A Multi-Frequency MAC Specially Designed for Wireless Sensor Network Applications

GANG ZHOU[*], YAFENG WU[†], TING YAN[‡], TIAN HE[§], CHENGDU HUANG[††], JOHN A. STANKOVIC[†], TAREK F. ABDELZAHER[§§]

[*]Computer Science Department, College of William and Mary

[†]Computer Science Department, University of Virginia

[‡]Eaton Innovation Center

[§]Computer Science Department, University of Minnesota

[††]Google

[§§]Computer Science Department, University of Illinois at Urbana-Champaign

Multi-frequency media access control has been well understood in general wireless ad hoc networks, while in wireless sensor networks, researchers still focus on single frequency solutions. In wireless sensor networks, each device is typically equipped with a single radio transceiver and applications adopt much smaller packet sizes compared to those in general wireless ad hoc networks. Hence, the multi-frequency MAC protocols proposed for general wireless ad hoc networks are not suitable for wireless sensor network applications, which we further demonstrate through our simulation experiments. In this paper, we propose MMSN, which takes advantage of multi-frequency availability while, at the same time, takes into consideration the restrictions of wireless sensor networks. In MMSN, four frequency assignment options are provided to meet different application requirements. A scalable media access is designed with efficient broadcast support. Also, an optimal non-uniform backoff algorithm is derived and its lightweight approximation is implemented in MMSN, which significantly reduces congestion in the time synchronized media access design. Through extensive experiments, MMSN exhibits the prominent ability to utilize parallel transmissions among neighboring nodes. When multiple physical frequencies are available, it also achieves increased energy efficiency, demonstrating the ability to work against radio interference and the tolerance to a wide range of measured time synchronization errors.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Wireless sensor networks, media access control, multi-channel, radio interference, time synchronization

1. INTRODUCTION

As a new technology, wireless sensor networks (WSNs) has a wide range of applications [Culler et al. 2004] [Estrin et al. 1999] [Akyildiz et al. 2002], including environment monitoring, smart buildings, medical care, industrial and military applications. Among them, a recent trend is to develop commercial sensor networks that require pervasive sensing of both environment and human beings, e.g., assisted living [Ganti et al. 2006] [Harvard CodeBlue 2008] [MIThril 2008] and smart homes [UVA Smart House 2008] [Kidd et al. 1999] [UFL Smart House 2008]. For these applications, sensor devices are incorporated into human cloths [Natarajan et al.

2007] [Zhou et al. 2008] [Shah et al. 2008] [Li et al. 2009] for monitoring health related information like EKG readings, fall detection, and voice recognition. While collecting all these multimedia information [Akyildiz et al. 2007] requires a high network throughput, off-the-shelf sensor devices only provide very limited bandwidth in a single channel: 19.2Kbps in MICA2 [Hill et al. 2000] and 250Kbps in MICAz [CROSSBOW 2008] and Telos [Polastre et al. 2005]. A potential solution for achieving improved network throughput with existing hardware is to utilize multiple frequencies for parallel communication. Unfortunately, even though multiple frequencies are available in existing hardware like MICAz and Telos, mainstream media access control (MAC) protocols in sensor network literature still focus on single frequency solutions [Ahn et al. 2006] [Buettner et al. 2006] [Ye et al. 2006] [Klues et al. 2007] [Polastre et al. 2004] [Ye et al. 2002] [Rajendran et al. 2003] [Dam and k. Langendoen 2003] [Woo and Culler 2001] [El-Hoiyi et al. 2004]. Therefore, it is imperative to design multi-frequency MAC protocols to improve network throughput for emerging human related sensor applications.

When it comes to general wireless ad hoc networks, a significant number of multi-frequency MAC protocols have been proposed in literature. However, these protocols are not suitable for typical WSN applications. First, to save energy and reduce product size and cost, each sensor device is usually equipped with a single radio transceiver. This single transceiver can not transmit and receive at the same time, nor can it function on different frequencies simultaneously. This restricted hardware is quite different from more powerful hardware assumed in other wireless systems. For example, protocols [Tang and Garcia-Luna-Aceves 1999] [Tzamaloukas and Garcia-Luna-Aceves 2000] are designed for frequency hopping spread spectrum (FHSS) wireless cards, and protocol [Deng and Haas 1998] assumes the busy-tone functionality on the hardware. In protocols [Nasipuri et al. 1999] [Wu et al. 2000] [Nasipuri and Das 2000] [Caccaco et al. 2002], the hardware is assumed to have the ability to listen to multiple frequencies at the same time. Second, the network bandwidth in WSNs is very limited and the MAC layer packet size is very small, 30∼50 bytes, compared to 512+ bytes used in general wireless ad hoc networks. Due to the small data packet size, the RTS/CTS control packets no longer constitute a small overhead that can be ignored. So protocols [Bahl et al. 2004] [Fitzek et al. 2003] [Li et al. 2003] [So and Vaidya 2004] [Jain and Das 2001] [Tzamaloukas and Garcia-Luna-Aceves 2000] [Tang and Garcia-Luna-Aceves 1999] that use RTS/CTS controls [1] for frequency negotiation and reservation are not suitable for WSN applications, even though they exhibit good performance in general wireless ad hoc networks.

Second, the network bandwidth in WSNs is very limited and the MAC layer packet size is very small, 30∼50 bytes, compared to 512+ bytes used in general wireless ad hoc networks. Due to the small data packet size, the RTS/CTS control packets in IEEE 802.11 [IEEE 802.11 1999] no longer constitute a small overhead that can be ignored. So protocols [Bahl et al. 2004] [Fitzek et al. 2003] [Li et al. 2003] [So and Vaidya 2004] [Jain and Das 2001] [Tzamaloukas and Garcia-Luna-

[1] RTS/CTS controls are required to be implemented by 802.11-compliant devices. They can be used as an optional mechanism to avoid Hidden Terminal Problems in the 802.11 standard and protocols based on that like [Raniwala and Chiueh 2005] [Adya et al. 2004].

Aceves 2000] [Tang and Garcia-Luna-Aceves 1999] that use RTS/CTS for frequency negotiation are not suitable for WSN applications, even though they exhibit good performance in general wireless ad hoc networks.

In this paper, we propose MMSN, abbreviation for Multi-frequency Media access control for wireless Sensor Networks. MMSN takes full advantage of multiple frequencies and is especially designed to meet WSN requirements. The detailed MMSN design is presented from two aspects: *frequency assignment* and *media access*, and its performance is evaluated through extensive simulation. The main contributions of this work can be summarized as follows:

- To the best of our knowledge, the MMSN protocol is the first multi-frequency MAC protocol especially designed for WSNs, in which each device is equipped with a single radio transceiver and the MAC layer packet size is very small.
- Instead of using pair-wise RTS/CTS frequency negotiation [So and Vaidya 2004] [Jain and Das 2001] [Tzamaloukas and Garcia-Luna-Aceves 2000] [Tang and Garcia-Luna-Aceves 1999], we propose lightweight frequency assignments, which are good choices for many deployed comparatively static wireless sensor networks [Mainwaring et al. 2002] [Xu et al. 2004] [He et al. 2004] [Simon et al. 2004]. Even though pair-wise frequency negotiation is efficient when devices are highly mobile, it involves unnecessary overhead and is too costly, especially for comparative static wireless sensor networks.

  This paper gives a complete study of tradeoffs among physical frequency requirements, potential conflict reduction and communication overhead, during frequency assignment. Four optional frequency assignment methods are proposed in MMSN, which exhibit distinguished advantages in different scenarios.
- We develop new toggle transmission and snooping techniques to enable a single radio transceiver in a sensor device to achieve scalable performance, avoiding the non-scalable "one control channel + multiple data channels" design [Li et al. 2003]. Also, MMSN has efficient broadcast support, which either is not addressed in [So and Vaidya 2004] or is implemented by repeated link-layer retransmission of broadcast packets enqueued by higher layers in [Bahl et al. 2004].

  Moreover, through strict theoretical analysis, an optimal non-uniform backoff algorithm is derived and its lightweight approximation is implemented in MMSN. Compared with a uniform backoff algorithm, this non-uniform design significantly reduces potential conflicts among neighboring nodes.

The rest of this paper is organized as follows: In Section 2, we present the motivation of this work. In Section 3, the design details of MMSN are explained. In Section 4, extensive experiments are conducted to evaluate MMSN's performance. In Section 5, we present the related work. Finally, in Section 6, we give the conclusions.

2. MOTIVATION

To obtain a better understanding of the cost that RTS/CTS control packets incur to multi-frequency protocols in general wireless ad hoc networks versus WSNs, we choose a typical multi-frequency MAC protocol, the MMAC [So and Vaidya 2004] protocol, proposed for general wireless ad hoc networks, as a case study. In MMAC, periodically transmitted beacons divide time into fixed-length beacon intervals. At the beginning of each beacon interval, there is a small window called the ATIM window, in which the nodes that have packets to send negotiate frequencies with destination nodes. After the ATIM window, nodes that have successfully negotiated frequencies with their destinations can send out data packets using the IEEE 802.11 protocol, i.e. exchanging RTS/CTS before sending out DATA packets. We implement MMAC in GloMoSim [Zeng et al. 1998], a scalable discrete-event simulator developed by UCLA, and observe the performance. We adopt the same experimental setup as in [So and Vaidya 2004]: 100 nodes are randomly placed in a 500m×500m terrain. The transmission range of each node is 250m. Each node has 3 physical frequencies. Forty nodes are randomly chosen to be sources, and 40 nodes are randomly chosen to be destinations. Source nodes generate CBR traffic to destinations with a rate of 10 packets per second. Each simulation runs for 30 minutes in GloMoSim. Figure 1 plots the aggregate MAC throughput of the network (an averaged result of 100 runs) with different packet sizes.

As shown in Figure 1, when the packet size is large, the MMAC protocol with 3 frequencies and a beacon interval of 100ms (the default configuration suggested in [So and Vaidya 2004]) impressively enhances the aggregate MAC throughput by a factor of nearly 2 over IEEE 802.11. This result is consistent with that presented in [So and Vaidya 2004]. However, the performance of both MMAC and IEEE 802.11 degrades when the packet size reduces. This is because the overhead of using RTS/CTS packets becomes more prominent when the data packet size is small. Moreover, the performance improvement of MMAC over IEEE 802.11 diminishes when the packet size becomes smaller. When the packet size is as small as 32 bytes, IEEE 802.11 has even a slightly higher throughput than MMAC. The reason is when the packet size reduces, more packets *could* be sent in a beacon interval. However, since nodes generally can not switch frequency during a beacon interval, the bandwidth wasted is more severe compared to the case when the packet size is large. Changing the length of the beacon interval could be beneficial, but the effect is two-sided. While lengthening the beacon interval can mitigate the overhead of having a fixed period of frequency negotiation, it deteriorates the bandwidth caused by the requirement that nodes have to stick to the frequency they have negotiated with some of their neighbors. In Figure 1, we also plot the cases with different beacon intervals. We can see that while using a shorter beacon interval (50ms) helps to some extent, MMAC with 3 frequencies still can not even outperform IEEE 802.11 with a single frequency, when the packet size is as small as 64 or 32 bytes. The main observation we make here is that while MMAC is a good multi-frequency MAC protocol for general wireless ad hoc networks where packets usually have large sizes, it is not suitable for WSNs where packets are much smaller.

3. MMSN PROTOCOL

This section presents the MMSN multi-frequency MAC protocol. MMSN is especially designed for WSNs, which is composed of hundreds of simple devices geographically dispersed in an ad hoc network over a large geographic area. Each device is equipped with a single transceiver and the packet size is very small, 30∼50 bytes. The MMSN protocol consists of two aspects: frequency assignment and media

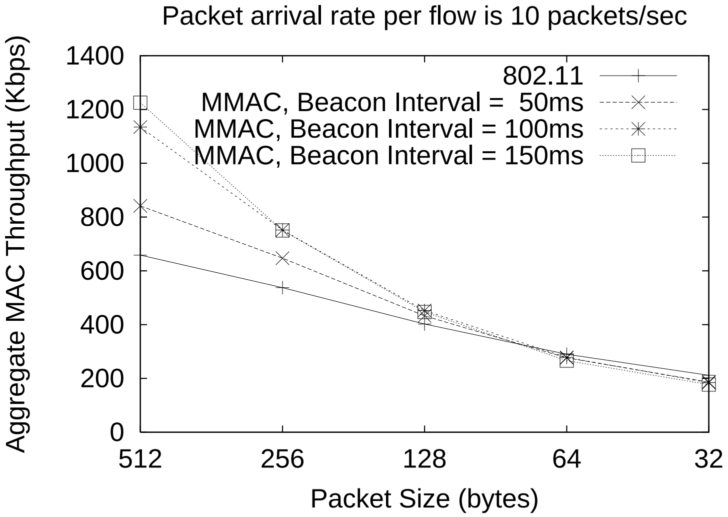


Fig. 1. Effect of Packet Size on MMAC

access. The frequency assignment is used to assign different frequencies if enough frequencies exist, or evenly allocate available frequencies if there are more neighbors than available frequencies, to nodes that have potential communication conflicts. MMSN allows users to choose 1 of 4 available frequency assignment strategies. In media access design, nodes that have potential conflicts coordinate to access the shared physical frequencies, in a distributed way.

