Scheduling for Frequency Hopped Access with Randomized Frame Lengths

Bill Kiki-Sagbe
Department of Electrical Engineering
École de Technologie Supérieure, Université du Québec,
1100 Rue Notre-Dame Ouest,
Montréal (Québec), Canada H3C lK3
Email: bill.kiki-sagbe@lacime.etsmtl.ca

François Gagnon
Department of Electrical Engineering
École de Technologie Supérieure, Université du Québec,
1100 Rue Notre-Dame Ouest,
Montréal (Québec), Canada H3C lK3
Email: Francois.Gagnon@etsmtl.ca

Abstract—**This paper deals with the allocation of resources to the MAC layers of frequency hopped wireless networks. The research has been carried out on new efficient scheduling methods for robust systems to handle distortion and reception. The frequency and time slots are variable during the same point to point or point to multipoint communication. The chosen time intervals are small enough so as to limit distortion. By combining random packets with priority settings such as signal strength and band allocation queue length, the total capacity of the link was increased up to twice that offered by random allocations. This is functional as long as there is band availability; a better latency is also obtained.**

Keywords-scheduling; frequency hopping; random frame length; high capacity; ad hoc.

I. Introduction

At the moment, as far as wireless communication is concerned, an ad hoc network system is one of the easiest to install because it does not require any fixed infrastructure; hence its desirability for military purposes, especially in hostile areas. These networks remain vulnerable because any good receiver can potentially receive or distort its messages [1]. Information confidentiality is therefore at risk.

This paper tackles the issue by proposing a new frequency hopping scheduling method. The literature in this field already has frequency hopping scheduling methods such as Frequency Hopping Spread Spectrum (FHSS) [2], Adaptive Frequency Hopping (AFH) [3] and many others. Most of these methods, but for a few exceptions, use FHSS as the basic principle. The latter works by exchanging a Service Set Identifier (SSID) code between the transmitter and the receiver. This code indicates the chosen hopping. The sizes of the time and frequency slots are fixed and are common knowledge. The possible hopping order and frequency interval between two successive hops are known in advance [2]. The security of the frequency hopping scheduling methods in the previous solutions derives from the fact that the equipment (spectrum analysers, oscilloscope) needed for long distance reception is costly and sophisticated.

The hypothesis studied here is one that varies the size of the frequency and time slots during the same communication. These variations are done in a dynamic fashion without taking into account the prior subdivision of the available band. The variable packet size renders jamming, by a malicious system much more difficult. For example, a sophisticated jammer follower requires a form of synchronization information to work efficiently[4]. This approach should also offer interesting actual throughputs because the variation of packet sizes could be used to manage communications according to their priority. Research work such as [5] also makes it possible to have different packet sizes, whose variations are however not dynamic. The possible sizes are known and this solution is proposed for frame retransmission.

The rest of the work is organised as follows: in section II, the hypotheses and specifications of the work are presented. In section III, we present the proposed approach as well as the shortcomings that were discovered and solved. The obtained results, an interpretation and a conclusion are presented in sections IV and V respectively.

II. Hypotheses and Specifications

A. The Hypotheses

Obtaining an efficient frequency hopped scheduling with random packet sizes is the main objective of this work. By maximizing bandwidth use and by allocating it according to the load to be transmitted, interesting actual throughputs and better latency times are obtained. To succeed, the distribution of resources according to user load must be varied so that the link capacity offered to this user can vary.

When communication begins, the master node sends to its slaves a frequency order that is valid only for the communication in progress. This would require the communication of extra bits in relation to FHSS. Here, FHSS is considered as the reference in the random access scheduling.

In FHSS, the users have the same probability of choosing a frequency slot $p=1/q$ (with q being the number of possible frequencies). The idea here is to vary this probability on the basis of certain criteria such as: the queue size per user and link capacity. This probability can then be conditioned by the

This work has been supported in part by Ultra Electronics TCS and the Natural Science and Engineering Council of Canada as part of the “High Performance Emergency and Tactical Wireless Communication Chair” at École de technologie Supérieure.

overall state of the system. With this variation, better service times can be obtained.

