A Novel Scheme to Interconnect Multiple Frequency Hopping Channels into an Ad Hoc Network

György Miklós [*], *Ferenc Kubinszky, András Rácz,*
Zoltán Turányi, András Valkó
Traffic Analysis and Network Performance Lab,
Ericsson Research, Hungary

Miklós Aurél Rónai, Sándor Molnár
High Speed Networks Laboratory,
Budapest University of Technology and Economics

Frequency hopping radios have very attractive features to be used as PAN links, but their use in ad hoc networking is problematic because of the difficulty to synchronize the channels and coordinate transmission attempts. We propose a novel mechanism to interconnect multiple frequency hopping channels into an ad hoc network based on an adapted version of CSMA/CA. The performance of the proposal is investigated using analytical and simulation tools. By using multiple channels, we achieve significant improvement in aggregate throughput despite the penalty of switching between channels. We show how this performance penalty can be decreased by grouping devices based on the traffic pattern.

Keywords: ad hoc, PAN, CSMA/CA, MAC, frequency hopping, multiple channels.

I. Introduction

Frequency hopping spread spectrum radio technology [12] possesses a number of advantages that has motivated its selection in PANs and also other radio systems. These advantages include robustness against interference, fading and noise, simplicity and low cost of implementation. A key advantage is that a number of such systems can be independently operated in the same coverage area with limited interference. There is no hard capacity limit for the number of interferers. Increasing their number results in a graceful degradation of performance.

Specifically, Bluetooth [3, 6] is one of the PAN technologies that makes good use of the advantages of frequency hopping, as it has been designed to allow a large number of channels to co-exist in the same coverage area. Bluetooth is primarily intended as a cable replacement radio technology, using a short range (10m) radio interface designed to facilitate the development of very small and cheap implementations. Thanks to the frequency hopping radios, the system is indeed robust against interference caused by other Bluetooth and non-Bluetooth interferers in the same band [19].

When a large number of frequency hopping channels are present, the question of channel establishment and synchronization must be addressed. In the case of Bluetooth, devices have to synchronize using a paging procedure to establish the channel referred to as a piconet. The node initiating the procedure becomes the master of the piconet. The formation of the piconet takes a relatively large overhead of several seconds, but makes data transmission straightforward once the piconet is established. This is in harmony with the requirements of cable replacement applications where a connection needs to be set up rarely, typically only once when the application is started or re-started. Once the piconet is established, the frequency hopping sequence is derived from the clock and address of the master node. The timing synchronization is defined by the transmissions of the master.

The channel establishment procedure makes it possible to set up multiple frequency hopping channels in the same coverage area. In Bluetooth, the hopping sequence is dependent on the master, which is why each piconet is using a different hopping channel. Although there can be a certain amount of interference, this provides a good separation of the radio channels. This also provides a logical separation since devices in different piconets do not even have to know about each other at all. Devices in the same piconet, on the other hand, need to be co-ordinated. This is performed by the master node using a centralized polling-based scheduling mechanism.

Once we have a large number of devices capable of communicating over a number of independent frequency hopping channels, it becomes a natural requirement to be able to connect them into a single PAN. This step, however, is problematic. In the case of Bluetooth, it is theoretically possible to form a network even though the system has been optimized for cable replacement scenarios. The specification allows a device to be a member in multiple piconets and several piconets can be connected into a so-called scatternet. Figure 1 shows an example of such a scatternet where two laptops L1 and L2 and projector Pj, together with other accessories, are connected into a network. Such scatternet networks are made possible by the specification, but a number of important issues remain unresolved, such as how to decide about piconet membership and master roles (i.e., connection setup), how to route packets, how to schedule the presence of a node in multiple piconets, and how to discover and manage neighbours. These problems have to be resolved in extension protocols to the

[*]Corresponding author. Phone: +36 1 437 7633, Fax: +36 1 437 7767, E-mail: Gyorgy.Miklos@ericsson.com, Address: H-1037, Laborc u. 1, Budapest, Hungary

core Bluetooth specification. Research (see for example [9, 11, 13, 15, 17]) and specification work [4] is ongoing to address these issues.

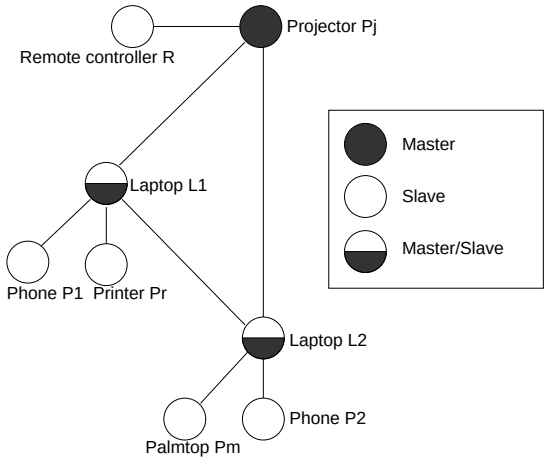

Figure 1: Example Bluetooth scatternet

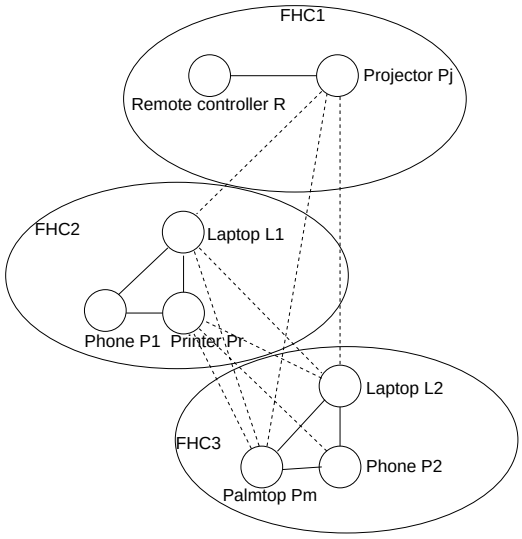

Figure 2: Example of the proposed MFHC PAN

In this paper we suggest a new approach to interconnect multiple frequency hopping channels into an ad hoc PAN. We propose Multiple Frequency Hopping Channel communication (MFHC) to address this problem, and investigate the performance of MFHC ad hoc networks. Our approach avoids the use of a scatternet network, and allows nodes to communicate with all neighbours that are in radio range in a connection-less fashion. Our solution uses CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) random access scheme [8] for each channel, with the extension that we allow a device to switch to a new frequency hopping channel (FHC for short) before each packet transmission. Each node has an associated home FHC that it follows by default. If a source node needs to send a packet to a destination node on the same home FHC, it uses the basic random access scheme on the common hopping channel. If, on the other hand, a source node needs to send a packet to a destination node that has a different home FHC than that of the source, then it switches to the home FHC of the destination and applies the random access scheme on the destination node's home FHC.

Figure 2 shows the application scenario of Figure 1 employing the proposed MFHC scheme. The devices form three frequency hopping channels, denoted by FHC 1-3. Nodes within the same FHC can communicate with each other directly using CSMA/CA. This is shown by the solid lines. Nodes can also send data to another node in radio range in another FHC by switching to the destination node's home FHC. This is how communication between nodes connected by a dashed line (and between every other pair of nodes in radio proximity) can take place. As the figure suggests, the MFHC scheme avoids the complexity associated with establishing a scatternet and selecting master and slave roles, determining, optimizing and maintaining the topology and scheduling transmissions. MFHC also avoids multi-hop communication between neighbours. Instead, nodes can send packets to any of their neighbours by switching to the destination node's home FHC and following the well-known CSMA/CA scheme. The solution allows the formation of a connected ad hoc network, but at the same time keeps the advantages of using a single frequency hopping channel for a group of devices. To achieve this, MFHC requires a neighbour discovery and synchronization mechanism. One potential neighbour discovery mechanism is discussed in [14].