3.1 Frequency Assignment

In frequency assignment, each node is assigned a physical frequency for data reception. The assigned frequency is broadcast to its neighbors, so that each node knows what frequency to use to transmit unicast packets to each of its neighbors. We do not adopt RTS/CTS frequency negotiation, because it involves unnecessary overhead for many deployed wireless sensor networks [Mainwaring et al. 2002] [Xu et al. 2004] [He et al. 2004] [Simon et al. 2004] where devices are generally not mobile. In WSNs, frequency assignment can either be done once at the beginning of the system deployment, or it can be done very infrequently just for adapting to system aging. In order to reduce communication interference and hence reduce hidden terminal problems [IEEE 802.11 1999], nodes within two communication hops[2] are evenly assigned available physical frequencies.

In this section, four optional frequency assignment schemes are put forth: ex-

[2] In [Zhou et al. 2005], it is pointed out that interference hops (connectivity based on interference relations), rather than communication hops, should be used for this purpose. The paper also points out that an interference hop can be longer than a communication hop, and hence it uses multiple radio sending powers to assist detecting the runtime interference relations. For simplicity, we use two communication hops in this work. All algorithms proposed here can be extended by first running the algorithm in [Zhou et al. 2005] to get the interference relations, and then replacing the two communication hops with two interference hops.

clusive frequency assignment, even selection, eavesdropping and implicit-consensus. Among these four, exclusive frequency assignment guarantees that nodes within two hops are assigned different frequencies, when the number of frequencies is equal to or greater than the node number within two hops. Implicit-consensus also provides this guarantee, with less communication overhead, but requires more physical frequencies. Even selection and eavesdropping do not provide this guarantee and are designed for use when the number of available frequencies is smaller than the node number within two hops. Among these two, even selection leads to fewer potential conflicts while eavesdropping is more energy efficient. Users of MMSN can choose any one of the four options depending on their WSN attributes. Details of these four schemes are presented in the following subsections.

3.1.1 *Exclusive Frequency Assignment.* In exclusive frequency assignment, nodes first exchange their IDs among two communication hops, so that each node knows its two-hop neighbors' IDs. A simple way to implement this is for each node to broadcast twice. In the first broadcast, each node beacons its node ID, so that each node knows its neighbors' IDs within one communication hop. In the second broadcast, each node beacons all neighbors' IDs it has collected during the first broadcast period. Hence, after the second beacon period, each node gets its neighbors' IDs within two communication hops. Currently, we do not consider radio irregularity and link asymmetry [Zhao and Govindan 2003] [Zhou et al. 2004] [Woo et al. 2003] [Cerpa et al. 2005]. Readers can refer to [Chang and Maxemchuk 1984] [Tourrilhes 1998] [Vollset and Ezhilchelvan 2003] for more information about reliability issues in broadcast.

After nodes collect ID information of all neighbors within two hops, they make frequency decisions in the increasing order of their ID values. If a node has the smallest ID among its two communication hops, it chooses the smallest frequency among available ones, and then beacons the frequency choice within two hops. If a node's ID is not the smallest one among two hops, it waits for frequency decisions from other nodes within two hops that have smaller IDs. After decisions from all those nodes are received, the node chooses the smallest available (not chosen by any of its two-hop neighbors) frequency and broadcasts this choice among two hops.

3.1.2 *Even Selection.* In exclusive frequency assignment, when there are not enough frequencies, it is possible that when a node makes its frequency decision, all physical frequencies have already been chosen by at least one node within two hops. In this case, the exclusive frequency assignment is extended by randomly choosing one of the least chosen frequencies. For convenience, we call this extension *even selection*, which makes an even allocation of available frequencies to all nodes within any two communication hops.

3.1.3 *Eavesdropping.* Even though the even selection scheme leads to even sharing of available frequencies among any two-hop neighborhood, it involves a number of two-hop broadcasts. To reduce the communication cost, we propose a lightweight eavesdropping scheme. In eavesdropping, each node takes a random backoff before it broadcasts its physical frequency decision. During the backoff period, each node records any physical frequency decision overheard. When a node's backoff timer fires, it randomly chooses one of the least chosen frequencies for data reception.

Compared with even selection, eavesdropping has less communication overhead, but it also results in more potential conflicts, because it only collects information within one hop for frequency decisions.

Algorithm 1 Frequency Number Computation

```
Input: Node $\alpha$'s ID ($ID_\alpha$), and node $\alpha$'s neighbors' IDs within two communication hops.
Output: The frequency number ($FreNum_\alpha$) node $\alpha$ gets assigned.
  $index$ = 0; $FreNum_\alpha$ = -1;
  repeat
    $Rnd_\alpha$ = Random($ID_\alpha$, $index$);
    $Found = TRUE$;
    for each node $\beta$ in $\alpha$'s two communication hops do
      $Rnd_\beta$ = Random($ID_\beta$, $index$);
      if ($Rnd_\alpha < Rnd_\beta$) or ($Rnd_\alpha == Rnd_\beta$ and $ID_\alpha < ID_\beta$) then
        $Found = FALSE$; break;
      end if
    end for
    if $Found$ then
      $FreNum_\alpha = index$;
    else
      $index$ ++;
    end if
  until $FreNum_\alpha > -1$
```

3.1.4 *Implicit-Consensus.* When physical frequencies are abundant, the communication overhead in exclusive frequency assignment can be further reduced, while all nodes within any two-hop neighborhood can still be guaranteed to get assigned different frequencies. To achieve this performance, we propose the implicit-consensus scheme, which is inspired by the pseudo random number generator algorithms proposed in the NAMA [Bao and Garcia-Luna-aceves 2001] paper. In NAMA, the pseudo random number generators are used to design distributed time scheduling in TDMA. In this paper, we extend this basic pseudo random number generator idea, proposing a distributed frequency assignment algorithm for multi-frequency MAC designs.

In implicit-consensus, nodes' IDs need to be collected within two hops, in the same way as what is done in exclusive frequency assignment. Then, each node calculates its frequency number with a local computation. In the system, all nodes share the same pseudo random number generator, which is able to generate a unique random number sequence for each specified seed, the node ID here. Algorithm 1 presents the scheme for each node to calculate its frequency number. To assist explanation, node α is taken as an example.

As algorithm 1 states, for each frequency number, each node calculates a random number (Rnd_α) for itself and a random number (Rnd_β) for each of its two-hop neighbors, with the same pseudo random number generator. A node wins the current frequency number if and only if its current random number is the highest among all current random numbers generated by all nodes within two hops. When two random numbers tie, the one with the larger node ID wins. In this way, each node explores all frequency numbers from zero to positive infinity until it finds the frequency for which it has the highest priority. By using the same pseudo random number generator, it is guaranteed that when a node decides that it wins frequency number $FreNum_i$, all nodes within two hops automatically agree with that decision and consensus is implicitly achieved, without any communication.

Here, a question may arise, since each node has a global ID. Why don't we just map nodes' IDs within two hops into a group of frequency numbers and assign those numbers to all nodes within two hops? Unfortunately, this scheme does not work, because a node's ID may get mapped to different frequency numbers in different two-hop neighborhoods. Also, it is not scalable to build a one-to-one mapping between nodes' IDs and all available frequencies, because this makes the frequency requirement depend on the network size, rather than the node density.

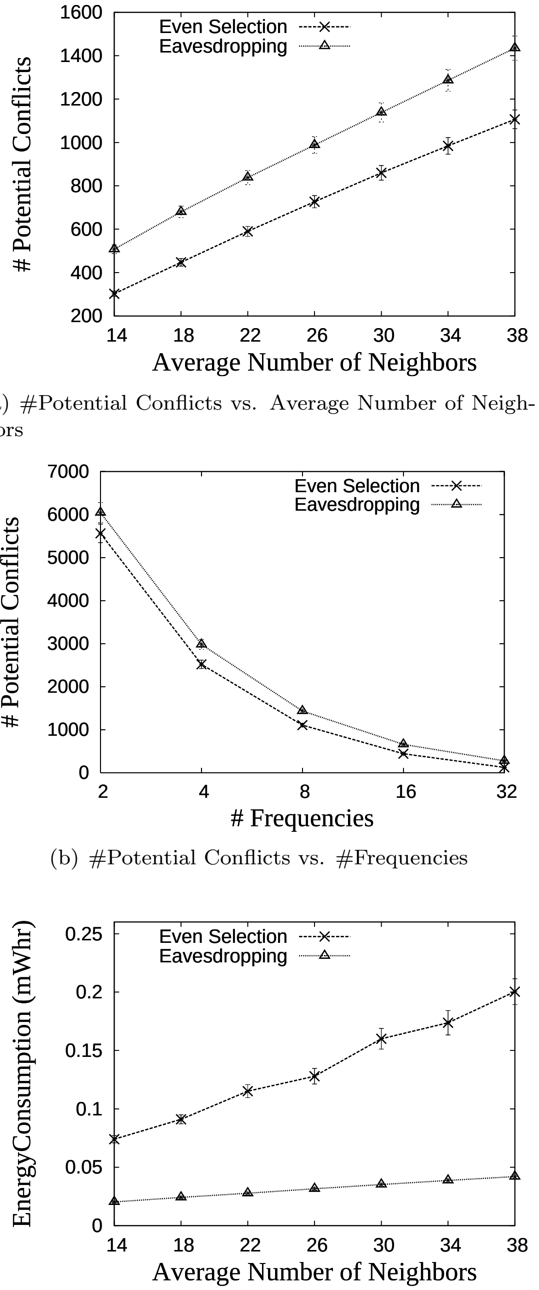
In implicit-consensus, when a node (node A) does not win the current frequency number ($FreNum_c$), because its random number is smaller than that of one of its two-hop neighbors (node B), it may happen that this neighbor (node B) has already won a previous frequency number. In this case, node B does not need the current frequency number. Node B should have already terminated its frequency number computation before it takes $FreNum_c$ into consideration, according to the repeat-until loop termination condition in algorithm 1. So, node A keeps trying larger frequency numbers until it finally finds one, while at the same time frequency number $FreNum_c$ is not chosen by any node within this two-hop neighborhood. Accordingly, the finally assigned frequency numbers among two communication hops may not be continuous. There may be holes, and some frequency numbers may not be assigned to any node, which is why the implicit-consensus scheme assumes that the available frequencies are abundant.

With the assigned frequency numbers, each node can easily calculate its physical frequency, with a local mapping. Let's put the available frequencies in a sorted list, $FreList = \{f_0, f_1, \ldots, f_N\}$, in increasing order. If the assigned frequency number is $FreNum_i$, the corresponding physical frequency is mapped to f_{FreNum_i}. After each node gets its physical frequency, it broadcasts this information to its one hop neighbors, so that each node knows what frequency to use to transmit packets to its neighbors.

3.2 Evaluation of Frequency Assignment

Among the four frequency assignment methods we proposed, exclusive frequency assignment and implicit-consensus provide the guarantee that there are no potential conflicts when there are enough frequencies available. Providing enough physical frequencies for these two methods is not a problem when the number of neighbors is not high. But when it comes to a dense wireless sensor network, there may be short of physical frequencies and two neighboring nodes may have to share the same channel. To balance data traffic among available frequencies, the even selection and eavesdropping methods are proposed. In this subsection, we evaluate the performance of even selection and eavesdropping.