B. Specifications

For experimental purposes, the variable size hops range between *W/20* and *W Hertz,* were *W* is typically in order of $40*10^6$ to $200*10^6$. In this paper, one value of *W* is used to present all result. Each hop will be associated with a time slot which will itself vary between *20000/W* sec. to *80000/W* sec.. These values can vary with usage, without, however, affecting the algorithm results presented in this paper.

The typical characteristics of a high capacity radio have been used to implement this algorithm. These specifications are: a frequency band varying between F_c and F_c+30*W, were F_c , carrier frequency, is typically in order of $200*W$ Hz to $300*W$ Hz; transmission on *W* Hz maximum and reception on $5*W$ Hz maximum; simultaneous transmission and reception (full duplex).

III. CONCEPT DESIGN

The proposed approach is presented in this section in the form of a flow chart showing the different functions (figure 1).

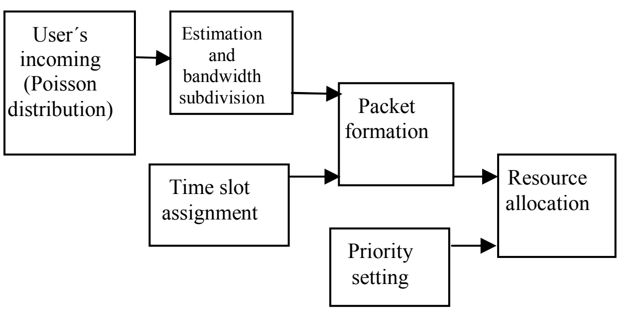

Figure 1. Flow Chart Of The Proposed Solution

First of all, the algorithm generates users with a certain amount of information that has to be transmitted. For a certain number of users and a certain data size, the M/M/S arrival law based on the Poisson distribution [6] is used. Without the loss of generality, the packet size is fixed at 3000 Bits.

A. Estimation bandwidth

Channel estimation consists of discovering the band available for distribution among the users. It is at this stage that cluster bandwidth allocation is done. When there are several clusters, the entire band is allocated according to the amount of information that has to be transmitted by each cluster. This estimation is done following the rule of three. For a *Bw_total* available band and a *k* number of clusters, each with an x_i load to transmit, we get $Bw\_cluster_i$ for each cluster on the band.

$$Bw\_cluster_i = \frac{x_i * Bw\_total}{\sum_{i=1}^{k} x_i} \tag{1}$$

Nevertheless, the maximum size of the $Bw\_cluster_i$ is set at $5*W$ Hz as a result of the specifications considered in this study.

After choosing the band size per cluster, the choice of the size and position of the frequency slots is done according to the algorithm illustrated in figure 2. The band for each cluster is subdivided into N_{user} (number of users per cluster) random values whose total is $Bw\_cluster_i$. This subdivision does not take the priority settings into consideration.