The MFHC solution makes it possible to send data to a different node on another FHC, but it is clear that communication is more efficient when switching between FHCs is not needed. This raises the question of how to arrange the devices into groups using a common channel. One example is the one shown in the figure, but a number of other alternatives exist, including the important special case where each device has an associated FHC of its own. We will examine the performance trade-offs involved with grouping devices into FHCs. We will investigate static FHCs, but we note here that it is possible to make the FHC selection algorithms dynamic, in which a node can change its home FHC membership based on traffic measurements.

The paper is organized as follows. Section II reviews related frequency hopping systems and their networking capabilities. Section III describes the proposed MFHC solution. Section IV presents three basic FHC configurations and compares them through a simple analytical model. Section V investigates MFHC via simulations. Section VI concludes the paper.

II. Related Work

A number of existing and proposed systems use frequency hopping spread spectrum radios, providing a limited networking capability. Here we provide a brief overview of such technologies in addition to Bluetooth that has already been introduced above. Table 1 summarizes the main features of the systems considered in this paper.

Currently the most widely used ad hoc networking platforms are based on the IEEE 802.11 wireless LAN stan-

dard [6, 8]. At the MAC layer, multiplexing of traffic on a single channel is achieved by CSMA/CA. An RTS (request to send) - CTS (clear to send) - data - ACK four-way handshaking mechanism is defined. The RTS-CTS message exchange decreases the overhead of collision (when packets are long) and solves the hidden terminal problem [8]. IEEE 802.11 defines a number of physical layers, frequency hopping spread spectrum being one of them. However, communication is possible only in a single channel (between nodes in the same Basic Service Set in the 802.11 terminology). (Note that existing products that use the frequency hopping physical layer do not support fully distributed ad hoc operation even at a single channel, despite the fact that the standard allows this and defines a distributed time synchronization method. Instead, ad hoc operation is supported by products based on the direct sequence spread spectrum physical layer.) To use multiple channels, we have to have an infrastructure of connected access points. Without any infrastructure, it could be possible to use several independent hopping channels on the same coverage area to share the available spectrum, but only nodes on the same channel could communicate with each other. MFHC addresses this problem: it allows nodes in different channels to communicate.

The Hop-Reservation Multiple Access (HRMA) protocol is introduced in [16] for frequency hopping spread spectrum packet radios. The protocol uses a hop reservation and RTS-CTS handshake mechanism to guarantee collision-free operation even in the presence of hidden terminals. The protocol uses a designated frequency for control message exchange and requires timing synchronization over the whole network. By relying on this common channel that every node listens to, collision avoidance and hop reservation for data transmission can be achieved, so that multiple data transmissions use different frequencies. However, the requirement of synchronizing the whole network in time and using a single common signalling channel may imply performance and robustness bottlenecks.

The design concepts used in the High Frequency (HF) Intra Task Force (ITF) Communication Network are discussed in [5] employing frequency hopping spread spectrum radios. The proposal incorporates the Linked Cluster Algorithm that structures nodes into disjoint clusters making use of two TDMA frames that are synchronized over the whole network. Once the clusters are formed, a second procedure called Link Activation Algorithm controls how slots are allocated on the links. The available frequency band is divided into several sub-bands, and an independent network is formed in each sub-band. This makes it feasible to perform re-configuration of the network in one sub-band while communication can still continue in other sub-bands. However, the complexity and performance implications of re-configuration of the clusters and schedules are unclear.

The proposed MFHC scheme is novel in the way it establishes a connected ad hoc network when multiple unsynchronized frequency hopping channels exist on the same coverage area. The architecture combines many of the advantages of frequency hopping systems. It facilitates the use of low cost frequency hopping radios as in Bluetooth, it is based on a simple connection-less approach with on-demand resource allocation as in the case of IEEE 802.11, it enables networking between all devices as in HRMA and HF ITF, but without the need for a network-wide synchronization mechanism. MFHC can be adapted to frequency hopping physical layers with very different characteristics, e.g., to the physical layer of Bluetooth or 802.11 FH. As we make numerical investigations (Sections IV and V), we will use parameters that are typical of today's Bluetooth implementations, but is clear that MFHC is applicable with a number of other physical layer parameter settings as well. We also note that MFHC could be used in multi-channel environments that are not using frequency hopping technology.

III. Multiple Frequency Hopping Channel Communication (MFHC)

To interconnect multiple frequency hopping channels that co-exist on the same coverage area, we apply an adapted CSMA/CA scheme. Channel access within a FHC is based on the CSMA/CA approach used by the IEEE 802.11 protocol [8]. This means that a node that has a packet to send on the FHC first waits until the channel becomes free for at least a minimum period of time, which we refer to as GS (guard space). Communication may begin at fixed slot boundaries. To resolve collisions due to more than one stations sending at the same time, a contention mechanism is applied as follows. Each station has a contention window, CW, and chooses a random backoff value B from the interval $[0, CW-1]$. In each slot when the channel is sensed free, the value of B is decreased if it is above zero. A node may transmit when the value of B reaches zero. If the transmission is successful, the value of CW is initialized to CW_{min}. If the transmission is unsuccessful, the value of CW is doubled unless it reaches CW_{max} and the transmission attempt will be repeated. This scheme ensures that collisions will be resolved after one or more stages of contention. (Note that in a practical implementation, it is important to limit the number of transmission attempts to avoid deadlock if the radio channel is down for any reason.)

We precede each packet transmission by an RTS-CTS message exchange, as in the 802.11 protocol. This handles the hidden terminal problem (the destination receives packets from a station that the source cannot receive from), and also decreases the overhead of contention in the case of long packets. In addition, the RTS-CTS message exchange provides a way for negotiation of parameters for the subsequent data transmission.

This scheme can be extended for multiple FHCs, as shown in the example of Figure 3. Even though it is allowed for a node to switch from one FHC to another, we associate a default FHC with each node, which we refer to as the home FHC of the node. The figure shows two FHCs, where FHC 1 is the home of nodes A and B, FHC 2 is the home of nodes C, D and E. A node may temporarily leave its home FHC, as node B does to visit FHC 2 (B'), but it returns to its home FHC as soon as it has finished contention

System	Networking	Channel Setup	Resource allocation	Synchronization
Bluetooth Specification	Single piconet	Piconet formation	Centralized scheduling	Piconet-wide
Bluetooth Scatternet PAN	Connection-oriented multihop	Scatternet formation algorithm	Distributed scheduling	Piconet-wide
802.11/FH ad hoc	Single BSS	Distributed synchronization	On demand, CSMA/CA	BSS-wide
HRMA	Connection-less	Distributed synchronization	On demand, hop reservation	Network-wide
HF ITF	Connection-oriented multihop	Linked Cluster Algorithm	Scheduled (Link Activation Algorithm)	Cluster-wide, network-wide
MFHC	Connection-less	FHC selection algorithm	On demand, CSMA/CA	FHC-wide

Table 1: Summary of related work and MFHC with ad hoc frequency hopping systems

or transmission. To initiate a data transmission to a node, we need to switch to the destination node's home FHC and wait until the node is available and the channel is free.

When the destination node's FHC is different from the source node's home, then the source node has to switch between the source and destination FHCs during contention. This is illustrated in the figure, where node B wants to send a packet to node C in FHC 2. First, it switches to FHC 2 (becomes B' after transition T1) and listens on the channel for at least a fixed amount of time (denoted by LN (listen) in the figure). This is needed to synchronize to the channel and determine if there is an ongoing data transmission in the FHC or not. If there is an ongoing data transmission, as in the example, then B must wait until this transmission is over (and observe the guard space, GS) before sending an RTS. In the figure, node D also wants to send to node C, and after colliding with B at the first RTS transmission, it wins the contention in the second stage. B notices this when it hears the RTS from node D and waits until this data transmission is over. For this period of time, it switches back to its home FHC (transition T2). To determine when it can try again with a new RTS, node B uses its estimate of the length of the data transmission given in the RTS packet (this information is also given in CTS packets). Node B switches back to FHC 2 (transition T3) such that it spends the period of LN before its backoff counter reaches zero. In the figure, node D wins the contention once again, and B switches back to FHC 1 (transition T4). In the meantime, node A initiates a data transmission to node B which is unsuccessful because node B is away at that time. The RTS is retransmitted later, and the subsequent data transmission is started to node B. This delays node B switching to FHC 2 once again. However, when the transmission in FHC 1 is over, node B can immediately switch to FHC 2 (transition T5). After a period of LN has passed and FHC 2 is sensed free, node B sends its RTS which is successfully received this time, allowing the consequent data packet transmission. Once this is over, node B switches back to its home FHC (transition T6).