In the experiments, performance is compared from three aspects. First, we com-



(a) #Potential Conflicts vs. Average Number of Neighbors

(b) #Potential Conflicts vs. #Frequencies

(c) Energy Consumption vs. Average Number of Neighbors

Fig. 2. Performance Evaluation of Frequency Assignment

pare the performance when the node density[3] increases while the number of available frequencies is fixed at 5. We use the number of potential conflicts as the performance metric, which is defined as the total number of node pairs in the system that satisfies the condition: the node pair is within two communication hops and both nodes share the same frequency. Since the two nodes are within two hops, two of their common neighbors may simultaneously transmit packets to them respectively. When they are assigned the same frequency, these two data transmissions interfere with each other, and packet loss may happen. So the number of potential conflicts measures the system's ability of fully utilizing multiple frequencies. Second, besides the average number of neighbors, we also vary the number of available frequencies, to test the performance stability of even selection and eavesdropping. Third, we measure the communication energy consumption of all nodes within the system to compare the cost each scheme pays for its performance. We also explore the cost variation when different numbers of neighbors are used.

The performance comparison is conducted in GloMoSim [Zeng et al. 1998], in which 289 (17×17) nodes are uniformly deployed in a terrain of 200m×200m square. In each simulation run, nodes have different locations. The radio type is set to RADIO-ACCNOISE [Zeng et al. 1998] and the radio bandwidth is set to 250Kbps. The performance results are illustrated in Figure 2. For each data value we present, its 90% confidence interval is given as well.

As shown in Figure 2 (a), for all the numbers of neighbors we set from 14 to 38, even selection always performs better than eavesdropping. For instance, when the average number of neighbors is 14, even selection has 302 potential conflicts, which is 40% less than the 507 potential conflicts that eavesdropping has. When the average number of neighbors increases to 38, even selection has 1106 potential conflicts, which is 23% less than the 1434 potential conflicts eavesdropping has. Even selection achieves this good performance because when a frequency decision is made, it is always the case that one of the least loaded frequencies is preferred within two hops. In this way, load is well distributed among all available frequencies within any two-hop neighborhood. However, in eavesdropping, nodes make frequency decisions based on overheard information within only one hop, which leads to a lower performance than even selection. From Figure 2 (a), it is also observed that the number of potential conflicts increases for both even selection and eavesdropping, when the average number of neighbors increases. This is because the number of frequencies is fixed at 5, so the increased number of neighbors results in the increased number of nodes that share the same frequency within two hops.

Besides the average number of neighbors, we also vary the number of available frequencies to compare the performance of even selection and eavesdropping. In Figure 2 (b), a similar phenomenon is observed: even selection performs consistently better than eavesdropping, for all the numbers of frequencies we choose from 2 to 32.

[3] In all experiments, nodes are randomly and uniformly deployed in a field, instead of using regular topologies. In addition, the topology changes at each simulation run, since different random number seeds are used in different simulation runs. The node density is defined as the average number of nodes within one communication hop, and different node densities are configured by setting different radio ranges.

For energy consumption, we measure the totaly energy consumed for transmitting and retransmitting all control and data packets. We do not include the energy consumption for idle listening and packet reception, because this heavily depends on a node's duty-cycle and node density. Also, existing duty-cycle techniques like [Polastre et al. 2004] can be incorporated into our MMSN design to save even more power, and in Appendix B we discuss how duty-cycle techniques can be integrated into MMSN. So for fair comparison and also reducing complication, we only consider transmission energy, and plot the result in Figure 2 (c). As shown in Figure 2 (c), even selection consumes more energy than eavesdropping, because even selection has two-hop neighbor discovery as well as two-hop broadcasts of frequency decisions, while eavesdropping only has one hop broadcasts.

However, this energy consumption can be amortized during data transmission, especially for comparatively static sensor networks like [Mainwaring et al. 2002] [Xu et al. 2004] [He et al. 2004] [Simon et al. 2004], where sensor devices are comparatively static and frequency assignment can be done very infrequently just for adapting to system aging and lossy links [Zhou et al. 2007] [Zhao and Govindan 2003] [Woo et al. 2003]. Accordingly, if the specific sensor network system is mostly static and the network congestion is a big issue, even selection seems to be a better choice. On the other hand, if the system topology varies a lot with time and the network is lightly loaded, eavesdropping can be used to save more energy. We will compare these two methods furthermore later in Section 4, in terms of network performance.

3.3 Media Access Design

After frequency assignment, each node gets a physical frequency for data reception. With the assigned frequencies, nodes cooperate to maximize parallel transmission within the shared space. To provide efficient broadcast support, nodes are time synchronized [Ganeriwal et al. 2007] [WernerAllen et al. 2005] [Maróti et al. 2004] [Lucarelli and Wang] [Ganeriwal et al. 2003] [Elson et al.] during media access. A time slot consists of a broadcast contention period (T_{bc}) and a transmission period (T_{tran}). During the T_{bc} period, nodes compete for the same broadcast frequency and during the T_{tran} period, nodes compete for shared unicast frequencies. The T_{tran} period also provides enough time for actually transmitting or receiving a broadcast or unicast data packet. The time slot size depends on the number of nodes that compete for the same frequency and the data packet size. The regular time slot size is 3∼5ms.

Within one time slot, a node is able to either transmit or receive one packet. Each node first checks the broadcast frequency f_0[4] for receiving or transmitting a broadcast packet. If there is no broadcast packet to transmit or receive, unicast packet transmission and reception are considered. Each node's behavior differs depending on whether it has one packet to transmit or not, as well as whether it has a unicast packet or a broadcast packet to transmit. What follows are the details.

[4] One specific physical frequency is used for broadcast during the T_{bc} period, and this frequency can be reused during the T_{tran} phase for unicast transmission. So all frequencies are fully utilized.

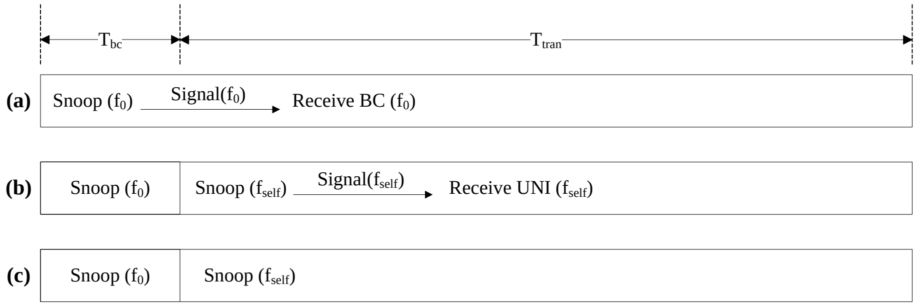


Fig. 3. When a Node Has no Packet to Transmit

3.3.1 *Has No Packet to Transmit.* If a node does not have any packet to transmit within a time slot, it behaves as Figure 3 presents. It first snoops on frequency f_0 during the time period T_{bc}. If the channel is busy, it becomes aware that another node is broadcasting a packet. So it receives the broadcast packet during the rest of the time slot, which is illustrated in case (a). On the other hand, if no signal is sensed during the time period T_{bc}, the node switches to snoop on frequency f_{self}, which is the frequency assigned to it for unicast packet reception. If a signal is sensed in frequency f_{self}, it receives the packet during the rest of the time slot, as shown in case (b). Here, we define $T_{Packet\,Transmission}$ as the time to deliver a packet after it gets the channel, which depends on the packet size and radio bandwidth. A nodes needs to keep on sensing the channel for a possibly incoming unicast packet, until the time left for the current time slot is shorter than $T_{Packet\,Transmission}$, as shown in case (c). When the time left for the current time slot is less than $T_{Packet\,Transmission}$, no neighboring nodes will send a packet to this node, so it turns off carry sensing until the start of the next time slot to save energy.

Since channel switching is needed in MMAC design, we measure the channel switching time using the widely used MicaZ [CROSSBOW 2008] sensor devices. Our experiments show that the measured channel switching time is within the range of $23.6us \sim 24.9us$, which is very small and hence implementing MMAC in a practical system with off-the-shelf sensor devices is feasible.. More detailed discussion can be found in Appendix A.

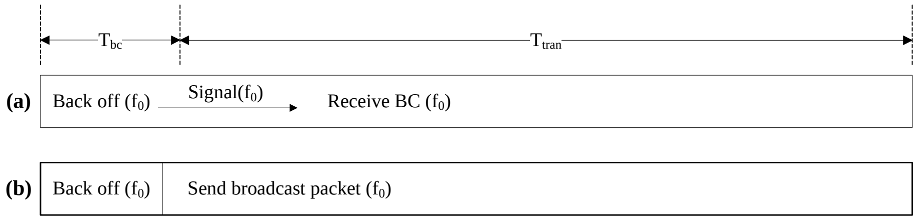


Fig. 4. When a Node Has a Broadcast Packet to Transmit

3.3.2 *Has a Broadcast Packet to Transmit.* If a node has a broadcast packet for transmission, it may have two different behaviors as illustrated in Figure 4. At the beginning of the time slot, the node uses frequency f_0, which is specified for transmitting and receiving broadcast packets. It first sets a random backoff within the time period T_{bc}. If it senses any signal during the backoff period, it becomes aware that another node is broadcasting a packet. In this case (case (a)), the node spends the rest of the time slot receiving the broadcast packet. There is another case, case (b), in which the node does not sense any signal in frequency f_0, during its backoff time period. In this scenario, a broadcast packet is sent out from this node, after the backoff timer fires.

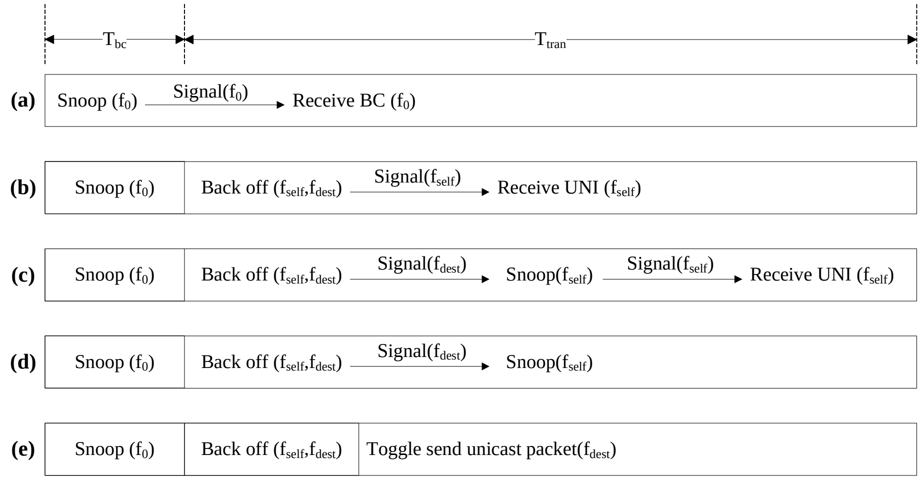


Fig. 5. When a Node Has a Unicast Packet to Transmit

3.3.3 *Has a Unicast Packet to Transmit.* Figure 5 illustrates the different behaviors a node may take, if it has a unicast packet for transmission. The node first listens to the broadcast frequency f_0 during time period T_{bc}. If it senses any signal during T_{bc}, which must be a broadcast packet, the node spends the rest of the time slot receiving the broadcast packet, as shown in case (a).

Cases (b)(c)(d)(e) illustrate the other four scenarios in which the node does not sense any broadcast signal during the time period T_{bc}. In these cases, the node takes a random backoff within the time period $T_{tran} - T_{Pakcet Transmission}$. During the backoff time period, the node snoops on two frequencies. On one hand, it snoops on frequency f_{self}, which is assigned to it for data reception, to get prepared for a possibly incoming unicast packet. On the other hand, it also snoops on frequency f_{dest}, which is assigned to the destination node of its unicast packet for data reception. If frequency f_{dest} is sensed busy, it can be aware that another node is transmitting a unicast packet to the same destination node, and it can choose not to transmit the unicast packet in the current time slot to avoid collisions. The node snoops on these two frequencies alternatingly, and we call this scheme *toggle snooping*, which is discussed in detail in subsection 3.3.4.

During toggle snooping, if the node senses any signal on frequency f_{self}, it gets to know that it itself is the destination of an incoming unicast packet. So it stops toggle snooping to receive the data packet, which is illustrated in case (b). During the toggle snooping, the node may also sense a signal on frequency f_{dest}. When frequency f_{dest} is sensed busy, the node gets to know that another node is competing for the shared frequency, by sending a unicast packet to the same destination node. In this case, the node stops toggle snooping and switches to snoop on frequency f_{self} only. It gives up transmitting a unicast packet in this time slot and prepares to receive possible data packet transmitted to it. So if any signal is sensed in frequency f_{self}, as shown in case (c), it receives the unicast packet during the rest of the time slot. Before the node senses any signal in frequency f_{self}, it keeps sensing the frequency until the time left for the current time slot is $T_{Packet\,Transmission}$, as shown in case (d). When the left time for the current time slot is shorter than $T_{Packet\,Transmission}$, it turns off carry sensing to save energy.