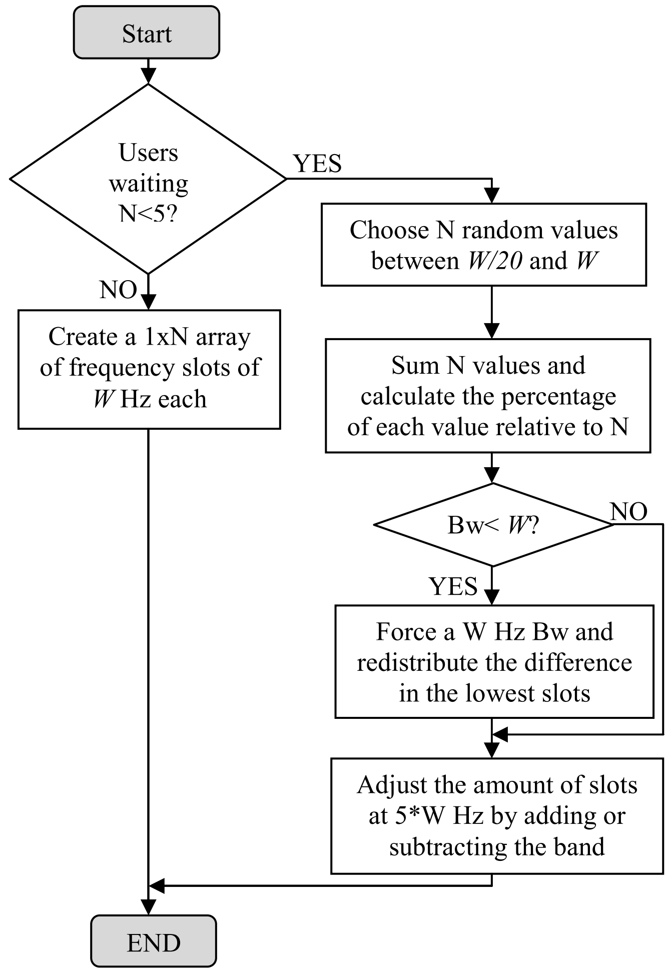

Figure 2. Frequency Slot Constitution Algorithm

The frequency slot subdivision algorithm presented in figure 2 works as follows:

First of all, the algorithm discovers the number of users. If this number is less than or equal to 5, the algorithm subdivides the cluster band into identical N_{user} slot values, while insuring that these slots do not exceed *W* Hz.

If the N_{user} is higher than 5, N_{user} values are chosen randomly between *W/20* and *W*, which is the $Rand_i$ value. The algorithm then calculates each value's percentage in relation to their total. The result is the $\%\_Rand_i$.

$$\%\_Rand_i = \frac{Rand_i}{\sum_{i=1}^{N} Rand_i} * 100 \tag{2}$$

It associates the calculated percentage to the percentage of the band covered by this user in the *Bw_cluster*. Therefore, the slot frequency is obtained by:

$$Bw_{user} = \frac{\%\_Rand_i * Bw\_cluster}{100} \quad (3)$$

If this value is not included in the *W/20* to *W* Hz segment, then an adjustment is done, with the total always being equal to *Bw_cluster*.

This algorithm can generate up to *5*W* frequency slots on a *5*W* Hz band with a negligible calculation time. It is used after each time slot. The time slot is chosen dynamically within an interval of *20000/W* sec. to *80000/W* sec. Given that the packet sizes vary, the users sometimes find themselves allocated packets which are much bigger than the size of their remaining queue. This results in band loss. To remedy this problem, the previous algorithm can be adjusted by checking the size of each user's queue after each time slot before generating possible packet sizes. This consists in identifying all the users whose remaining queue is less than the maximum size limit of the formed packets. The packets are then made-to-measure for these users by respecting the minimum and maximum limits of the frequency and time slots. The service for these users takes priority over all the others. Hence, part of the lost band is, in the meantime, recovered to serve other users. The following figure illustrates band recovery:

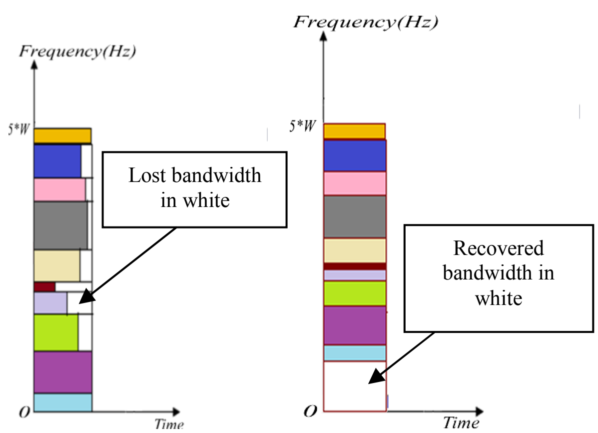

Figure 3. Lost Bandwidth Vs Recovered Bandwidth

B. Packet formation

As presented by Shannon [7], the capacity of a link is determined by the bandwidth and the SNR. Hence, the Shannon throughput gives:

$$D = B * \log_2(1 + SNR) \quad (4)$$

Were **B** denotes the bandwidth and **SNR** is the signal to noise ratio.

