{\tau_{\max}}{T}$ represents the maximal tolerable PER for a fully loaded piconet. This choice is made to break the symmetry in the observation of the PER when two collocated piconets have overlapping hopsets. As explained in the previous section, if the thresholds are different, the master in one of the piconets will be triggered before the other.

4.5 Intra-Piconet Dissemination of Hopset Information

Let π_k use the hopset $\mathbf{h}_k(t_0)$ and let its master μ_k be triggered at the slot $t_1 < t_0 + T$. Then μ_k generates the jump J_k and estimates the slot $t_2 > t_1$ in which the new generation offset should be applied. The time slot t_2 is estimated such that the master can broadcast the information with some predefined level of reliability within $(t_2 - t_1)$ slots. If estimated $t_2 > t_0 + T$, then μ_k does not initiate a hopset change and π_k continues with nominal rolling. The master sends the information about J_k and t_2 to the slaves by broadcasting. μ_k calculates the needed number n_B of broadcast packets. In Bluetooth, the master can broadcast only in even-numbered slots, such that for transmission of n_B packets μ_k uses $2n_B$ slots. Hence, we can write that if

$$t_1 + 2n_B > t_0 + T \tag{19}$$

then the master cannot transmit the planned number of broadcast packets and the piconet continues with the nominal frequency rolling.

The choice of n_B is related to the desired reliability of the broadcast. Let ρ [%] denote the target broadcast reliability and let p_B denote the probability that a broadcast packet is not received correctly by the piconet members. If the packet is broadcasted n_B times, then the probability that all broadcasts are received incorrectly is $p_B^{n_B}$. The probability that at least one broadcast packet has been received correctly should be higher than the desired reliability:

$$1 - p_B^{n_B} \geq \frac{\rho}{100} \tag{20}$$

which results in:

$$n_B = \left\lceil \frac{\log\left(1 - \frac{\rho}{100}\right)}{\log p_B} \right\rceil \tag{21}$$

The key element of (21) is to determine suitable value of the probability p_B. The master μ_k sets p_B to be equal to the expected PER, estimated when the piconet is triggered. We adopt the following estimation of the expected PER. Let π_k use the hopset $\mathbf{h}_k(t_0)$ in $[t_0, t_0 + T)$ and let the first packet error occur at slot $t_0 < t'_0 < t_0 + T$. Then μ_k sets a slot counter $s_k = 0$ and s_k is incremented in each slot until μ_k is triggered in slot $t_1 > t'_0$. In slot t_1 we have $s_k = t_1 - t'_0$, the number of errors within those s_k slots is $\tau_k - 1$, such that μ_k sets the estimated PER to be:

$$\hat{p}_B = \frac{\tau_k - 1}{s_k} \tag{22}$$

We have chosen the estimation (22) rather than using the expression:

$$\frac{\tau_k}{t_1 - t_0} \tag{23}$$

This is because (23) would give optimistic estimation of the PER when π_l is not interfering with π_k in t_0, but it starts at some instant after t_0. The pessimistic estimate is needed in order to better approximate the target reliability.

5. NUMERICAL RESULTS

5.1 Simulation Model

We consider a quasi-static scenario where piconets (WPAN users) arrive at a hotspot (e.g., lounge at airport, conference room, etc), and leave the hotspot after a random dwell time. We assume a Poisson process with generation rate λ for the piconet arrivals and exponential distribution with the average of $\bar{T}$ [sec] for the dwell time at the hotspot. Since our target scenario is a quasi-static environment, we set the minimal dwell time to be 20 [sec](see [15]), which makes the exponential distribution shifted for 20 [sec]. In this case, the average number of piconets at the hotspot, denoted $\bar{N}$, can be obtained from Little's formula as [16]:

$$\bar{N} = \lambda \times (\bar{T} + 20). \tag{24}$$

In the simulation, we fix $\bar{T}$ at 60 [sec], and change the average number of piconets by setting different value of λ. We intentionally choose the values for the dwell time to be short in order to create dynamic conditions which tend to create the worst-case scenarios for the FR performance. If the dwell time values are larger, the same set of piconets stays collocated for longer time. Then the piconets have enough time to separate their hopsets and operate for a considerable period without self-interference and triggering, which certainly improves the overall performance. We set the total number of frequencies $M = 79$ as in Bluetooth. In order to set the minimum threshold for FR, we have to know the maximal estimated PER due to noise errors in a single piconet, $\hat{p}_n$. We set this value to be $\hat{p}_n = 1$[%] [7], resulting in the minimum number of threshold to trigger the jump, τ_{min}, of 7. We use 14, two times larger than τ_{min}, for the maximum number of threshold, τ_{max}. We calculate the number n_B of broadcast packets to be sent for each jump based on the the target reliability of the broadcast failure rate, which is set to be 99.9 [%]. In the following subsections, we show the simulation results averaged over 20 simulations where each of the simulation lasts for 3000000 [slots] which corresponds to almost 30 minutes, having a slot of 625 [μs].

5.2 Simulation Results

5.2.1 *Effect of Hopset Size on Average Goodput Performance*

Fig. 5 shows average goodput against hopset size H with different average number of piconets. Here, the goodput is defined as the fraction of time that is used by successful packet transmissions excluding overhead packets to convey the jump information. The average goodput is obtained by averaging the goodput of all the piconets which have been in the hotspot during the simulation. For the reference, we put the performance of collocated piconets that use conventional pseudorandom FH, where in each slot the piconet picks a frequency pseudorandomly from the complete set of 79 channels.

First, from this figure, we can see that the smaller H results in the higher average goodput performance. With the smaller hopset size, each piconet occupies smaller portion of the frequencies, and it is easy for collocated piconets to find hopsets which are not overlapping with each other. This results in the smaller number of collisions and higher average goodput performance.

Second observation is that the performance of FR becomes closer to that of pseudorandom FH as the hopset size and average number of piconets increase. In these "congested" situations, most of the hopsets of collocated piconets are overlapped, and the rolling and random jump in FR result in similar behavior to the pseudorandom FH, which leads

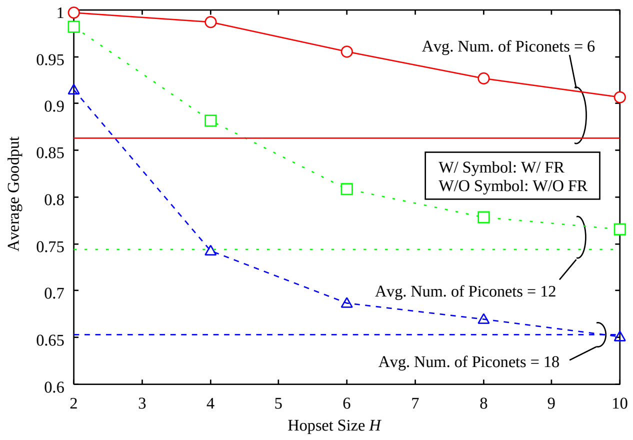

Figure 5: Average goodput against hopset size H with different average number of piconets in the hotspot.

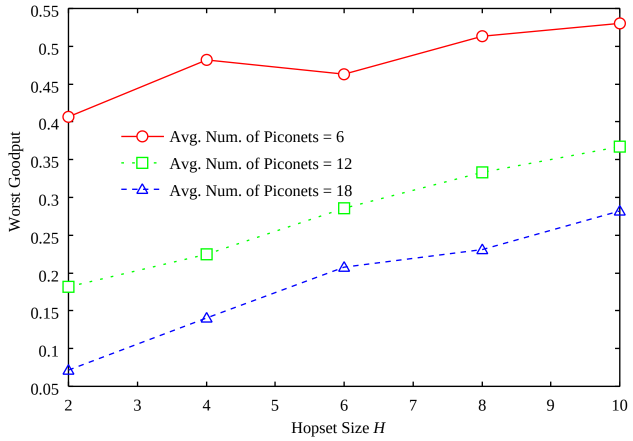

Figure 6: Worst goodput against hopset size H with different average number of piconets in the hotspot.

to the almost same performance for FR and pseudorandom FH.