The address and home FHC of a neighbouring node is known from a neighbour discovery mechanism. This is either based on a static configuration, or on beacon packets sent by the nodes. Beacon packets can be sent at a dedicated frequency, or on a special frequency hopping sequence. In addition, beacon packets are sent on each FHC in order to synchronize the channel timing [8]. Note that while it is clear that we must ensure timing synchronization between two nodes that communicate with each other, MFHC does not require a network-wide synchronization mechanism. Here we do not consider the synchronization mechanism in detail, but we will consider the overhead of beacon packets used for channel synchronization in the analysis of Section IV.

Since MFHC makes it possible for the nodes to communicate with all neighbours within radio range in a connection-less fashion, it is possible to apply any of the MANET routing protocols [10] to extend connectivity over multiple hops. In this case, we use beacon packets to keep track of the neighbours. One issue that needs special attention in this case is that of broadcasts. Since MFHC uses multiple channels, a broadcast packet needs to be transmitted separately to neighbours on different channels.

IV. Analysis of FHC Configurations

We now investigate the question of selecting the FHCs so that the performance of the communication is maximized. For this, we introduce three FHC configurations and compare their performance based on a simple analytical model. To enable the analysis, we first introduce a model for the contention mechanism. This will be followed by a system model that will be used for the subsequent performance comparison. In the comparison of this section we concentrate on the FHC configurations and simplify the details of the backoff mechanism, packet types and local retransmissions. Later in Section V we repeat and elaborate the analysis based on simulations of an implementation of the architecture.

IV.A. Modeling of Contention

For our analysis we use a very simple performance model of the contention mechanism that captures the impact of the number of competing nodes on the time needed to resolve

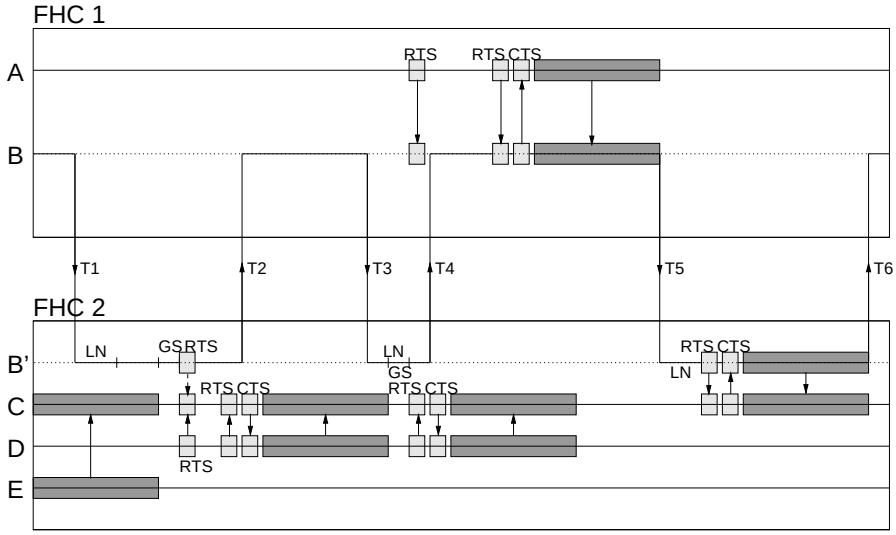

Figure 3: Example of Multiple Frequency Hopping Channel communication

contention. In [1] and [2] the authors aim at modelling the behaviour of the IEEE 802.11 contention mechanism as closely as possible. Here we neglect most of the details and make some additional simplifying assumptions so that our performance model remains analytically tractable even in a multi-channel environment.

In our model, there is one round of contention in each slot. This means that we assume that the nodes sending the RTS packets get immediate feedback on the success or failure of the contention, and we neglect the possible loss and associated delay with the CTS packet. We also assume that each contending node is aware of an ongoing transmission on the channel and does not attempt to send an RTS during this period. Therefore in this model we consider only the slots when contention takes place. A node either sends an RTS in a slot, or defers sending its RTS, depending on whether its backoff counter has reached zero or not.

It has been observed [1, 2, 18] that the initial value of the contention window, CW_{min} may impact the overhead of the contention and its optimal value is dependent on the number of competing nodes. In our analysis we do not consider the question of setting the CW_{min} constant, instead we use the optimal setting of the contention window based on the number of contenders. (Note that this issue is considered in detail in [2] where an adaptation mechanism is proposed and it is shown that the performance of the adaptive setting is close to the optimal settings.)

Let the number of contending devices be denoted by k, and let the size of the contention window at each node be constant W. Using a simple Markovian model, it can be shown [2] that a single node transmits an RTS in a given slot with a probability of $\tau = 2/(W+1)$. In a given slot a new packet transmission is initiated when exactly one node transmits an RTS. Assuming independence between nodes, its probability is

$$P_{tx} = k(1-\tau)^{k-1}\tau(1-p). \tag{1}$$

where p is the probability of non-collision error (interference, fading, noise). Our purpose here is to find the value of τ (and W) that maximizes P_{tx}. Taking the derivative of the expression and solving it for zero, we get

$$\tau = \frac{1}{k} \tag{2}$$

and consequently $W = 2k-1$. The probability of successful RTS transmission is then $P_{tx} = (1-1/k)^{k-1}(1-p)$. It is well known that the first factor in the formula goes to $1/e$ as k increases. We can thus approximate the probability as

$$P^*_{tx} = \frac{1}{e}(1-p) \tag{3}$$

In Figure 4 we show the probability of successful contention in a slot as a function of the number of competing nodes. The figure shows the results of simulation of the exponential backoff procedure with a fixed value of $CW_{min} = 8$, the value of P_{tx} for the optimal case with constant window as computed above, and the approximation P^*_{tx}. (In this case, $p = 0$ was used.) The simulated performance curve shows a slight increase which is due to the fixed initial contention window setting: when the number of contenders grows, the initial suboptimal setting no longer limits the performance. The optimal window performance P_{tx} gives an upper bound that tends to the simulated values of the backoff procedure as the number of nodes increase, similarly to the approximation P^*_{tx}.

In the following, we will use the approximation P^*_{tx} since it is close to the simulated backoff results especially as the number of nodes increases, and it gives an analytically tractable approximation which is independent of the implementation details and parameters of the contention procedure. (For simplicity, we will extend this approximation even for the $k = 1$ case.) With this approximation, the average number of slots it takes until one of the k nodes wins the contention is therefore

$$C = \frac{1}{P^*_{tx}} = \frac{e}{1-p}. \tag{4}$$

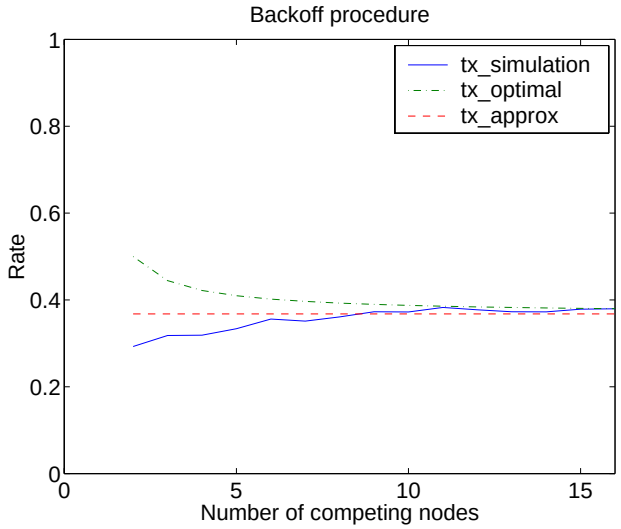

Figure 4: Simulated and analytical performance of contention

IV.B. System Model

To model system performance, we introduce a network and traffic model, and compare a number of FHC configurations. Our primary performance metric will be the total system throughput. We will compare the throughput performance of three different FHC configurations.