Each resulting packet corresponds to a specific throughput lasting a certain time. The packet size is obtained by:

$$Packet\_length(t) = Time\_slot(t) * D(t) \quad (5)$$

The packets have an equal chance of appearing, regardless of their size. Once the packets are formed, their association to users is done according to priority parameters.

C. Priority Parameters and Packet Allocation

The priority parameters used are the size of the user's queue and the available energy. It is important to allocate bigger packets to those users who have more queues to transmit. Simulations show that when priority parameters are not used, the queues of waiting users increase exponentially starting from a 60% network utilisation rate. This bad latency is due to the equiprobability that the users with small loads to transmit will receive big packets and use only part of the resource, while the users with big loads to transmit receive small packets. The variable capacities of the offered links can range from:

$$C_{min} = (W / 20) * \log_2(1 + SNR) \text{ Bits/sec to } C_{max} = 20 * C_{min} \quad (6)$$

IV. Experiment Results and Discussion

The algorithm was implemented under Matlab. The mathematical modelings of the aforementioned functions were used to obtain the performances of the solution. These performances were compared with those of FHSS. For all the results presented in this section, the binary error rate is supposed to be zero. The throughput used is thus the maximum capacity given by Shannon's law as presented in section II.

A. Recovered bandwidth

As presented in section II, part of the band is lost during allocation. This loss is due to the fact that with the size of the packet not being fixed, the size of the user's queue is not often its multiple. This problem is usually noticed at the end of the transmission when a user whose remaining queue size is less than the permitted packet size needs to be served. To remedy this problem, the solution proposed in subsection A of section II is used. Figure 4 and Table I present the recovered band after correction in bits.

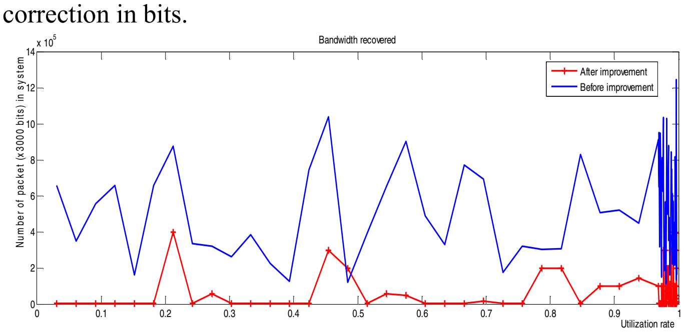

Figure 4. Evolution Of The Recovered Band According To The Network's Utilisation Rate

TABLE I. AVERAGE RECOVERED BAND VS. IDEAL RESULTS (IN BITS)

	Before solution	After solution	TARGET RESULTS
Mean	5.35 e+5	6.65 e+4	0
Standard deviation	2.77 e+5	1.08 e+5	0

The algorithm does not recover the entire lost band because of the small authorised packets even if the queue size is smaller.

B. Service Latency

Latency is a user's service time from the moment he gets into the queue. To obtain service latency, Little´s formula [8] is used:

$$N = lambda * T_m \quad (7)$$

With: ***N***: the average number of 3000bits packets, ***Lambda***: the average rate of packet arrival and ***Tm***: the duration of packet stay.

The simulation results of this algorithm are compared with those of FHSS and presented in figures 5 and 6.