If the node does not sense any signal in both frequency f_{self} and f_{dest} during the backoff time period, as shown in case (e), it sends out a unicast packet with the toggle transmission technique, which is illustrated in Figure 6.

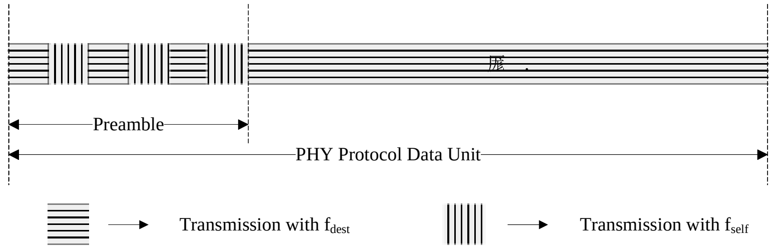


Fig. 6. Toggle Transmission

As Figure 6 illustrates, the preamble bytes of the physical layer protocol data unit (PPDU) is transmitted with two frequencies, f_{self} and f_{dest}, in an alternating way. The rest of the PPDU is transmitted to the destination node in frequency f_{dest}. The toggle transmission scheme is useful to reduce collisions. As shown in Figure 7 (a), when node B is transmitting a unicast packet to node C with the toggle transmission technique, the preamble transmitted in frequency f_{self} informs other nodes that this channel is busy, so that any node that wants to send a packet to node B can back off. On the other hand, the preamble transmitted in frequency f_{dest} informs any node that wants to send a data packet to node C to back off and avoid possible collisions. The relation of toggle transmission and toggle snooping is analyzed in the following subsection.

3.3.4 *Toggle Snooping and Toggle Transmission.* When a node has a unicast packet for transmission, toggle snooping is used during the T_{tran} period and the node snoops on two frequencies alternatingly: the frequency it uses for data reception (f_{self}), and the frequency the destination node of its unicast packet uses for data reception (f_{dest}). The time a node takes to snoop on both of the two frequencies for one round is called the toggle snooping period, represented by parameter T_{TS}. In toggle transmission, a node transmits the preamble bytes of the PPDU with two frequencies, the frequency the node itself uses for data reception (f_{self}) and the frequency the destination node of the unicast packet uses for data reception

(f_{dest}). The transmitter switches between these two frequencies alternatingly and the time the node sweeps the two frequencies for one round is called the toggle transmission period, represented by parameter T_{TT}.

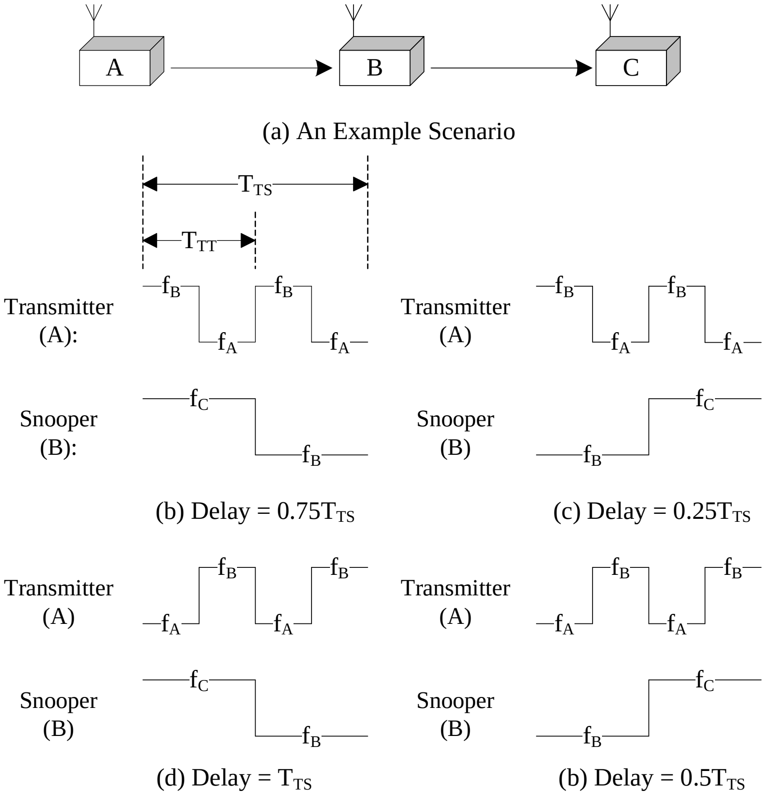


Fig. 7. Toggle Snooping

In the MMSN protocol, we let $T_{TS} = 2 \times T_{TT}$, so that when one node sends out a packet using the toggle transmission scheme, any other node that is snooping using the toggle snooping scheme is able to detect this transmission within a maximal delay of T_{TS}, if toggle transmission and toggle snooping have any shared frequency. With the help of Figure 7, we make this point more clear. In Figure 7 (a), node A uses frequency f_A for packet reception and it has a unicast packet to send to node B. Node B uses frequency f_B for packet reception and it has a unicast packet to send to node C, which uses frequency f_C for packet reception. During the T_{tran} time period, both A and B set up backoff timers and snoop on two frequencies. Node A snoops on frequency f_A and f_B and node B snoops on frequency f_B and f_C. Let's suppose that node A's timer fires first. So node A switches from toggle snooping to toggle transmission, while node B is still in the toggle snooping state. In different application scenarios (not only limited to the case in Figure 7 (a)), node B may take different time delays to become aware that node A is transmitting, as shown in Figure 7 (b)(c)(d)(e). In the scenario presented in case (b), node B is able to detect node A's transmission in frequency f_B after the time delay of $0.75\,T_{TS}$. In case (c), the delay to detect node A's transmission is $0.25\,T_{TS}$. In case (d) and (e), the delays are T_{TS} and $0.5\,T_{TS}$, respectively.

According to the above analysis, it is guaranteed that when one node transmits a packet using the toggle transmission scheme, the maximum time delay for another node, which uses the toggle snooping scheme, to detect the transmission is T_{TS}. Accordingly, if the backoff timer used in the slotted time period in Figure 5 is only allowed to fire at the end of a toggle snooping period, a node whose backoff timer fires after the previous one can have enough time to detect the previous node's transmission, and hence abandon its transmission in the current time slot to reduce congestion.

3.4 IMPLICATION OF BACKOFF ALGORITHMS

In media access, neighboring nodes may compete for the same physical frequency in both the broadcast contention period (T_{bc}) and the transmission period (T_{tran}), as explained in Section 3.3. To reduce congestion, random backoff is needed for both broadcast and unicast transmissions. Taking unicast backoff as an example, we give theoretical analysis to prove that a uniform backoff algorithm is not a good choice for the time synchronized media access in MMSN, and a non-uniform backoff algorithm achieves better performance. We derive an optimal non-uniform backoff algorithm, and choose its lightweight approximation for implementation in MMSN. All results derived here also apply for the broadcast transmission in MMSN.

During the backoff in the T_{tran} period in Figure 5, the time slot is further divided into small time slices. As explained in the previous section, each time slice has the length of T_{TS} and each backoff timer is only allowed to fire at the end of a time slice. If any two nodes choose the same backoff time slice, there is a collision. In order to minimize the probability of collision, we derive an optimal bound and a simple suboptimal distribution of the backoff time slices.

First we derive the optimal probability distribution $P(t)$ of backoff time slice t to minimize the probability of collisions when two nodes attempt to grab the same time slice after backoff. $P(t)$, $t = 0, 1, ..., T$, denotes the probability that a node attempts to grab time slice t and T is the maximum backoff time slice. Obviously $0 \leq P(t) \leq 1$ and $\sum_{t=0}^{T} P(t) = 1$. We assume that each node independently selects the backoff time slice conforming to the same distribution.

According to the analysis in Section 3.3, in a time slot, the node that selects the earliest backoff time slice gets the physical channel, and all nodes whose backoff timers fire later should abandon their transmission. Hence, a node successfully grabs time slice t if all other nodes attempt to grab time slices after t. If at least two nodes in the same neighborhood attempt to grab the same earliest time slice, there is a collision. We need to find the probability distribution $P(t)$ to maximize P_{nc}, the probability that there is only one node that grabs the earliest time slice, to avoid collisions as much as possible. Assuming the earliest time slice is i, $0 \leq i \leq T-1$, and there are N nodes in the neighborhood, the probability that one and only one node attempts to grab this time slice and all other nodes attempt to grab later time slices is $N \cdot P(i) \cdot S_{i+1}^{N-1}$, where $S_{i+1} = \sum_{t=i+1}^{T} P(t)$. Considering all possible earliest time slices, we have

$$P_{nc} = \sum_{i=0}^{T-1} N \cdot P(i) \cdot S_{i+1}^{N-1}.$$

Now we apply a recursive approach to decide the optimal probability distribution $P(t)$. First we assume that the values for $P(t)$, $t = 0, ..., T-2$, are already known. From the constraints that the sum of all $P(t)$ values is 1, $S_{T-1} = P(T-1) + P(T)$ is also known. The question is how to divide S_{T-1} between $P(T-1)$ and $P(T)$ to maximize P_{nc}. This division only affects the term $N \cdot P(T-1) \cdot P(T)^{N-1}$ in the calculation of P_{nc}. The other terms are not affected by the way S_{T-1} is divided. For simplicity we denote $P(T)$ as a and $P(T-1)$ as b. The first order condition for maximization is

$$\frac{d}{da}(Nba^{N-1}) = N(N-1)S_{T-1}a^{N-2} - N^2a^{N-1} = 0,$$

and we have

$$S_T = P(T) = a = k_T S_{T-1},$$

where $k_T = \frac{N-1}{N}$.

We omit the validation of the second order condition for brevity, but for $N \geq 2$, the above equation does give a maximized result

$$N \cdot P(T-1) \cdot P(T)^{N-1} = k_T^{N-1} S_{T-1}^N.$$

Then we consider the division of probability S_{T-2} between $P(T-2)$ and S_{T-1}, assuming that the values of $P(t)$, $t = 0, ..., T-3$ are known. For simplicity we denote $P(T-2)$ as c. The terms affected by this division are only

$$NcS_{T-1}^{N-1} + k_T^{N-1} S_{T-1}^N.$$

The first order condition is

$$N(N-1)S_{T-2}S_{T-1}^{N-2} + (k_T^{N-1} - N)NS_{T-1}^{N-1} = 0,$$

and therefore

$$S_{T-1} = k_{T-1}S_{T-2},$$

where $k_{T-1} = \frac{N-1}{N-k_T^{N-1}}$. Then we obtain the optimal value of the sum of the two terms:

$$NcS_{T-1}^{N-1} + k_T^{N-1} S_{T-1}^N = k_{T-1}^{N-1} s_{T-2}^N.$$

With a similar approach we get the recursive formulas for S_t and k_t as follows.

$$S_{t+1} = k_{t+1}S_t,$$

where $t = 0, ..., T-1$, and $S_0 = 1$.

$$k_{T-t-1} = \frac{N-1}{N-k_{T-t}^{N-1}},$$

where $t = 0, ..., T-2$, and $k_T = \frac{N-1}{N}$.

Therefore, the optimal distribution $P(t)$ is

$$P(t) = S_t - S_{t+1},$$

where $t = 0, ..., T-1$, and $P(T) = S_T$.

The optimal distribution gives an optimal bound of the non-collision probability. However, the distribution depends on the number of competing nodes, which may vary from neighborhood to neighborhood in deployed systems. Also, the process of computing the distribution is complicated and hence too costly for power-limited sensor devices. Accordingly, if a simple solution can provide a non-collision probability close to the optimal bound, it is more favorable. We propose a suboptimal distribution to be used by each node, which is easy to compute and does not depend on the number of competing nodes. A natural candidate is an increasing geometric sequence, in which

$$P(t) = \frac{b^{\frac{t+1}{T+1}} - b^{\frac{t}{T+1}}}{b-1}, \tag{1}$$

where $t = 0, ..., T$, and b is a number greater than 1.