To model a number of groups of devices using a common application over the same coverage area in an ad hoc networking scenario, we use a group-based traffic model: devices send most of their data to other members of the same group. The total of N nodes are divided into groups of size G. In our numerical analysis, we consider the extreme case where nodes within a group send packets to the members of the same group only. (Later in Section V we will investigate the effect of inter-group communication.) Sources are assumed to be greedy, which means that sources always have a packet to send. Before each packet transmission, the destination is chosen randomly and independently according to a uniform distribution from the other nodes in the same group. Each of the N nodes are within transmission range of each other, so transmissions in different groups at the same time and same frequency collide.

We assume that transmission errors can be detected by an ARQ (Automatic Repeat reQuest) protocol but the details of this protocol are not considered here. In the analysis we assume that there exists a segmentation and reassembly mechanism, and the ARQ protocol retransmits the errored segments only. Therefore we model the additional load caused by retransmissions through the increase of the packet length by a factor of $1/(1-p)$, since $1-p$ is the success rate of data segment transmissions, where p is the transmission failure probability for a segment. (In the three configurations that we consider below, we will denote this probability by p_c, p_g and p_d.) We also assume segments of one slots in length, where a slot corresponds to the time the channel remains at one frequency hop (similarly to the Bluetooth system [3]). Furthermore we have an RTS packet that also takes one slot and we approximate the non-collision error probability of sending an RTS packet with that of sending a segment of one slot in length (that is, p). These assumptions are not essential to the MFHC proposal, and are used to facilitate the analysis in a potential application scenario.

We distinguish three different FHC configurations based on the set of nodes that have a common home FHC. In the *common* FHC case the same single channel is used by all of the N nodes. This will be our reference case where devices do not need to switch to a different FHC. In the *device* FHC case there is a separate FHC for contention and data transfer for each device. In this case, for each destination a node has to switch to a new FHC. The third FHC configuration that we investigate represents a compromise between the two extremes. In the *group* FHC case, every group of G nodes has its own FHC for contention and data transmission. Since in our traffic model of this section packets are sent only within the group, therefore nodes do not have to switch to a different FHC in this case, either.

Figures 5 - 7 illustrate the three cases. The dark rectangles represent the data packets sent on a given hopping channel, while the lightly shaded rectangles represent contention periods, with the arrows showing the direction of the data transmission and the contention. The winner of the contention is marked by a solid arrow, while other contenders are marked with a dashed arrow. Note that a device may simultaneously compete to transmit to other nodes while receive RTS packets. This is achieved by switching between transmitting an RTS and receiving, as illustrated in the example of Figure 3.

In the following we will analyze each of these configurations separately.

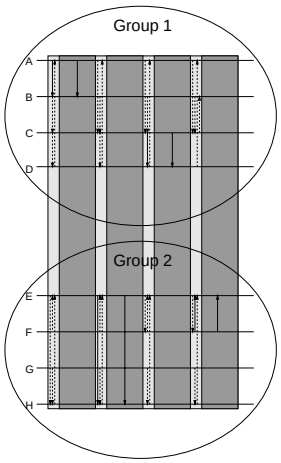

Figure 5: *Common* FHC: the same channel is used by all of the nodes.

IV.C. Common FHC

In the common FHC case each device communicates on the same single frequency hopping channel (Figure 5). The channel is occupied by alternating transmission and contention periods.

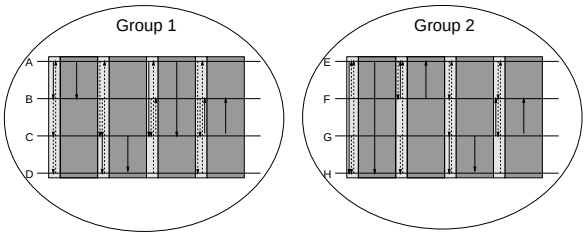

Figure 6: *Group* FHC: every group has its own channel.

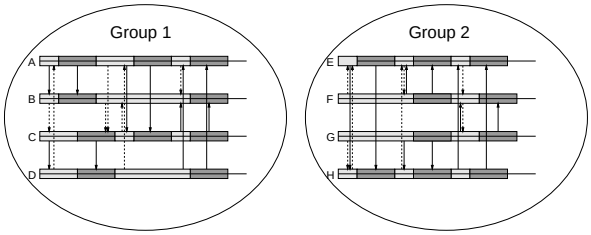

Figure 7: *Device* FHC: there is a separate channel for each of the nodes.

To find an approximation for the system throughput, we have to consider the length of the data transmissions and the length of the contention periods. The length of the data transmissions is taken to be constant L_0. To find an approximation for the time spent with contention, we use the results of Section IV.A. The approximate average time until one of the nodes wins the contention is $C_c = e/(1-p_c)$ where p_c is the non-collision error probability when sending an RTS packet. In this analysis we only consider errors caused by interference, but in the *common* FHC case there is only one channel and therefore in this model we have $p_c = 0$.

From this, the load on the common frequency hopping channel (i.e., the fraction of time spent with packet transmission, including retransmissions) is

$$\Lambda_c = \frac{L_0}{L_0 + e}. \tag{5}$$

The traffic offered to the channel by a single node (i.e., the fraction of time spent with packet transmission, including retransmissions) is therefore:

$$\lambda_c = \frac{\Lambda_c}{N} = \frac{1}{N}\left(\frac{L_0}{L_0 + e}\right). \tag{6}$$

To find the total throughput, we also take into account that the channel synchronization must be maintained. This requires the exchange of packets that consume overhead. Here we consider that synchronization is maintained by the transmission of special single-slot beacon packets with a base period of T_b slots. This decreases the capacity of the channel by a factor of $1 - 1/T_b$. The total throughput is then

$$\Theta_c = \Lambda_c(1 - 1/T_b) \tag{7}$$

measured in the unit of the capacity of a single frequency hopping channel.

IV.D. Group FHC

The group FHC case is characterized by each group of G devices using a common frequency hopping channel for both contention resolution and data transmission (Figure 6).

We use a similar approach to find an approximation for the system throughput as in the previous subsection. We have to characterize both the length of the data transmissions and the length of the contention periods. To characterize the length of the data transmissions, we assume that each packet transmission takes L slots, where the amount of data transmitted corresponds to a constant L_0 slots.

$L > L_0$ because there may be errors on the channel causing retransmissions, making the transmission of a complete packet longer. We only consider the errors caused by interference and use an independent and identically distributed error model with a segment error probability of p_g. The extra load caused by the retransmissions are approximated as $L = L_0/(1-p_g)$ (as described in Section IV.B).

To find an approximation for the time spent with contention, we use $C_g = e/(1-p_g)$ where p_g is also the probability that an RTS is lost due to non-collision error (interference in our model). The load on a single FHC can then be computed as:

$$\Lambda_g = \frac{L}{L + C_g} = \frac{L_0}{L_0 + e}. \tag{8}$$

The traffic offered by a single node then becomes

$$\lambda_g = \frac{\Lambda_g}{G} = \frac{1}{G}\left(\frac{L_0}{L_0 + e}\right). \tag{9}$$

We now approximate the probability of interference error, p_g. A single frequency hopping channel is disturbed by $N/G - 1$ other similar channels. Each channel hops on K different carriers independently in a pseudo-random manner. Since the channels are not synchronized with each other, a transmission in a single slot in one channel may disturb two slots in a different channel if the carriers collide. If we neglect the interference caused by RTS and CTS packets, the probability that a transmission of one slot is successful despite the interference caused by another channel with a load of Λ_g is $1 - \Lambda_g \frac{2}{K}$. The error probability can then be approximated as

$$p_g = 1 - \left(1 - \Lambda_g \frac{2}{K}\right)^{N/G-1}. \tag{10}$$

The total throughput is obtained by summing the traffic in each channel, taking into account the synchronization overhead and that data transmission has an efficiency of $1 - p_g$ due to errors:

$$\Theta_g = \frac{N}{G}\Lambda_g(1 - p_g)(1 - 1/T_b). \tag{11}$$

IV.E. Device FHC

The device FHC is characterized by each node having a separate channel of its own (Figure 7). This means that

each time a source node sends a data packet to any destination node, the source first has to switch to the FHC of the destination.