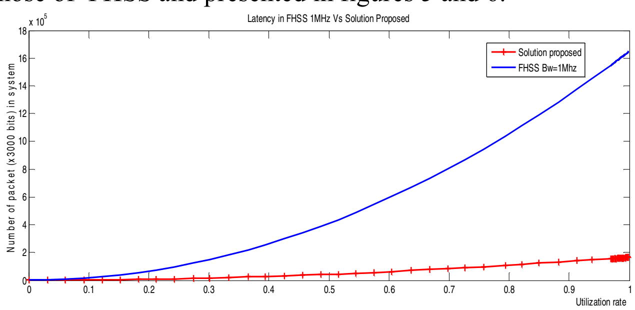

Figure 5. Latency Of The Proposed Solution Vs. The 1MHz FHSS

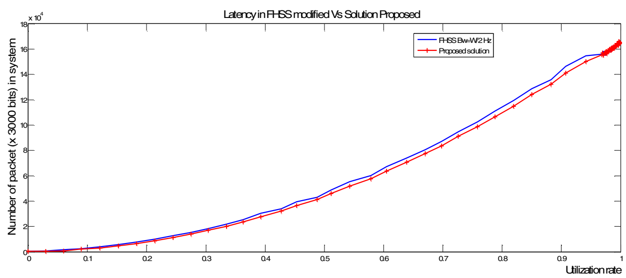

Figure 6. Latency Of The Proposed Solution Vs. The *W/2* Hz FHSS

The tests are done on a *5*W* Hz band for 10 served users. Firstly, a comparison was made with a FHSS offering 1MHz frequency slots for a $(8*10^6)/W$ sec. time slot. After simulation, figure 5 shows that the latency of the proposed solution is far better than that of FHSS. This result is obvious because, as a matter of fact, FHSS with *5*W* Hz frequency slots doesn´t covers 100% of the *5*W* Hz band for 10 users, while the solution in this paper covers the entire *5*W* Hz band.

For a reasonable comparison to be done, FHSS was modified to cover the entire band. In order to do this, the size of the frequency slots was modified to *W/2* Hz. This FHSS frequency slot value does not conform to any norm, but was chosen for experimental reasons. The results of this test are presented in figure 6. It turns out that the proposed algorithm offers a better latency than FHSS for the same covered band and the same number of users to be served. This gain in service time can be explained by the fact that throughput per user in the modified FHSS is fixed at *2.47*W* bits/sec, while in the proposed solution, this throughput per user varies and can double that of the modified FHSS, reaching *4.94*W* bits/sec as long as there is band availability. Given that the priority is applied if the load to be transmitted is big, there is a big probability of having throughputs that are bigger than FHSS throughput because from the moment some users have been served before the others, the users that are still being treated have to share the band left by the served users. This ends up increasing the average frequency slot per user. The presented curves are for all ten users. Thus, with this increase of slot average from *W/2* Hz to a higher value which can get as high as *W* Hz, the proposed solution offers a better latency than FHSS.

C. Actual Wireless Throughput

By using the same size as for the FHSS header for this solution, the approximate actual theoretical throughput was obtained. The FHSS header for each packet is 128 bits [2]. Therefore, the theoretical throughput gives:

$$\text{Throughput} = \frac{D(t)*Time\_slot(t) - sum\_headers}{\sum time\_slot} \quad (8)$$

The results of the experiment are presented in figure 7.

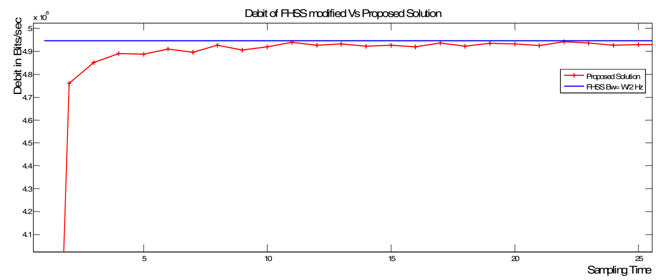

Figure 7. Actual Theoretical Throughput Of The Proposed Solution Vs. That Of FHSS Modified To Bw=*W/2* hz

Figure 7 presents the evolution of the actual theoretical throughput of the modified FHSS and the proposed algorithm. The FHSS throughput is slightly better. This result can be explained by the fact that to serve 10 users, for example, on a *5*W* Hz band with the characteristics of the modified FHSS (frequency slot=*W/2* Hz and time slot=$(8*10^6)/W$ sec.), $W/(4*10^5)$ packets are formed per second on the entire band. By using a 128 bit FHSS header, there is a throughput loss of: $(W/(4*10^5))*128$ Bits/sec.

In the proposed solution, the time slot varies between *20000/W* sec. to *80000/W* sec.. This means that within a second, the number of formed packets can vary between: *W/4000* to *W/1000* packets.