5.2.2 *Effect of Hopset Size on Worst Goodput Performance*

Although the smaller hopset size always leads to higher average goodput performance, it can cause temporally high collision probability if more than two hopsets of collocated piconets are overlapped. This is because collision probability of overlapped piconets depends on the number of frequencies employed for FH. In order to check this adverse effect, we measured the goodput experienced by each piconet for every 6 second after every jump, and picked the worst value during the simulation as *worst goodput*. We plot the worst goodput against the hopset size with different average number of piconets in Fig. 6. As we can see from the figure, as the hopset size becomes smaller, the system has the smaller worst goodput. This result means that the smaller hopset size does not necessarily lead to the best performance, and we have to carefully select the optimum hopset size according to the requirements from applications.

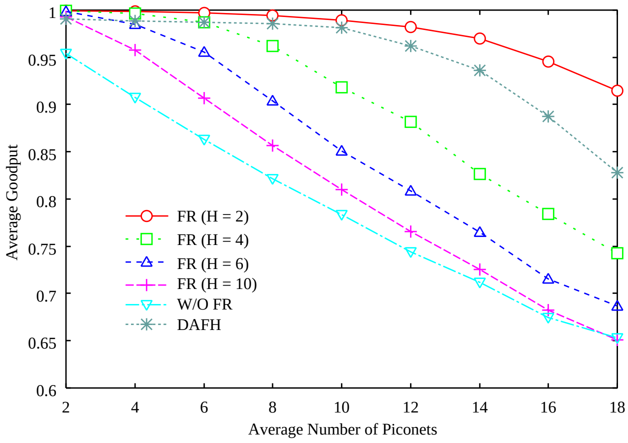

Figure 7: Average goodput against average number of piconets in the hotspot with different hopset size.

5.2.3 *Impact of FR on Average Goodput Performance*

Fig. 7 shows the average goodput against the average number of piconets with different hopset size. As a reference, we have put the performance of DAFH algorithm from [12]. If conventional pseudorandom FH is applied, the goodput of each individual WPAN is severely degraded as the average number of collocated piconets increases. This is due to the fact that each piconet uses the complete set of frequencies, such that the probability that a piconet will experience packet collision increases almost linearly with the number of collocated piconets. On the other hand, it can be seen that FR significantly improves the average goodput as compared to the case with pseudorandom FH. Indeed, for small H and low number of collocated piconets, the average goodput of FR is close to 1.0. This means that piconets experience very few collisions by achieving the orthogonal assignment of hopset for most of the dwelling time in the hotspot. The degradation of the average goodput in FR is due to the collisions in overlapping piconets experienced for triggering the jump and the overhead to convey the jump information.

As compared to DAFH, FR with $H = 2$ yields better average goodput, while DAFH outperforms FR for $H > 2$. Although DAFH has comparable performance with FR with $H = 2$, note that DAFH is not compliant with the current regulation for FH in unlicensed band. However, as shown in the previous subsection, $H = 2$ is not the best choice from the viewpoint of worst goodput. Conclusively, FR can comply to the requirements form the higher layers by trading off the average and worst goodput, while being in accordance with the regulation.

5.2.4 *Performance of Broadcast Failure Rate*

We show in Table 1 broadcast failure rates with different average number of piconets against hopset size. Here, broadcast failure is an event when there is error in all packets from a train of n_B broadcasts. Broadcast failure rate is the ratio of the number of broadcast failures to the total number of jumps triggered during the simulation in all piconets. Recall that we set the broadcast reliability to 99.9 [%], which means that the target broadcast failure rate is 0.1 [%]. From this table, we can see that, for the average number of piconets of 6, the broadcast failure rate is close to the target value of 0.1 [%]. However, for the larger average number of piconets and smaller hopset size, we have slightly higher value of the failure rate than the target value. To explain this effect, observe two collocated π_k and π_l. Let μ_k be triggered at t_0 and let it calculate n_B, such that the random jump is applied in $t_1 > t_0$. Note that the overlapping between π_k and π_l can be such that if π_l does the hopset change due to its nominal rolling, the number of common frequencies for π_k and π_l increases by 1. Hence, if π_l does such hopset change in t, where $t_0 < t < t_1$, then the actual PER induced to π_k is increased. This is especially severe when $H = 2$, because the number of common channels increases from 1 to 2. In that case, from equation (18) it can be seen that the actual PER after t_0 may have approximately double value with respect to the PER estimated before t_0. Nevertheless, the Table 1 shows that the broadcast reliability can be preserved to be of almost the same order as the target reliability.