The problem is which value of b should be chosen. We choose various b values to calculate the corresponding non-collision probabilities and compare them with that from the optimal $P(t)$. To be consistent with the evaluation section, we choose the same number of time slices and node densities. The results are shown in Figure 8. From the figure we can see that if we choose $b = 1000$, the difference between the simple solution's non-collision probability and that of the optimal $P(t)$ is smaller than 6% for the average number of nodes we choose and $T = 33$, which is the number we use in the simulation.

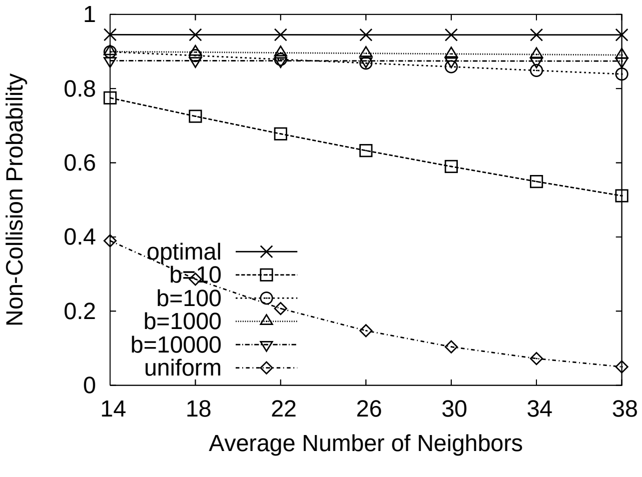


Fig. 8. Non-Collision Probability with Various b Values

A similar approach is also proposed in [Tay et al. 2004], which minimizes collisions in slotted CSMA. We deem that the optimal solution is more relevant to our MAC than the slotted CSMA. The slice time for slotted CSMA can be chosen as small as the sum of propagation delay, detection time and other processing delays, which are in microseconds typically. Compared with the maximum backoff time, the slice time is orders of magnitudes smaller, so the number of slices is large. When the slice number approaches infinity, the non-collision probability for a uniform distribution is

$$\lim_{T\to\infty} P_{nc} = \lim_{T\to\infty}(N\sum_{i=0}^{T-1}\frac{1}{T+1}(\frac{T-i}{T+1})^{N-1}).$$

Let $a = \frac{T-i}{T+1}$, from the definition of the Riemann integral, we have

$$\lim_{T\to\infty} P_{nc} = N\int_0^1 a^{N-1}da = N\cdot\frac{1}{N} = 1.$$

Therefore, when the slice number $T+1$ approaches positive infinity, the non-collision probability approaches 1, which means even the uniform distribution gives a very small chance of collision. Calculation shows that if we have 1000 time slices, even when 200 nodes compete, the non-collision probability for a uniform distribution is still above 90%. Since the slice number we use in MMSN is much smaller ($T+1 = 34$ in our simulation), to reduce protocol overhead, the suboptimal approach shows a significant performance improvement over the uniform distribution, as shown in Figure 8.

In our algorithm, we use the suboptimal approach for simplicity and generality. We need to make the distribution of the selected backoff time slice at each node conform to what is shown in Equation (1). It is implemented as follows: first a random variable α with a uniform distribution within the interval $[0,1)$ is generated on each node; then time slice i is selected according to the following equation:

$$i = \lfloor (T+1)\log_b[\alpha(b-1)+1]\rfloor.$$

It can be easily proven that the distribution of i conforms to Equation (1).

4. PERFORMANCE EVALUATION

We implement MMSN in GloMoSim [Zeng et al. 1998] and conduct extensive experiments to evaluate its performance and compare it with CSMA as well. In this evaluation, MMSN uses even selection for frequency assignment, since it results in fewer potential conflicts. For this performance evaluation, six groups of experiments are designed. In the first group, different traffic patterns are used. In the second group, different system loads are considered, and in the third group of experiments, the average number of neighbors is varied. In the fourth group of experiments, we increase the radio interference range and study the impact of different radio interference levels on the system performance. In the fifth group of experiments, we study the performance when different frequency assignment methods are use. Finally, in the sixth group of experiences, we review the state-of-the-art time synchronization techniques in wireless sensor network research, and evaluate the impact of time synchronization errors on the system performance.

In the experiments, we use four performance metrics: aggregate MAC throughput, packet delivery ratio, channel access delay, and energy consumption. The

Table I. Simulation Configuration

TERRAIN	(200m×200m) Square
Node Number	289
Node Placement	Uniform
Application	Many-to-Many/Gossip CBR Streams
Payload Size	32 Bytes
Routing Layer	GF
MAC Layer	CSMA/MMSN
Radio Layer	RADIO-ACCNOISE
Radio Bandwidth	250 Kbps
Radio Range	20m∼45m

aggregate MAC throughput measures the performance gain and is calculated as the total amount of useful data successfully delivered through the MAC layer in the system per unit time. The packet delivery ratio is calculated as the ratio of the total number of data packets successfully delivered by the MAC layer over the total number of data packets the network layer requests the MAC to transmit. The channel access delay measures the time delay a data packet from the network layer waits for the channel before it is sent out. The energy consumption reflects the cost each protocol pays to achieve its performance, which is calculated as the energy consumed to successfully deliver a useful data byte. Since we have measured the cost for each frequency assignment scheme in Section 3.2 and this energy consumption is amortized during data transmission, it is no longer counted here.

During all the experiments, the Geographic Forwarding (GF) [Karp 2000] routing protocol is used. GF exploits geographic information of nodes and conducts local data forwarding to achieve end-to-end routing. It is more flexible and less costly. GF is also used as a baseline routing protocol in a real sensor network system [Li et al. 2003] as well as in simulation study [He et al. 2003],[Cao and Abdelzaher 2006] and [Lu et al. 2002]. Our simulation is configured according to the settings in Table I. Each run lasts for two minutes and repeated 100 times. For each data value we present in the results, we also give its 90% confidence interval.

4.1 Performance Evaluation with Different Traffic Patterns

In the first group of experiments, two different traffic patterns are used, many-to-many and gossip traffic patterns. The many-to-many traffic pattern is used to simulate the typical sensor network application scenario: multiple sensor nodes report their readings to multiple base stations over multiple hops. Since the routing design affects the contention level at the MAC layer (e.g., hot spots), the MAC performance is more statistically valid when a simulation can isolate the effect from the routing layer. Therefore, we also evaluate the MAC performance with the gossip traffic pattern, in which each node only communicates with its neighbors. For both of these two traffic patterns, we increase the number of available frequencies to observe the performance variation. In this group of experiments, 50 CBR streams are used and the average number of neighbors is set to 38, by configuring the radio range to 40m. To achieve meaningful results, we evaluate the performance when the packet delivery ratio in the MAC layer is reasonably high, higher than 93%. The small amount of packet loss is due to hidden terminal problems [IEEE 802.11 1999].

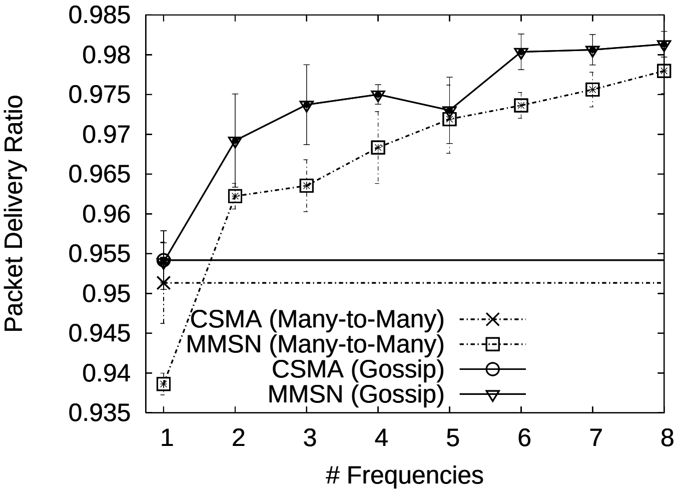


Fig. 9. Packet Delivery Ratio with Different #Physical Frequencies

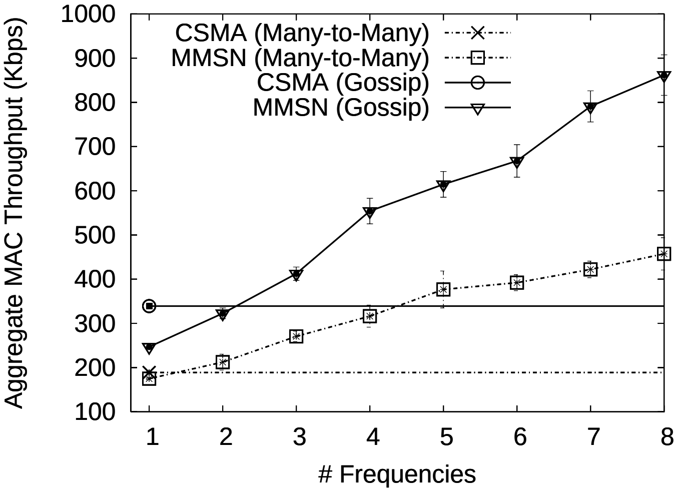


Fig. 10. Aggregate Throughput with Different #Physical Frequencies

The performance results illustrated in Figure 9 $\sim$ 12 confirm MMSN's scalability. When the number of frequencies increases from 1 to 8 and the gossip traffic is used, Figure 9 illustrates that the packet delivery ratio increases from 95.4% to 98.1%, Figure 10 shows that the aggregate MAC throughput increases from 246.9 Kbps to 861.8 Kbps, Figure 11 informs that the average channel access delay decreases from 0.069s to 0.016s, and Figure 12 states that MMSN becomes more energy efficient: the energy consumption per byte of successfully delivered data decreases from 2.47×10^{-7} mWhr to 2.40×10^{-7} mWhr. Similar performance increase is also exhibited when many-to-many traffic pattern is used. MMSN's performance increases, because available physical frequencies are evenly shared within two-hop neighboring

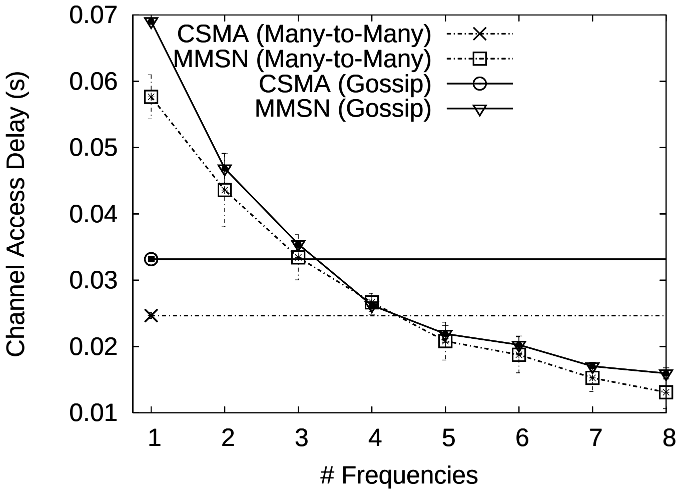


Fig. 11. Average Channel Access Delay with Different #Physical Frequencies

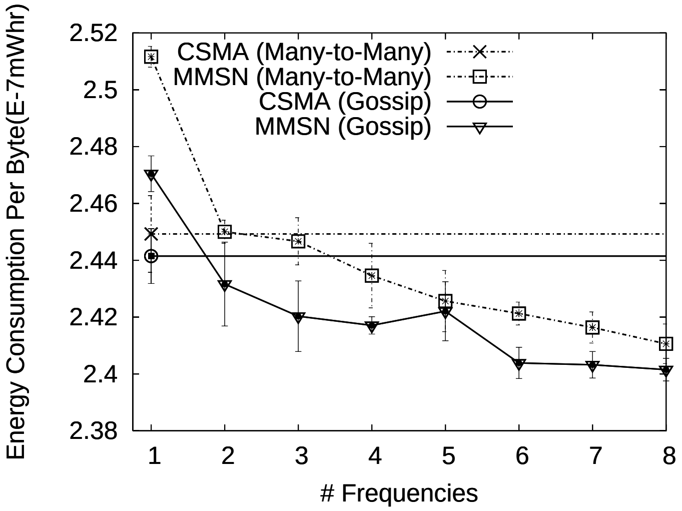


Fig. 12. Energy Consumption with Different #Physical Frequencies

nodes, and the increase of available frequencies leads to a higher degree of parallel data transmission within each neighborhood. When more physical frequencies are used, more nodes are able to conduct simultaneous transmissions in the deployed system without collisions, so the aggregate MAC throughput increases. Plus, fewer nodes are assigned to use the same frequency within two hops. So communication interference decreases, which leads to less backoff and decreased channel access delay. Also, the decreased communication interference leads to less packet loss, and more useful data bytes are successfully delivered with the same amount of energy. On the other hand, MMSN does not achieve 8 times performance improvement when 8 frequencies are used compare to the case when one frequency is used. This is due to the fundamental hardware limitation of using a single transceiver in each sensor device.