To find the traffic offered by a single node, we approximate the average time taken with data reception as follows. Similarly as above, a single packet reception takes $L = L_0/(1-p_d)$ slots, where p_d is the segment error probability. A packet reception is preceded by a contention period. This would take on average $e/(1-p_d)$ slots in general (since there is a non-collision error with probability p_d for an RTS).

However, contention is prolonged in this case for the following reasons. While waiting for incoming RTS packets initiating a data transfer, each node also has a packet to send at the same time. This means that a node has to switch between its own frequency hopping channel and that of the frequency hopping channel of its destination, as described in Section III. The node participates in two contentions simultaneously, once as a potential transmitter and once as a potential receiver. Even if this could be done with 100% efficiency, this would double the time of the average contention. However, switching between the channels necessarily implies inefficiency. In addition, the contention window of source nodes are increased due to the fact that a destination node does not respond to an RTS when it has switched to a different FHC. The extent of these effects depends on many implementation dependent factors, such as the time needed to switch to a different FHC, and the setting of the maximum contention window. We approximate these effects by assuming that contention is prolonged by a factor of β due to the inefficiency incurred by switching between different channels. Our approximation for the contention period is therefore $C_d = \beta e/(1-p_d)$. We have $\beta > 2$ since the length of contention is at least doubled. (We will investigate the value of β through simulations in Section V.)

Due to symmetry between nodes and roles, each node spends the same amount of time with transmitting and receiving, and consequently transmits on average one packet for each packet reception. This follows that the fraction of time spent with reception at a given node is

$$\lambda_d = \frac{L}{2L + C_d} = \frac{L_0}{2L_0 + \beta e}. \tag{12}$$

Due to symmetry of the traffic model, λ_d is also the time spent with transmission by a given node.

Similarly to the previous subsection, the segment error probability can be found:

$$p_d = 1 - \left(1 - \lambda_d \frac{2}{K}\right)^{N-1}. \tag{13}$$

To find the total throughput, we have to take into account the synchronization overhead. Each node in a group has to synchronize to all other nodes in the group, giving a factor of $1 - G/T_b$. We can write the total system throughput as

$$\Theta_d = N\lambda_d(1-p_d)(1-G/T_b). \tag{14}$$

IV.F. Performance Comparison

We now evaluate the performance of the FHC configurations based on the analytical model of the previous subsections. First, we plot the total throughput as a function of the total number of nodes N, see Figure 8. In the figure we use tentative parameter settings: the group size was fixed at 10 nodes, packet length was 12 slots, number of hop frequencies was set to 79, we used $\beta = 4$, and the synchronization overhead was not included. In the upper left box, we plot the offered traffic by a single node, that is, the fraction of time spent with data transmission by a node. First of all we can observe that this is constant for the *group* and *device* FHC configurations. To explain this, notice that the groups are logically independent and do not depend on each other except for the interference. The increase in the interference is shown in the upper right. Interference causes data transmissions to be longer, but it also prolongs the contention period by the same factor explaining why the offered traffic remains constant. (Note that in a given implementation the effect of packet losses may cause a different factor of increase for the data transmission time and for the contention. This may result in slight changes in the offered traffic as will be visible in the simulation results of the next section.) In the *common* FHC case, nodes share the same channel which causes the per node offered traffic to decrease as the number of nodes increases.

The figure in the lower left box shows the total throughput of the system measured in the unit of the capacity of a single frequency hopping channel. This is constant for the *common* FHC case since the total capacity of a single channel is used, and it is not affected by interference. In the *device* and *group* cases, the total throughput increases with increasing number of nodes. This is because the number of FHCs are increased providing multiplexing gain. The slope of the curves decreases though because of the increased interference. The *device* FHC case allows for a greater number of parallel data transmissions to be multiplexed than the *group* FHC case which allows only a single transmission per group. This explains the significantly higher total throughput of the *device* FHC configuration.

Also plotted in the lower right box is a measure of the spectral efficiency. We obtained this measure by dividing the total throughput by the number of hop carriers, K. If all carriers were continuously transferring data, this measure would yield 1; its value therefore represents the efficiency of utilizing the available spectrum. We made an exception in the *common* FHC case, where we did not divide the total throughput by K, since only a single common channel is used in this case, which could - hypothetically - span even the whole available spectrum without causing any interference. In the rest of the cases, this is not possible since many channels need to be multiplexed that could interfere with each other.

The results show that the spectral efficiency is highest in the *common* FHC case, and it is lower for the other configurations. This observation can be interpreted as follows. If a single common high-speed channel can be used by all devices on an on-demand time-division basis, it can give a much more efficient usage of the available spec-

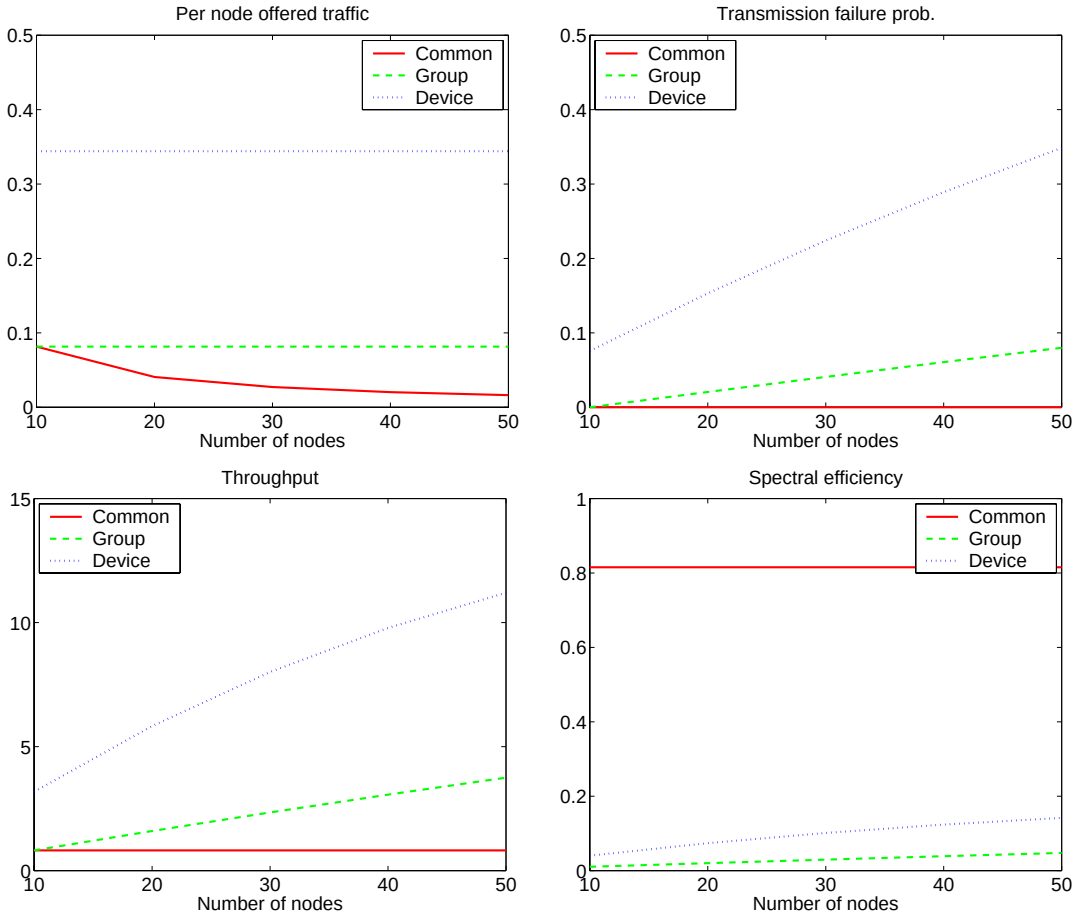

Figure 8: System performance, $G = 10, L_0 = 12, T_b = \infty$

trum than dividing it into many uncoordinated low-speed channels. We have to keep this in mind when considering the other configurations employing multiple uncoordinated frequency hopping channels. However, a high-speed common channel may be difficult or costly to realize in practice. Note also that frequency hopping radios naturally lead to the use of multiple channels rather than a common channel of higher bandwidth. A full comparison of these two cases, involving other aspects such as hardware limitations, cost, radio propagation and error characteristics, is out of the scope of this paper.