Table 1: Broadcast Failure Rate with the different average number of piconets against different hopset size

H	Failure (%) ($\bar{N} = 6$)	Failure (%) ($\bar{N} = 12$)	Failure (%) ($\bar{N} = 18$)
2	0.1299	0.4241	0.7232
4	0.0878	0.2907	0.3203
6	0.1603	0.1756	0.2152
8	0.1499	0.1525	0.1991
10	0.0912	0.1244	0.2279

5.2.5 *Effect of Packet Errors due to Channel Noise*

Fig. 8 shows average goodput against the average number of piconets with different hopset size. The PER due to channel noise $\hat{p}_n$ is set to be 1 [%]. By comparing the results with Fig. 7, we can see that the goodput performance is degraded by the channel noise, especially for the smaller hopset size. The noise may trigger the master μ_k and μ_k initiates random jump even if there is not self-interference from the other piconets. The implication of this is that piconet π_k unnecessarily leaves a hopset that does not overlap with a hopset of any collocated piconet π_l. The new, randomly selected hopset of π_k, may in turn overlap with the hopset of some π_l. Note that, in the absence of noise, if a set of collocated piconets has established an orthogonalized rolling, then this situation can be changed only if a newly arrived piconet has a hopset that overlaps with the hopset of some of the present piconets. Therefore, the overall effect of the noise is that a piconet is more frequently urged to do random jumps and thereby more frequently entering into non-orthogonalized situations. Furthermore, the smaller hopset size causes higher collision probability once they cause the self-interference to each other. That is why the performance of the smaller hopset size is largely degraded. However, even with the maximal value of the estimated PER due to channel noise, the performance of FR is always better than that of the pseudorandom FH. This result means that FR is robust to channel noise. The robustness comes from the design of minimal threshold value to trigger the jump, τ_{min}, in which the maximal estimated PER due to channel noise is taken into account to decrease the number of noise-triggered jumps.

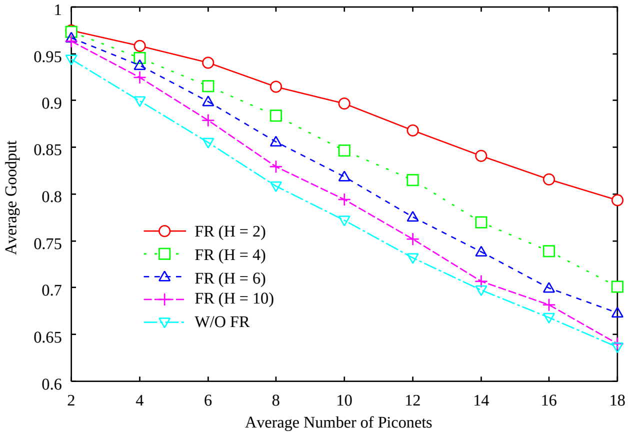

Figure 8: Average goodput against average number of piconets in the hotspot with different hopset size with PER due to channel noise of 1 %.

6. DISCUSSION

This section is reserved to discuss some important details, not implemented in the current basic version of FR, as well as possible extensions to the basic FR mechanism.