Compared with CSMA, MMSN has similar or a little lower performance when the number of frequencies is small. This is because MMSN has a fixed backoff time period allocated within each time slot, while CSMA can fire the backoff timers at any time within the backoff window. However, when the number of frequencies increases, more parallel transmissions within each neighborhood occur,which results in more gains than the cost paid due to the fixed backoff period, and MMSN outperforms CSMA.

We are also aware that MMSN has constantly increasing aggregate MAC throughput when the gossip traffic pattern is used, while the speed of throughput increase slows down when the many-to-many traffic pattern is used. This is because the many-to-many traffic consists of a number of many-to-one traffic, in which multiple nodes transmit data packets to the same destination node. In this case, all these transmitters use the same physical frequency that the destination node gets assigned, and hence there is no potential parallel transmission that can be utilized. This is also one major difference between the single-transceiver[5] multi-frequency MMSN protocol and the multi-transceiver multi-frequency protocols proposed in [Nasipuri et al. 1999] [Wu et al. 2000] [Nasipuri and Das 2000].

4.2 Performance Evaluation with Different System Loads

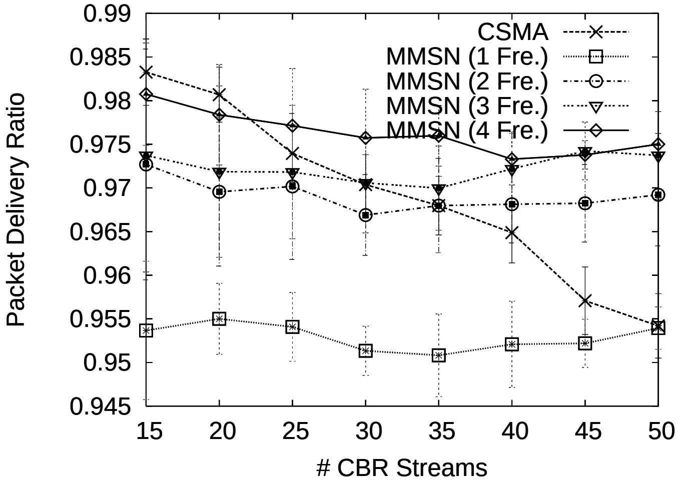


Fig. 13. Packet Delivery Ratio with Different System Loads

In the second group of experiments, we explore MMSN's performance when different system loads are used, which are generated by different numbers of CBR

[5] One solution is for each base station to have multiple transceivers. The multiple transceivers snoop on different frequencies, so that the base station can receive simultaneous data packets from multiple nodes.

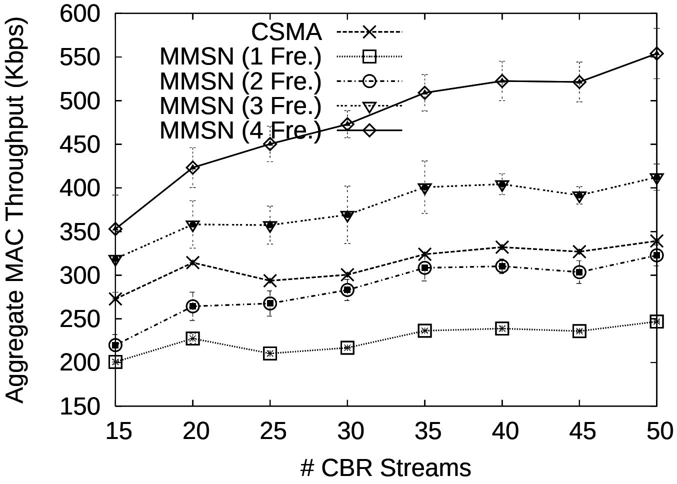


Fig. 14. Aggregate Throughput with Different System Loads

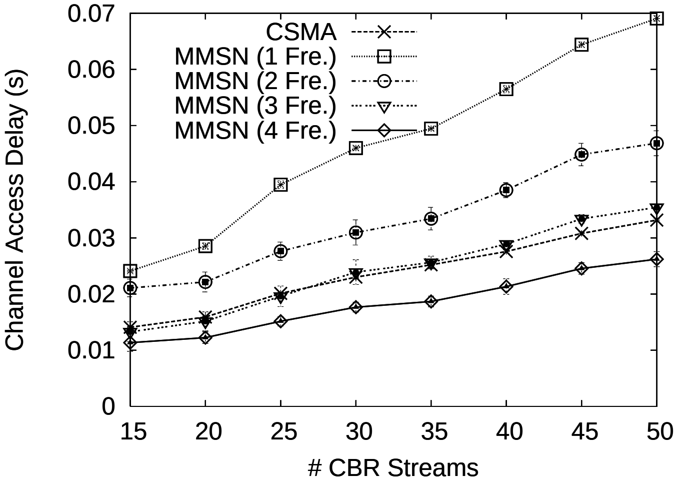


Fig. 15. Average Channel Access Delay with Different System Loads

streams. To analyze performance scalability, we conduct all experiments with different numbers of frequencies as well. In the experiments, the average number of neighbors is set to 38, and the gossip traffic pattern is used.

As Figure 13 ∼ 16 shows, for all the system loads we configure from 15 CBR streams to 50 CBR streams, it is observed that MMSN always exhibits better performance when more frequencies are used, which is consistent with the result presented in the previous group of experiments. For example, when the number of frequencies increases from 1 to 4 and 40 CBR streams are used, MMSN's packet delivery ratio increases from 95.2% to 97.3% in Figure 13. At the same time, MMSN's aggregate MAC throughput increases by 119% from 239Kbps to 523Kbps as shown in Figure 14, and the channel access delay decreases to 0.021s, which is 37.5%

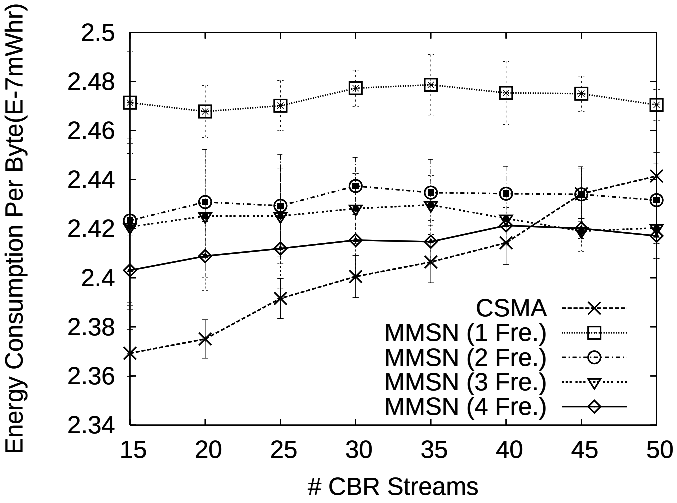


Fig. 16. Energy Consumption with Different System Loads

of the delay when only 1 frequency is available, as shown in Figure 15. In such a case, Figure 16 also informs that MMSN's energy consumption for each successfully delivered data byte decreases from 2.48×10^{-7}mWhr to 2.42×10^{-7}mWhr. MMSN achieves improved performance when the number of frequencies increases, because the increased frequencies lead to increased parallel transmission within the same neighboring space and to decreased congestion for the same physical frequency.

Figure 13 also shows that CSMA has a decreased packet delivery ratio from 98.3% to 95.4%, while MMSN does not have such an obvious packet loss. This is because the non-uniform backoff algorithm design is more tolerant to the system load variation than the uniform backoff algorithm. The sharply increased system load, from 15 CBR streams to 50 CBR streams, leads to more congestion and more packet loss in CSMA while the slotted backoff is not impacted as much. In Figure 14, the aggregate MAC throughput increases with the increase of system load, because more nodes get involved in communication and more parallel data transmissions occur. In addition, the increased nodes becoming involved in communication result in increased congestion and hence channel access delay increases in Figure 15. Since CSMA is more sensitive to system load and has lower packet delivery ratio, it is less energy efficient when the system load increases, while MMSN' packet delivery ratio is more tolerant to system load increase and hence does not exhibit apparent decrease of energy efficiency.

For similar reasons as explained in the previous experiments, MMSN is observed to have a lower performance than CSMA when there is only one, or two in some cases, physical frequencies available as shown in Figure 13 ∼ Figure 16. However, MMSN outperforms CSMA when three or more frequencies are used, which is also exhibited in Figure 13 ∼ 16.

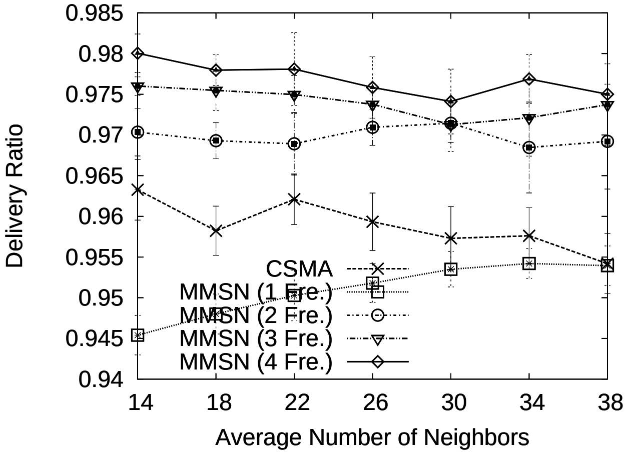


Fig. 17. Packet Delivery Ratio with Different Number of Neighbors

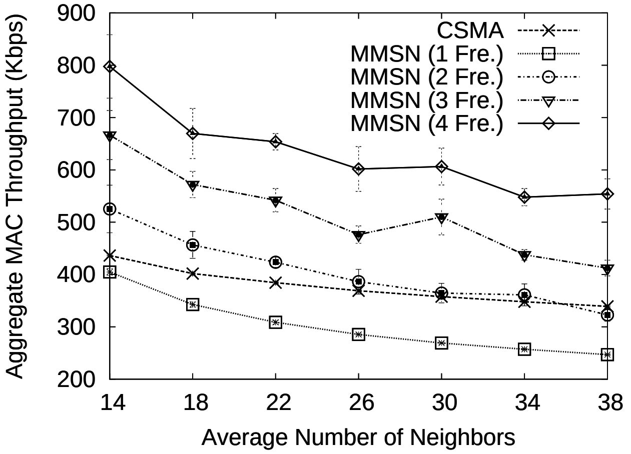


Fig. 18. Aggregate Throughput with Different Number of Neighbors

4.3 Performance Evaluation with Different Number of Neighbors

In many deployed sensor network systems [Mainwaring et al. 2002] [Xu et al. 2004] [He et al. 2004] [Simon et al. 2004], providing node redundancy is an efficient and effective method to increase the system lifetime. So, in the third group of experiments, we evaluate MMSN's performance when different numbers of neighbors are utilized. The number of neighbors is increased from 14 to 38, by configuring different radio ranges, and a gossip traffic pattern is used that consists of 50 CBR streams. We also measure the performance difference when different numbers of frequencies are used.

Once again, the experimental results confirm that MMSN always achieves a higher performance when more frequencies are available, which can be observed

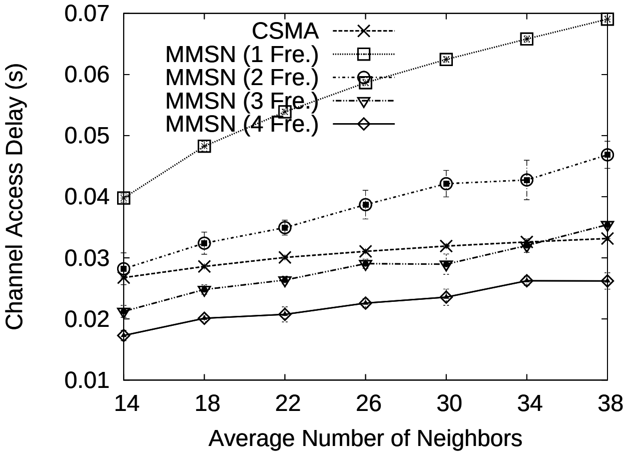


Fig. 19. Average Channel Access Delay with Different Number of Neighbors

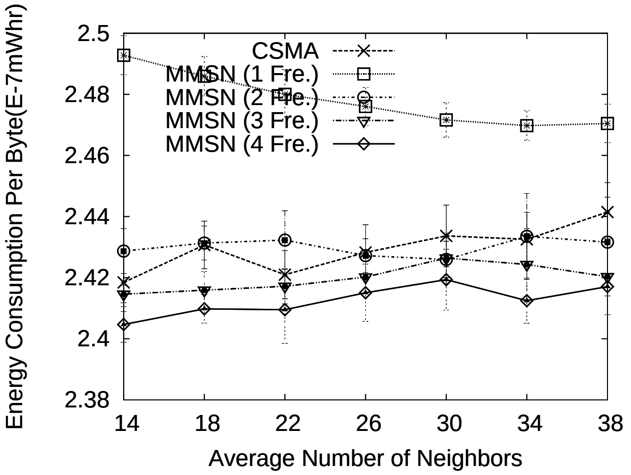


Fig. 20. Energy Consumption with Different Number of Neighbors

in Figure 17 $\sim$ 20. The corresponding reasons can be found in the first two groups of experiments and are not repeated here.