Figure 9 shows the dependence of the traffic offered by a node on the group size and packet length, with only a single group present. (Note that the offered traffic determines the total throughput.) We can see that the *device* FHC case offers a constant per node offered traffic, while the *group* and *common* FHC cases (which are identical in this scenario) yield a decreasing per node offered traffic. The reason for this is that the *device* FHC configuration allows multiplexing of data transmissions within a group. To compare the two curves at $G = 2$, notice that we have $\beta > 2$, which follows that $\lambda_d < \lambda_g$. This means that for a group of two nodes, the *device* FHC is necessarily less efficient than the *group* FHC. Depending on the implementation dependent value of β, the two curves must intersect each other, marking the group size where the *device* FHC configuration is equally efficient as the *group* FHC. By comparing the two graphs for long and short packet size, we can observe that the intersection point is also dependent on the packet size. When packets are shorter, the effect of backoff overhead is increased, therefore the per node offered traffic (and the total throughput) is lower.

Figure 10 shows the dependence of the total throughput on the synchronization overhead. This overhead depends on the accuracy of the clocks that are used: the less accurate they are, the more frequently we need to send beacon packets to keep the synchronization. The figures plot the base beacon sending period. The figures show that the *device* FHC case is the most sensitive to synchronization overhead, especially for higher group size. This is attributed to the fact that in this case, a node in a group has to synchronize to all other nodes in its group to be able to send data, while in the other cases nodes have to synchronize to one channel only. Note also that we took a very conservative computation for the synchronization overhead, since only a singe slot was wasted for a beacon packet. In a practical implementation, however, this overhead might be much higher, which further emphasizes that the *device* FHC configuration is very sensitive to accurate synchronization.

V. Simulation Study

To investigate the performance of the implementation of different FHC configurations, we have implemented the

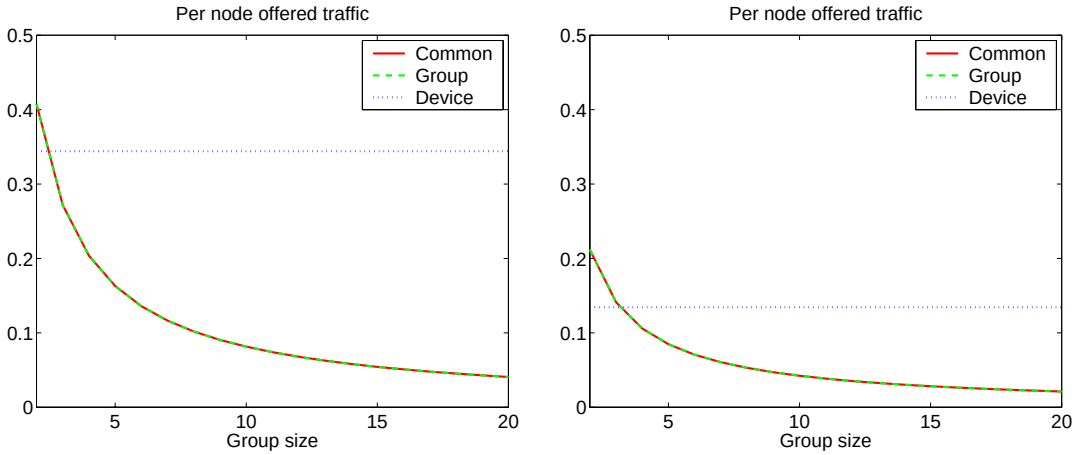

Figure 9: Dependence of the per node offered traffic on the group size and packet length. $N = G, T_b = \infty$. On the left $L_0 = 12$, on the right $L_0 = 2$.

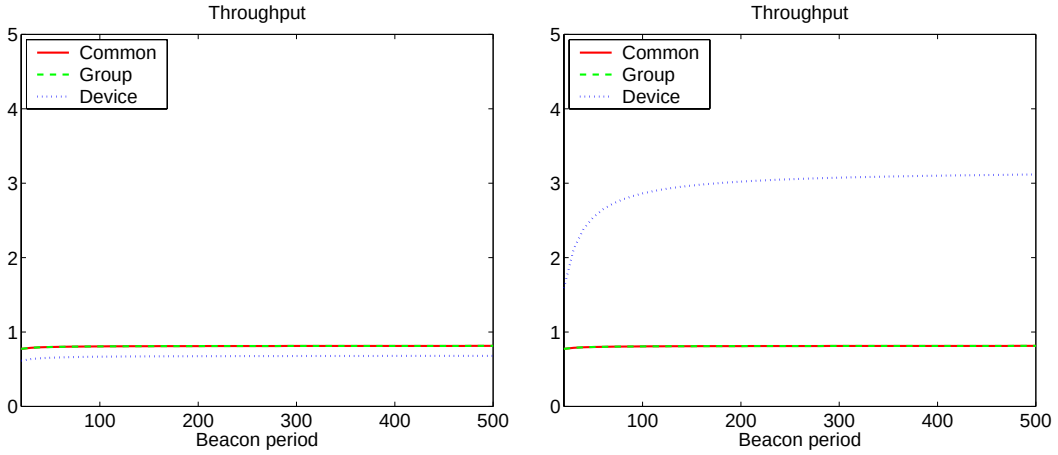

Figure 10: Dependence of system throughput on the beacon period and group size. $N = G, L_0 = 12$. On the left, $G = 2$, on the right, $G = 10$.

MFHC scheme in a packet level simulator [7]. Figure 11 shows the architecture of the simulator. The physical layer consists of a packet collision detector which determines the reception status of every individual packet. Each node has an associated FHC object in the physical layer (this association is shown by the dashed lines). The link layer representation of each node connects to exactly one FHC in the physical layer at a time, the one that it follows at the given moment as determined by the MFHC protocol implementation in the link layer (this connection is shown by the solid lines).

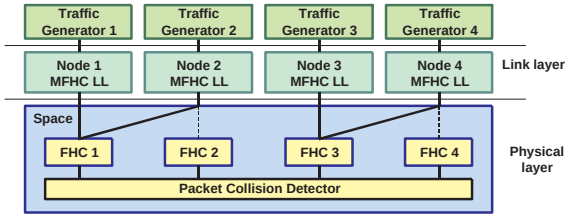

Figure 11: Simulator architecture

We consider scenarios where all nodes are within radio range of each other, which represents the worst case in terms of interference. In the physical layer model, packets can be lost due to interference (i.e., two or more packets are sent on the same frequency at the same time) or collision (i.e., an RTS packet collides with another packet sent to the same destination), otherwise they are delivered correctly. In the link layer, we model the contention mechanism as described in Section III. FHCs are independent of each other using a pseudo-random frequency hopping pattern. We have implemented a segmentation and reassembly mechanism, and an ARQ protocol that gives feedback on the reception status after each segment. Lost segments are retransmitted immediately. Packets can be sent at the beginning of a slot. The slot timing is aligned to frequency hopping: there is a guard time at the end of each slot to allow devices to tune to a new frequency. Table 2 lists the parameters used in the simulations. Note that the channel capacity and the number of hop frequencies were selected to reflect the constraints of the 2.4GHz ISM band, and the other parameters were selected to reflect the current capabilities of typical Bluetooth implementations.