Although the broadcast of the hopset information is done with high reliability, it can happen that one or more slaves do not receive correctly all n_B broadcast packets from the master. Let the master apply a random jump at slot t_0. Then, the slave that did not receive the information about the hopset change loses the synchronization to the piconet channel. That is, for any slot $t_1 > t_0$, the slave cannot correctly estimate the frequency that should be used in the piconet. The actual implementation of the FR must be equipped with a fault-tolerance mechanism to cope with this situation. For that purpose, we assume that the pseudorandom values of the random jumps are generated by the master at the piconet initialization and shared with all slaves. If a slave does not detect transmission from the master after a timeout, it adds the next unused value of the random jump to its generation offset and tries to look for the master's transmissions in another hopset.

To describe another type of problem introduced by FR, let π_k roll to a new hopset in t_0 and let it be triggered in some $t_1 < t_0 + T$. If the estimated PER is high, then n_B is large, such that π_k cannot manage to send all the broadcast packets until $t_0 + T$. If the cause of such high error rate are other collocated piconets, then it may happen that no piconet can initiate hopset change by a random jump, since the estimated number of broadcasts exceeds the rolling interval. Hence, if no measure is taken, the basic version of FR cannot cope with this situation and high PER is tolerated. In other words, it can happen that a master is triggered in several consecutive rolling intervals without initiating random jump. Then, we can simply set the following rule: let the master μ_k be triggered in W consecutive intervals, but without initiating random jump. Then, μ_k can use the $(W + 1)-$th rolling interval for broadcasting the information about hopset change and apply the random jump at the start of $(W + 2)-$th rolling interval.

Here we briefly discuss the possible extension of the FR to the case when a frequency-static interferer (e.g. WLAN) is collocated with the WPAN. Note that DAFH is inherently immune to static interferers [12]. The strategy of removing the bad channels from the hopset, discussed in the Introduction, can be also applied in FR. For example, the piconet can make statistics of PER for individual channels. Let the frequency-static interferer be present at channel m_1. Then, whenever m_1 is in the hopset, the piconet estimates a high error rate at this channel. If such situation occurs for several consecutive rolling intervals, then the master decides to remove the channel m_1 for some time and do the rolling only over $M - 1 = 78$ channels. This procedure can be applied to remove other channels, but the total minimal number of channels used in the piconet cannot be less than 15.

We have observed that the noise can degrade the performance of FR, especially for low hopset size. This can be tackled by designing an adaptive triggering threshold; that is, after being triggered several times, a master can increase the value of $\tau_{\min}$, which basically means an online correction of the maximal noise estimate. Besides that, the adaptive threshold can be used to further stimulate cooperation among piconets. In this paper we choose the values $\tau_{\min}, \tau_{\max}$ to be identical and fixed for all piconets. The adaptive threshold would mean that each piconet adapts the interval $[\tau_{\min}, \tau_{\max}]$ individually. For example, after being triggered several times within some period T_k, the piconet π_k may increase $\tau_{\min}$ and $\tau_{\max}$ within the next period of length T'_k. Such action makes π_k more difficult to be triggered. On the other hand, let there be π_l which becomes collocated to π_k and which has not been triggered within a longer time interval, such that it has low values for $\tau_{\min}$ and $\tau_{\max}$. If π_k and π_l start to interfere, then π_l is more likely to be triggered first, initiate random jump and experience some performance degradation due to the overhead for hopset change. The mechanisms to apply adaptive threshold should depend on the tolerable error in the piconet and the requirements imposed by the link- and higher layers.

It must be noted that in this paper we have assumed that all piconets use the same value of H which is chosen in a predefined manner. In fact, the described FR mechanism can be applied when each piconet chooses its H in a distributed manner. In that case, a discussion is needed on what are the criterions that a piconet applies in order to choose certain hopset size. Note that, if H is small, then the piconet can easily find a “free” position after a jump. However, with small H, the piconet is very vulnerable to the channel impairments (e.g. frequency selective fading) and frequency-static interferer. The decision on what H should certain piconet choose is again influenced by the requirements from the link- and higher layers. A more elaborate design of the FR, taking into account all extensions described in this section, is out of the scope of this paper and is the subject of our future work.