Supposing that, for the proposed solution, a fixed header size of 128 bits per packet formed is sufficient to manage the communication, the lost throughput varies between: *W/4000* *128 Bits/sec to *W/1000* *128 Bits/sec.

By increasing the time slot size, this loss can be minimized. As such, a simulation with time slots varying from $(4*10^6)/W$ sec. to $(12*10^6)/W$ sec., meaning a $(8*10^6)/W$ sec. average time slot, would result in the theoretical throughputs lost in the packet headers varying from *W/4688* bits/sec to *W/1562* bits/sec. By applying these time slot values to the proposed algorithm, the obtained latency is very poor. In fact, the latency increases 10 times faster than in the previous solution.

These poor results are due to the fact that by increasing the time slot, the segment variation of the packet size gets bigger, in which case, it becomes very likely that packet sizes will be considerably bigger than queue sizes. Hence, the bandwidth is lost, service time is increased and the queue grows exponentially. Therefore, time slot size influences latency and actual theoretical throughput differently. It is therefore necessary to find the right time slot segment in order to be able to obtain better latency as well as good actual throughput.

D. Latency With Considered Actual Wireless Throughput

By applying the actual throughput obtained at each scheduling latency, it turns out that the FHSS latency is slightly better than that of the proposed solution. Given that the differences are not obvious on a graph, the results are presented in table 2.

TABLE II. LATENCY RESULTS OBTAINED AFTER THE APPLICATION OF THE ACTUAL THEORETICAL THROUGHPUT

Utilization rate	Number of packets in FHSS	Number of packets in this scheduling	Difference between FHSS and This Scheduling
0.0318	608.76	632.32	23.55
0.2747	1.49 e4	1.50 e4	79.58
0.8828	1.37e+5	1.37e+5	209.44
0.9858	1.61e+5	1.61e+5	245.96
0.9981	1.48e+6	1.48e+6	2.39e+3

This table shows that with a better link capacity than that of FHSS, the proposed solution can have a lower actual throughput than FHSS. In order to improve this throughput, the proposed solution's time slot variation segment has to be increased. But, as it was presented in the *sub-section C.* of section IV., by disproportionately increasing this segment, the throughput is considerably diminished. This contradiction proves that, while trying to reduce the flow lost due to the header, there are limits of the time slot should not to exceed to have a good latency.

V. CONCLUSION

A new frequency-hopping scheduling method has been proposed. These frequency hopping schemes combined with variable length packets offer a solid resistance to distortion. It has been demonstrated that when there is band availability, better latency is obtained with this method as compared to FHSS.

This paper lays the foundations for a new approach of scheduling frequency hopping. The work presented is not exhaustive. Future research should focus on the implementation of the proposed solution on a platform such as the Bluetooth to present the bit error rate and compared with FHSS.

REFERENCES

[1] V. Gayraud, L. Nuaymi, F. Dupont, S. Gombault, and B. Tharon, "La sécurité dans les réseaux sans fil ad hoc," 2003.

[2] P. Roshan and J. Leary, *Réseaux Wi-Fi : notions fondamentales*. Paris: CampusPress, 2004.

[3] C. Hodgdon, "Adaptive frequency hopping for reduced interference between Bluetooth and wireless LAN, May 2003," *Ericsson Technology Licensing*.

[4] E. B. Felstead, "The vulnerability region of frequency hoppers against follower jammers," 2009, pp. 1-7.

[5] J. K. AHN, K. J. KIM, J. H. LEE, and S. H. YOON, "METHOD FOR TRANSMITTING DATA USING FREQUENCY HOPPING SCHEME AND METHOD FOR DETERMINING A SUB-BAND FOR THE FREQUENCY HOPPING," ed: WO Patent WO/2008/084,977, 2008.

[6] D. P. Bertsekas and R. G. Gallager, *Data networks*, 2nd ed. Upper Saddle River, N.J.: Prentice-Hall, 1992.

[7] A. S. Tanenbaum, *Réseaux*, 4e éd. ed. Paris: Pearson Education France, 2003.

[8] L. Kleinrock and R. Gail, *Queueing systems : problems and solutions*. New York: Wiley, 1996.