From Figure 18, it is observed that the aggregate MAC throughputs in both CSMA and MMSN decrease when the the average number of neighbors increases. This is because when the number of neighbors increases, the same number of frequencies are shared by more nodes within two hops. When the same percentage of nodes participate in communication, congestion is increased and hence backoff and channel access delay are increased, as shown in Figure 19. We do not observe consistent trends for packet delivery ratio variation in Figure 17 and energy consumption variation in Figure 20, when the number of frequencies is greater than one and the number of neighbors is increased from 14 to 38. But we do notice that when there is only one frequency, the packet delivery ratio of MMSN increases in Figure 17, with the increase of neighbors. We think this is because of decreased hidden terminal problems, when the radio range gets increased to increase the average number of neighbors, while at the same time the system topology is fixed to be 200m×200m. When the number of frequencies increases, this effect becomes very small and no similar trend is observed. Also, because of the increased packet delivery ratio, the energy consumption becomes more efficient as shown in Figure 20, when MMSN uses one frequency and the average number of neighbors increases from 14 to 38.

4.4 Performance Evaluation with Different ICR Values

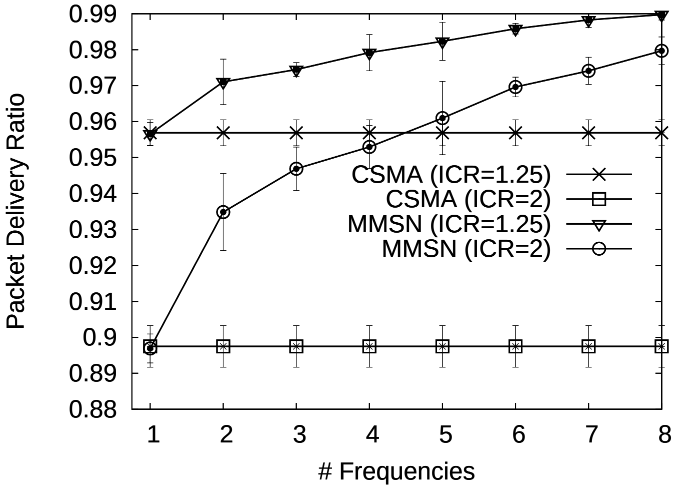


Fig. 21. Packet Delivery Ratio with Different ICR

This experiment is designed to explore the performance sensitivity to different ICR ratios, which is defined as ICR = RI/RC. Here RI is the radio interference range and RC is the radio communication range. This experiment is important because different hardware have different communication abilities, and ICR values may be different from device to device. While a complete study of the communication/interference relations can be found in [Zhou et al. 2005], here we give a brief definition of the radio interference range RI: RI is the maximum distance between a noise node and the receiving node, when the noise node's signal can still interrupt the receiving node's data reception from a transmitting node that is RC away.

In previous performance evaluations, the ICR value is 1.25, and in this evaluation we compare the performance when both 1.25 and 2 are used. We evaluate the packet delivery ratio and plot the data in Figure 21. Several interesting observations can be made in Figure 21.

First, we observe that for both ICR=1.25 and ICR=2 cases, MMSN always achieves a higher packet delivery ratio than CSMA, when more than 1 physical frequencies are used. This is because MMSN balances traffic load among multiple physical frequencies and hence reduces the congestion within each frequency.

Second, MMSN's packet delivery ratio increases when the number of available frequencies increases, no matter ICR is 1.25 or 2. This demonstrates that the frequency assignment and media access control in MMSN help achieve scalable performance.

Third, when ICR is 2, MMSN's packet delivery ratio increases more quickly than the case when ICR is 1.25. For instance, when ICR is 1.25, MMSN's packet delivery ratio increases from 95.66% (1 frequency) to 98.97% (8 frequencies), which is 3.5% performance increase. However, when ICR is 2, MMSN's packet delivery ratio increases from 89.69% (1 frequency) to 97.97% (8 frequencies), which is 9.2% performance increase and almost 3 times better than the 3.5% performance increase. The reason for the significantly greater performance increase is that when ICR increases, the congestion is severer and hence the traffic is comparatively heavier, which brings in more space for potential performance improvement.

Fourth, we observe that the two curves that represent MMNS's packet delivery ratios tend to converge when more physical frequencies are available. For example, the difference of MMSN's packet delivery ratios is 95.66%-89.69%=5.97% when only 1 frequency is available. The 5.97% performance difference is exclusively due to the increase of the ICR value. However, the performance difference decreases to 2.63% when 4 physical frequencies are available, and further decreases to 1% when 8 physical frequencies are available. This is because the frequency assignment in MMSN leads to perfect traffic balance, which reduces congestion and functions against the increased radio interference.

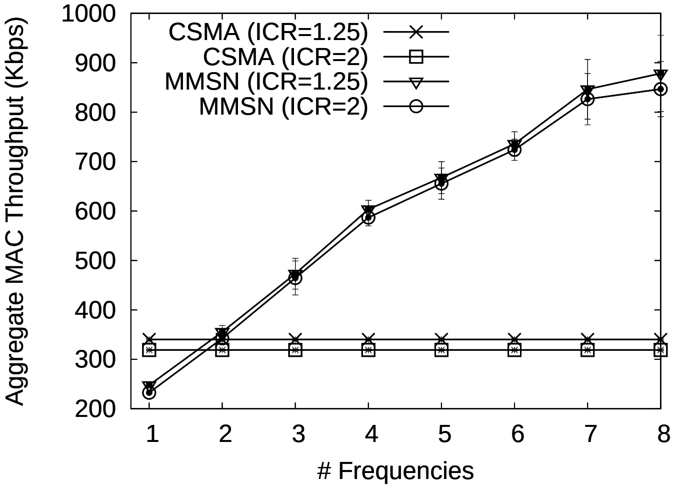


Fig. 22. Aggregate Throughput with Different ICR

In this experiment, we also evaluate the aggregate MAC throughput and plot the results in Figure 22. As presented in Figure 22, no matter the radio interference range is 1.25 or 2 times of the radio communication range, MMSN achieves a consistently higher aggregate MAC throughput than CSMA, when 2 or more physical channels are available. This demonstrates that current sensor network systems can greatly benefit from the MMSN protocol, because the currently widely used sensor devices like MicaZ [CROSSBOW 2008] and Telos [Polastre et al. 2005] have 16 physical channels. Due to the load balance among available frequencies, MMSN also achieves scalable performance when more physical frequencies are available, which is also consistent with what we observed in previous experiments.

From Figure 22, we also observe that both MMSN and CSMA achieve slightly lower aggregate MAC throughput when the interference range increases from 1.25 to 2 times of the communication range. For example, when 4 physical channels are available and the ICR is 1.25, MMSN's aggregate throughput is 603.4 Kbps; but when ICR increases to 2, MMSN's aggregate throughput decreases to 586.7Kbps. For CSMA, the aggregate throughput is 340.2Kbps when ICR is 1.25, and it decreases to 318.8Kbps when ICR increases to 2. For both CSMA and MMSN, the slight aggregate MAC throughput decrease is due to the increased congestion that comes from the increased radio interference range.

4.5 Performance Evaluation with Different Frequency Assignment Methods

In Section 3.1, we propose two frequency assignment methods, eavesdropping and even selection, to balance potential traffic among available physical frequencies. In eavesdropping, each node first overhears what frequencies its neighbors have chosen, and then randomly picks one of the least chosen frequencies. In even selection, available physical frequencies are guaranteed to be most evenly assigned to all nodes within any two communication hops. We have compared the performance of even selection and eavesdropping, in terms of the number of potential conflicts, that is, the number of node pairs that use the same physical frequencie within two hops. The performance we demonstrate in Section 3.1 illustrates that even selection performs better than eavesdropping.

In this subsection, we compare even selection and eavesdropping, with more detailed metrics, including the aggregate MAC throughput, time delay, and packet delivery ratio. This evaluation is important, since whether two simultaneous transmissions result in packet loss is not only determined by whether the two transmitters are within two communication hops but also the received data signal strength, the background noise strength, the interference signal strength, and the receiver's SNR threshold. All these details are simulated in the GloMoSim[Zeng et al. 1998] simulator that we use here to evaluate the network performance. The simulation configuration is the same as that in the previous experiment, and the simulation results are illustrated in Figure 23, Figure 24, and Figure 25.

As shown in Figure 23, even selection achieves a higher aggregate MAC throughput than eavesdropping, for a wide range of node densities. This is because even selection guarantees that in any two-hop region, the number of node pairs that use the same physical frequencie is minimized, while in eavesdropping each node only locally minimizes the potential conflict within one-hop.

From Figure 24 and 25, we observe that even selection achieves a slightly lower packet delivery ratio but a much smaller time delay than eavesdropping. As shown in Figure 24, even selection's packet delivery ratio is only 0.63% higher than that of eavesdropping, but in Figure 25, even selection's time delay is 14.28% higher than that of eavesdropping. This also explains why even selection achieves a higher aggregate MAC throughput than eavesdropping. The reason is that even selec-

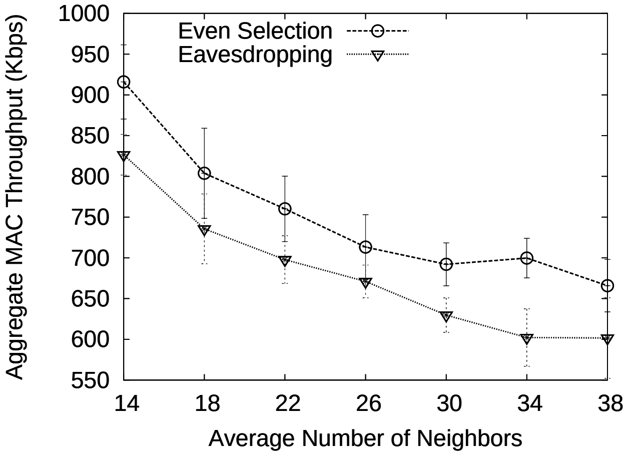


Fig. 23. Aggregate Throughput with Different Frequency Assignment Methods

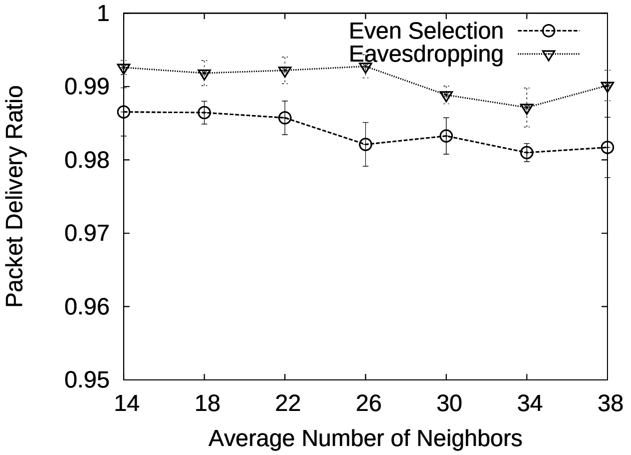


Fig. 24. Packet Delivery Ratio with Different Frequency Assignment Methods

tion successfully delivers more data bytes to sinks within a unit time period than eavesdropping.