Channel capacity	1 Mbit/sec
Slot length	1 ms
Hop frequencies	79
Segment length	1, 2, 3, 4 or 5 slots
Segment header length	164 bits
Guard time for frequency hopping	0.1 ms per slot
Length of RTS, CTS, ACK packets	1 slot
Minimal contention window	8 slots
Maximal contention window	64 slots
Synchronization overhead	0 (not considered)
Listen time on new FHC	6 slots

Table 2: Simulation parameters

First we investigate the extent of multiplexing that can be achieved by using the *group* and *device* FHC configurations. Figure 12 shows the per node offered traffic and total system throughput as the number of groups, each with ten member nodes, are increased. The results are in accordance with the analysis of Section IV, showing that the *device* FHC configuration increases the total system throughput by a factor of two. However, the multiplexing gain that is achieved by the *device* FHC is only present with large group sizes. Figure 13 shows that with a group size of two nodes, the *device* FHC case actually performs worse by about 30%, because it is less efficient in contention and can not make use of multiplexing.

Figure 14 investigates the differences between small and large groups. The results follow the same trend as the analytical model shown in Figure 9. Fitting the formulas of Section IV to the simulation results, we can approximate the value of β, which shows the inefficiency of contention in the *device* FHC case. We get a value of $\beta = 16$ in the case of 1500 byte packets and $\beta = 8$ in the case of 250 byte packets. These values are much greater than the minimal value of 2, showing that the switching of FHC during contention introduces a significant amount of extra overhead. Note also that this factor is not constant: in the case of long (1500 byte) packets and minimal group size of 2 nodes, the *device* FHC becomes more efficient (β decreases to 8 in this case). When there are only two nodes in the group, the intended destination node does not communicate with other nodes. That would cause failed RTS attempts whose number is much bigger in the case of long packets, which explains why we see this effect to a much greater extent in the case of long packets.

So far we have allowed traffic only within a group in our model. We now extend our traffic model to investigate the effect of traffic between the groups as well. In the extended model, with a probability p_{ng} a nodes chooses its destination from all the other nodes in the network, not just its own group. Figure 15 shows the throughput performance as a function of the probability p_{ng} which determines the non-group traffic. The *device* FHC configuration is not sensitive to this change since it does not depend on the formation of groups. It only shows a slight throughput decrease in the case when the group size is two, which is explained by the reasoning above in the previous paragraph. The *group* FHC shows a decrease in both small and large group sizes, but the decrease is much more significant when the group size is small. The reason for this is that when the group size is high, there is a higher chance that at least one of the potential destinations is available, and so the channel can be utilized. In both cases, the results show that the *device* FHC configuration gives higher performance in the case of heterogeneous traffic, that is, when there is significant traffic between the groups.

Finally we observe the effect of changing the traffic pattern within a group to model a client-server application (with no traffic between the groups). In this case we designate one node in all groups to be a server and the other nodes in the group to be clients. All nodes remain greedy as before in that they always have a data to send, but with a constant probability p_s, the clients choose the server as their destination. Figure 16 shows the total throughput as the constant p_s is increased from 0 to 1 (server-client traffic only). In this experiment the total throughput of the *group* FHC configuration remains constant since this is determined by the capacity of the group channel. On the other hand the performance of the *device* FHC configuration decreases to that below the *group* case. When there is only server-client traffic, the *device* FHC case can not achieve multiplexing gain, and it uses a less efficient contention scheme than the *group* FHC which explains its lower performance.

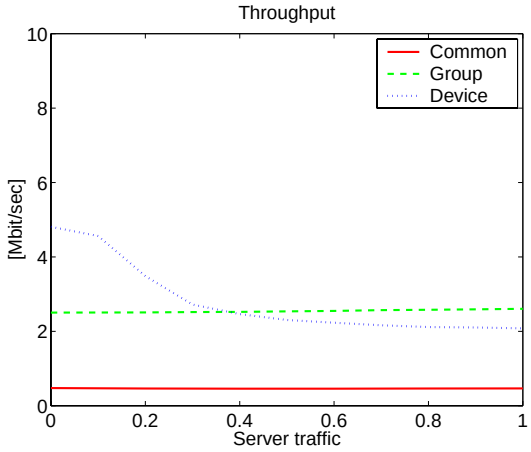

Figure 16: The effect of server-client traffic on the total throughput. Group size is fixed at 10, number of nodes is 50.

VI. Conclusion

We have proposed Multiple Frequency Hopping Channel communication (MFHC), a scheme that forms a connected ad hoc PAN from multiple frequency hopping channels. Our scheme relies on the notion of home FHC. Each device participating in an ad hoc network has a home FHC which determines the frequency hopping scheme it follows whenever it is not transmitting at another FHC. To transmit to a particular device, it is necessary to switch to that particular device's home FHC, listen to the channel and resolve contention. The difference from a traditional random

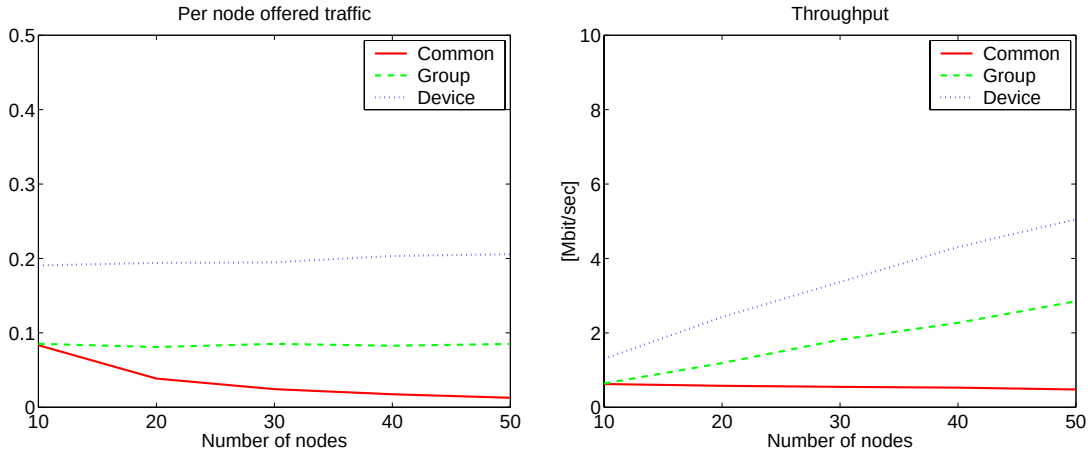

Figure 12: Per node offered traffic and total throughput as the number of groups are increased. Group size is fixed at 10.

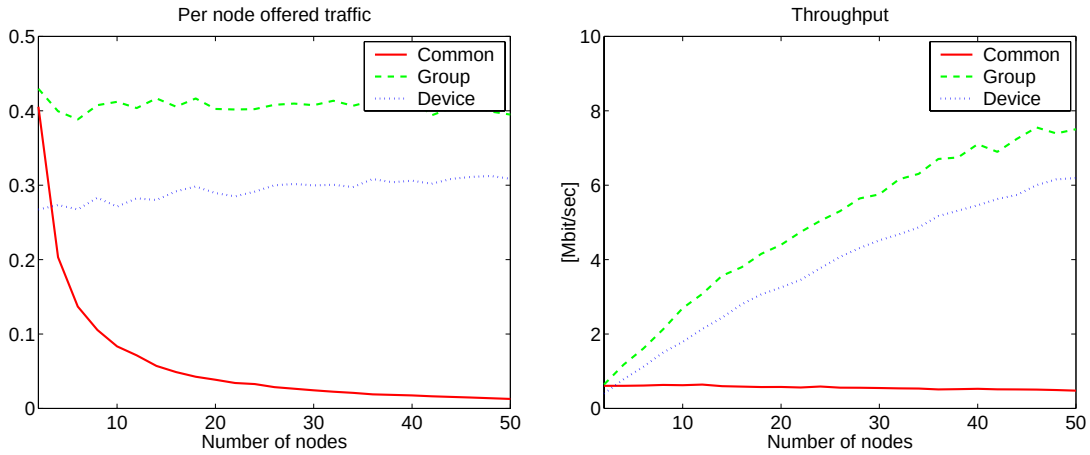

Figure 13: Per node offered traffic and total throughput as the number of groups are increased. Group size is fixed at 2.

access scheme is that besides the possibilities of success or collision, a third option is that the destination is "away" at another FHC.