7. CONCLUSION

In this paper we have proposed the Frequency Rolling (FR) for WPANs. It is a novel method for frequency hopping in unlicensed spectrum which enables the collocated WPANs to cooperatively avoid the self-interference. Bluetooth is a current technology that supports WPANs based on pseudorandom FH in the ISM band. The design of FR is founded on the careful consideration of the current regulation of the FH systems in the unlicensed ISM band. We prove that a WPAN that uses FR never occupies the channels in the unlicensed spectrum more than what is permitted by the current regulation. In its operation, the FR uses solely the observed packet error rate (PER) as an algorithm input. The effect of the FR over a longer interval is that the WPANs use all disposable channels in an implicit time-division manner. We compare the goodput offered by FR to the goodput of the collocated piconets when conventional pseudorandom FH is used. The simulation scenario includes a randomized dynamic pattern by which the WPANs become/cease to be collocated. Our simulation results show that FR has superior goodput performance even under such dynamic scenario. However, we have shown that if the parameter choice leads to the best average goodput, it also results in the minimization of the worst goodput experienced by a WPAN. Therefore, a concrete choice of the parameters in the FR must be driven by the application. The design of the FR is made robust with respect to the packet errors that are introduced by the noise and we also discuss some issues pertaining to the fault-tolerant implementation of the FR. In summary, this paper introduces the generic idea behind the FR. It opens the possibility to be extended further and developed into a complete solution for adaptive frequency hopping for the systems in unlicensed spectrum. The first step in the future work is to extend the FR in order to account for the scenarios in which the WPANs are collocated to a frequency-static interferer, such as WLAN.

8. REFERENCES

[1] Bluetooth SIG, “Bluetooth Core Specification,” Nov. 2003. [Online]. Available: http://www.bluetooth.com

[2] IEEE 802.15 Working Group for WPAN . [Online]. Available: http://grouper.ieee.org/groups/802/15/

[3] Federal Communications Commision (FCC), *Part 15.247: Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz*, In Part 15: Radio frequency devices, Oct. 2002.

[4] N. Golmie, R. van Dyck, and A. Soltanian, “Bluetooth and 802.11b interference: Simulation model and system results,” in *Proc. IEEE GLOBECOM’01*, Tech. Rep. IEEE802.15-01/195R0, Apr. 2001.

[5] A. El-Hoiydi, “Interference between Bluetooth networks - upper bound on the packet error rate,” *IEEE Commun. Lett.*, vol. 5, no. 6, pp. 245–247, June 2001.

[6] T.-Y. Lin and Y.-C. Tseng, “Collision analysis for a multi-Bluetooth picocells environment,” *IEEE Commun. Lett.*, vol. 7, no. 10, pp. 475–477, Oct. 2003.

[7] F. Mazzenga, D. Cassioli, P. Loreti, and F. Vatalaro, “Evaluation of packet loss probability in Bluetooth networks,” in *Proc. IEEE Intl. Conf. Commun (ICC’02)*, vol. 1, New York, Apr. 2002, pp. 313 –317.

[8] G. Pasolini, “Bluetooth piconets coexistence: Analytical investigation on the optimal operating conditions,” in *Proc. IEEE ICC ’03*, vol. 1, Anchorage, Alaska, May 2003, pp. 198–202.

[9] B. Treister, K. C. Chen, and A. Batra, “Clause 14.3: Adaptive frequency hopping,” IEEE P802.11 Working Group Contribution, IEEE P802.15-TG2-366r1, July 2001.

[10] IEEE 802.15 Coexistence Task Group. [Online]. Available: http://www.ieee802.org/15/pub/TG2.html

[11] C. F. Chiasserini and R. Rao, “Coexistence

mechanisms for interference mitigation between IEEE 802.11 WLANs and Bluetooth," in *Proc. 21st IEEE INFOCOM*, vol. 2, New York, June 2002, pp. 590 –598.

[12] H. Yomo, P. Popovski, and R. Prasad, "Adaptive radio resource sharing for collocated Wireless Personal Area Networks," in *Proc. IEEE WPMC '03*, Yokosuka, Japan, Oct. 2003.