Besides allowing ad hoc PANs to benefit from the advantages of frequency hopping, this scheme increases their throughput compared to using a single channel only, but it requires additional coordination. We have investigated the impact of this additional coordination on the system's performance using analytical and simulation tools. In particular, we have compared the extreme case of MFHC, where each device has its own distinct FHC, to a reference case where the entire ad hoc network uses the same single FHC. The results show that the former case (*device* FHC) provides significantly higher total throughput than the reference case (*common* FHC).

We have also analyzed a case where subsets of an ad hoc network form a partially closed communication group in the sense that members of one group communicate mostly with other members of the same group and rarely with other nodes of the ad hoc network. This scenario may be typical in some realistic ad hoc networks. We have shown how MFHC can adapt to this case such that members of one group share the same home FHC. This case, referred to as *group* FHC, represents a compromise between the *device* FHC and *common* FHC cases. We have shown that it is especially well suited to server-client type traffic patterns, but it is ill-suited for heterogeneous traffic patterns. The *group* FHC configuration makes the contention mechanism more efficient and it requires less overhead for channel synchronization, in exchange for lower multiplexing gain and consequently lower total throughput.

We have also raised a number of issues that we intend to investigate in detail as a continuation of the work presented here. These issues include the analysis of neighbour discovery and synchronization mechanisms, a dynamic FHC selection algorithm that optimizes the performance of the network, and an analysis of the multihop networking scenario.

References

[1] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, March 2000.

[2] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In *Proc. PIMRC*, Oct 1996.

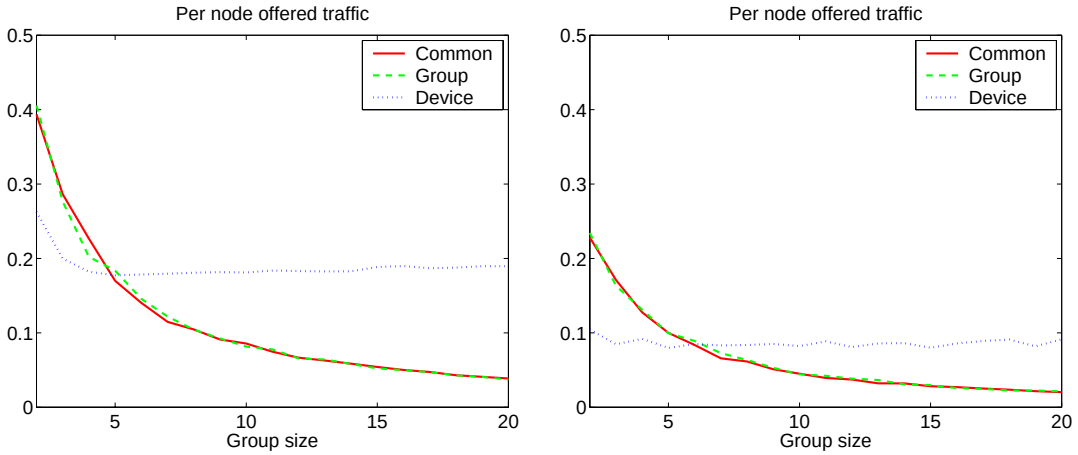

Figure 14: Per node offered traffic in the case of a single group. On the left packet length is 1500 byte, on the right packet length is 250 byte.

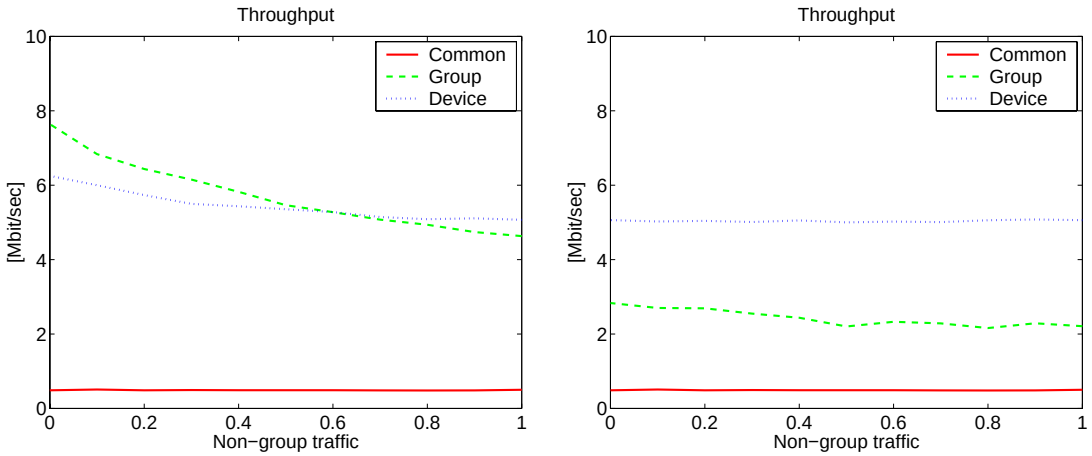

Figure 15: The effect of non-group traffic on the total throughput. On the left group size is 2, on the right group size is 10. The number of nodes is fixed at 50.

[3] Bluetooth specification 1.1.

[4] Bluetooth SIG PAN working group.

[5] A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 1987.

[6] M. Hännikäinen, T. D. Hämäläinen, M. Niemi, and J. Saarinen. Trends in personal wireless data communications. *Computer Communications*, 25(1), 2002.

[7] Z. Haraszti, I. Dahlquist, A. Faragó, and T. Henk. PLASMA - an integrated tool for ATM network operation. In *International Switching Symposium ISS '95*, Berlin, Germany, April 1995.

[8] IEEE 802.11 family of standards.

[9] C. Law, A. K. Mehta, and K.-Y. Siu. Performance of a new Bluetooth scatternet formation protocol. In *Mobihoc*, 2001.

[10] J. Macker and S. C. (chairmen). MANET (Mobile Ad Hoc Networking) working group of the IETF.

[11] G. Miklós, A. Rácz, Z. Turányi, A. Valkó, and P. Johansson. Performance aspects of Bluetooth scatternet formation. In *Proceedings of MobiHOC*, August 2000.

[12] A. Pursley. The role of spread spectrum in packet radio networks. *Proceedings of the IEEE*, 75(1), 1987.

[13] A. Rácz, G. Miklós, F. Kubinszky, and A. Valkó. A pseudo random coordinated scheduling algorithm for Bluetooth scatternets. In *Proceedings of MobiHOC*, 2001.

[14] M. A. Rónai and E. Kail. A simple neighbour discovery procedure for Bluetooth ad hoc networks. In *Proceedings of GlobeCom*, Dec 2003.

[15] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Distributed topology construction of Bluetooth personal area networks. In *Proceedings of Infocom*, 2001.

[16] Z. Tang and J. J. Garcia-Luna-Aceves. Hop-reservation multiple access for multichannel packet

radio networks. *Computer Communications*, 23(10), 2000.

[17] G. V. Záruba, S. Basagni, and I. Chlamtac. Bluetrees – scatternet formation to enable Bluetooth-based ad hoc networks. In *Proceedings of ICC*, 2001.

[18] E. Ziouva and T. Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. *Computer Communications*, 25(3), 2002.

[19] S. Zürbes, W. Stahl, K. Matheus, and J. Haartsen. Radio network performance of Bluetooth. In *Proceedings of ICC*, 2000.