[13] C. de M. Cordeiro and D. P. Agrawal, "Mitigating the effects of intermittent interference on Bluetooth ad hoc networks," in *Proc. 13th IEEE Intl. Symp. PIMRC'02*, vol. 1, Lisbon, Sept. 2002, pp. 496 –500.

[14] A. Zanella, A. M. Tonello, and S. Pupolin, "On the impact of fading and inter-piconet interference on Bluetooth performance," in *Proc. IEEE WPMC '02*, vol. 1, Honolulu, Hawaii, Oct. 2002, pp. 218–222.

[15] T. Camp, J. Boleng, and V. Davies., "A survey of mobility models for ad hoc network research," *Wireless Comm. and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends, and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[16] L. Kleinrock, *Queueing Systems Volume 1: Theory.* John Wiley & Sons Inc., 1975.

APPENDIX

THEOREM 2. *A system designed according to **S2** cannot produce channel occupancy that is higher than what is allowed by the regulation **S1**.*

PROOF. Let us observe an interval $T_1 > 6$ [sec] and let a piconet π_{S2} satisfy **S2** in each subinterval $T_0 \subset T_1$, where $T_0 = 6$ [sec]. π_{S2} may use any channel from C_M in T_1. Without losing generality, let 0 be the channel with maximal occupancy in T_1. Denote this occupancy $t_0^{S2}(T_1)$. Let a reference piconet π_{ref} use only the first 15 channels $C_{15} = \{0, 1, \ldots, 14\}$. We want to show that we can always create a hopping pattern for π_{ref} within T_1 which satisfies **S1**, while the occupancy of channel 0 in π_{ref}, denoted $t_0^{ref}(T_1)$, is always $t_0^{ref}(T_1) \geq t_0^{S2}(T_1)$. Referring to Fig. 9, let π_{ref} use the channel 0 in the same slots within T_1 in which π_{S2} uses channel 0. The slots in which π_{S2} does not use channel 0 are "filled" with the other 14 channels of π_{ref}, as it is done on Fig. 9. After channel 14 is used, the next slot not used by channel 0 is again filled with channel 1. There are two cases:

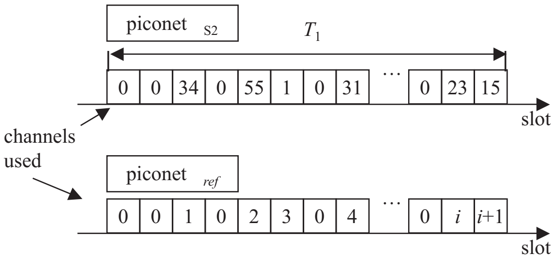

Figure 9: The maximal occupancy in piconet π_{S2}, which is in agreement with S2, within the interval T_1 occurs for channel 0. We can always find a reference piconet π_{ref} that uses only 15 channels for hopping and is compliance with S1, while the occupancy of channel 0 in π_{ref} is identical to that of π_{S2}.

1. In each interval $T_0 \subset T_1$, channel 0 is used for exactly 0.4 [sec]. Then, from the uniform way in which the channels $\{1, 2, \ldots, 14\}$ are filled into slots unused by 0, each channel is used for 0.4 [sec] within each T_0. In this case, the hopping pattern of π_{ref} is in agreement with **S1**, while $t_0^{ref}(T_1) = t_0^{S2}(T_1)$.
2. There are intervals $T_0 \subset T_1$ in which channel 0 is used for less than 0.4 [sec]. Then if we just fill the slots unused by 0 with channels from $\{1, 2, \ldots, 14\}$, the statement **S1** would be violated, since there must be at least one channel from $\{1, 2, \ldots, 14\}$ that is used for more than 0.4 sec. Hence, we have to make the usage of channel 0 to be 0.4 [sec] by additionally using it in slots in which π_{S2} does not use it, and then apply the previous case in which **S1** is satisfied. Clearly, now we have $t_0^{ref}(T_1) > t_0^{S2}(T_1)$.

□