Even selection achieves a slightly lower packet delivery ratio than eavesdropping, because even selection minimizes simultaneous transmissions within two hops, while simultaneous transmissions within two hops may or may not actually result in packet loss. Compared with two-hop simultaneous transmissions, one-hop simultaneous transmissions are more likely to cause packet loss, and this is the minimization goal of eavesdropping. However, when it comes to time delay, a node may goes through a similar back off time period, no matter it overhears a one-hop or two-hop neighbor's transmission. This is because a node backs off when it overhears a weak or median signal but only receives a packet when it hears a much stronger

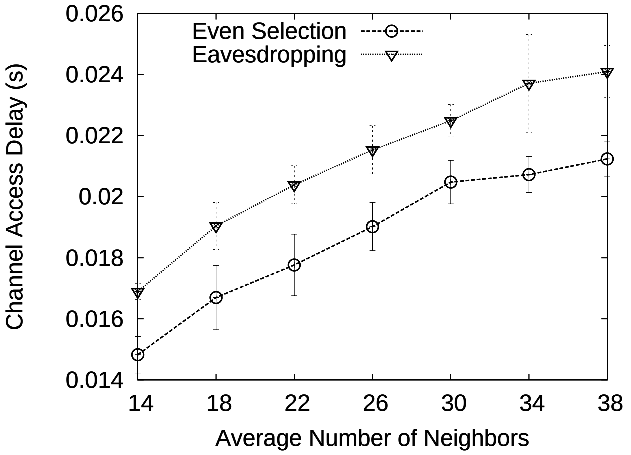


Fig. 25. Time Delay with Different Frequency Assignment Methods

signal. Since even selection minimizes simultaneous transmissions within two hops while even selection does not, even selection achieves a much lower time delay than eavesdropping.

4.6 Time Synchronization and its Impact

As we have discussed in Section 3.3, it is important to have a globally synchronized time clock for achieving the best MMSN performance. During the past a few years, a number of efficient and effective time synchronization protocols, [Ganeriwal et al. 2007] [WernerAllen et al. 2005] [Maróti et al. 2004] [Lucarelli and Wang] [Ganeriwal et al. 2003] [Elson et al.], have been developed to provide such a globally synchronized time clock for networked sensor devices. Among them, FTSP [Maróti et al. 2004], TPSN [Ganeriwal et al. 2003] and RBS [Elson et al.] can achieve within a few microseconds synchronization accuracy over a minute period, and RATS [Ganeriwal et al. 2007] can achieve a few hundred microseconds synchronization accuracy over half an hour period.

In this experiment, we use both FTSP and RATS to study the impact of time synchronization errors on the system performance. We choose FTSP because it is one of most widely used time synchronization solutions in deployed running systems like VigilNet [He et al. 2004]. FTSP achieves 1 us time synchronization error in a 60 nodes Mica2 testbed [Maróti et al. 2004], when there is no node failure. Even when there are frequent node failures, the observed maximum time synchronization error for FTSP in the 60 nodes testbed is no greater than 67 us. To achieve this performance, each node in FTSP only pays 2 beacon messages within every minute. In our GlomoSim [Zeng et al. 1998] evaluation, we configure the maximum time error as 5 us to simulate the case when FTSP is used and there is no node failure. We also configure the maximum time error as 67 us to simulate the case when FTSP is used but frequent node failures happen.

Besides FTSP, we also choose RATS for our performance study, because RATS greatly reduces the time synchronization overhead while at the same time maintains a less accurate but also reasonably good global time. In [Ganeriwal et al. 2007], RATS is reported to maintain a 225 *u*s error bound, with faulty ratio of 1.8%, in an indoor Mica2 testbed, by paying 1 control packet for each node within every 29.8 minutes. When it comes to an outdoor Mica2 testbed, RATS achieves a 225 *u*s error bound, with faulty ratio of 2.1%, by paying 1 control packet for each node within every 25.6 minutes. In our GlomoSim [Zeng et al. 1998] evaluation, we configure the maximum time synchronization error to be 225 *u*s to simulate the case when RATS is used. We plot the observed packet delivery ratio in Figure 26 and the aggregate MAC throughput in Figure 27.

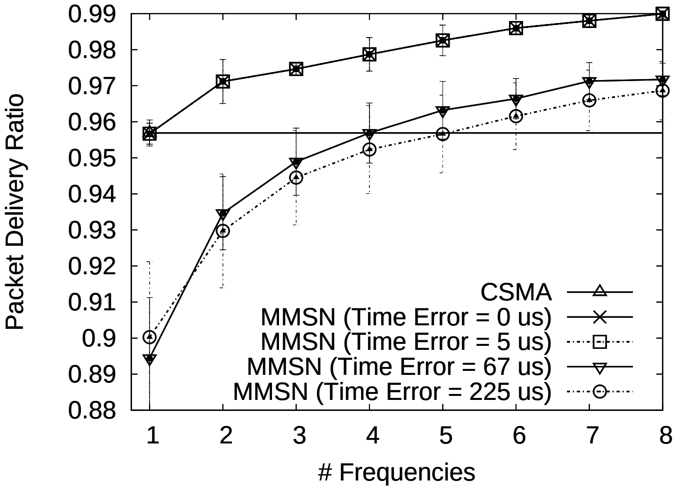


Fig. 26. Packet Delivery Ratio with Different Time Synchronization Errors

In Figure 26, we observe that the MMSN performance curve when time error is 5 *u*s almost completely overlaps with that when time error is 0. This observation informs us that, in a normal sensor network system with FTSP time synchronization, MMSN achieves close to ideal performance. MMSN is very tolerant to time synchronization errors that happen within a normally operating sensor network system. Even in a sensor system that has frequent node failures, FTSP reports that the time error can still be bounded by 67 *u*s, and our experimental result in Figure 26 shows that MMSN can still achieves above 90% packet delivery ratio. Although MMSN's packet delivery ratio slightly decreases compared to that when there is no time error, the performance difference between the two cases also decreases when more physical frequencies are available. For example, when 1 physical frequency is used, the difference between MMSN' packet delivery radio when time error is 5 *u*s and that when there is no time error is 2.7%. But when 7 physical frequencies are used, the difference decrease to 1.7%. The 37% decrease demonstrates that the load balance brought by MMSN's frequency diversity design can actually reduce the impact of time synchronization errors. Similar observations can also be obtained when the RATS time synchronization protocol is used to achieve greatly reduced time synchronization overhead with reduced time accuracy.

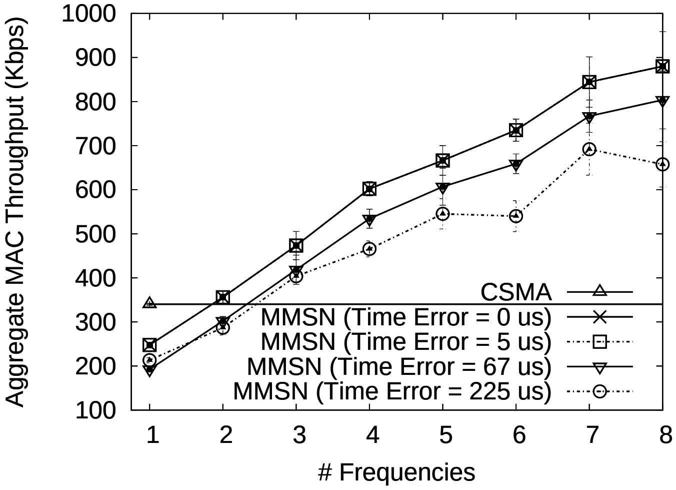


Fig. 27. Aggregate Throughput with Different Time Synchronization Errors

According to our experimental data in Figure 26, no matter there is no time error, or the time error is 5 *us* when FTSP is used in a normal system, or the time error is 67 *us* when FTSP is used in a high node-failure system, or the time error is 225 *us* when RATS is used for an ultra low duty-cycle system, MMSN always achieves a higher aggregate MAC throughput than CSMA. This fully demonstrates MMSN's superior performance compare to conventional solutions, even when a wide range of time synchronization errors are present.

In Figure 26, we also observe that MMSN's performance is scalable even when different time errors exist. Consider the case when the maximum time error is 67 *us* for FTSP to get adapted to high node failure, MMSN's MAC throughput increases from 192.5Kbps with 1 frequency, to 418.4Kbps with 3 frequencies, to 606.8Kbps with 5 frequencies, and to 803.8Kbps with 8 frequencies. MMSN keeps getting higher and higher MAC throughput when more physical frequencies are used. This observation also holds when the time error is 0, 5 *us*, 67 *us* or 225 *us* for tolerating a wide range of time errors observed in running systems. Since 67 *us* and 225 *us* time errors represent two extreme cases in sensor systems: high node failure and ultra low duty-cycle, respectively, the time errors for a large part of sensor network systems will lie in between the lines of 225 *us* and 5 *us*. As shown in Figure 26, any performance data lying between these two lines is still superior than that of conventional CSMA solution, which informs that in most sensor network systems, MMSN is feasible as well as beneficial.

5. RELATED WORK

In wireless sensor networks, media access control has received intense research attention, and a number of solutions have been proposed [Ahn et al. 2006] [Buettner et al. 2006] [Ye et al. 2006] [Klues et al. 2007] [Polastre et al. 2004] [Ye et al. 2002] [Rajendran et al. 2003] [Dam and k. Langendoen 2003] [Woo and Culler 2001] [El-Hoiyi et al. 2004]. While these solutions work well when one physical frequency is used, parallel data transmission when multiple frequencies are available is not considered. To address multi-frequency issues in sensor network MAC, a preliminary version of our work [Zhou et al. 2006] was published in IEEE INFOCOM 2006. We are also aware that an industrial solution called TSMP [Dust Networks] exists, which, instead of using frequency assignment, requires a connection survey of entire network on every physical frequencies. Rather than developing MAC layer solutions, [Xu et al. 2007] and [Wood et al. 2007] propose to use frequency diversity to defend jamming attacks, and [Le et al. 2007] use frequency diversity as well as control theory to optimize network throughput in a forest based data collection application.

In the general wireless network, frequency diversity has been studied for years and a significant number of multi-frequency MAC protocols have been proposed. However, these protocols are not suitable for typical WSN applications. First, to save energy and reduce product cost, each sensor device is usually equipped with a single radio transceiver. This single transceiver can not transmit and receive at the same time, nor can it function on different frequencies simultaneously. This restricted hardware is quite different from more powerful hardware assumed in other wireless systems. For example, protocols [Tang and Garcia-Luna-Aceves 1999] [Tzamaloukas and Garcia-Luna-Aceves 2000] are designed for frequency hopping spread spectrum (FHSS) wireless cards, and protocol [Deng and Haas 1998] assumes the busy-tone functionality on the hardware. In protocols [Nasipuri et al. 1999] [Wu et al. 2000] [Nasipuri and Das 2000] [Caccaco et al. 2002], the hardware is assumed to have the ability to listen to multiple frequencies at the same time. Second, the network bandwidth in WSNs is very limited and the MAC layer packet size is very small, 30∼50 bytes, compared to 512+ bytes used in general wireless ad hoc networks. Due to the small data packet size, the RTS/CTS control packets in IEEE 802.11 [IEEE 802.11 1999] no longer constitute a small overhead that can be ignored. So protocols [Bahl et al. 2004] [Raniwala and Chiueh 2005] [Adya et al. 2004] [Fitzek et al. 2003] [Li et al. 2003] that are based on IEEE 802.11, and protocols [So and Vaidya 2004] [Jain and Das 2001] [Tzamaloukas and Garcia-Luna-Aceves 2000] [Tang and Garcia-Luna-Aceves 1999] that use RTS/CTS for frequency negotiation are not suitable for WSN applications, even though they exhibit good performance in general wireless ad hoc networks. Different from all the above solutions, the MMSN protocol we develop in this paper resolves the unique hardware and application challenges brought by sensor network applications.

Instead of assigning different frequencies to neighboring nodes in an infrasctructureless wireless sensor network, in an infrastructure based cellular network, different frequencies are assigned to different cellular towers that use very high sending powers and usually form a hexagon graph. In a cellular network, the frequency assignment is normally formulated as a multicoloring problem on a varying weight graph [Janssen et al. 2000], and distributed algorithms are proposed to address it. To improve flexibility and reduce computational complexity, [Battiti et al. 2001] models it as an optimization problem. It first provides a localized solution, then extends the solution to a distributed one by ensuring mutual exclusion. In [Crescenzi et al. 2004], a generic solution template is proposed and applied to cellular frequency assignment for temporary tasks with a load balancing constraint. Since all these are developed for high power, infrastructure based, and normally hexagon topology cellular networks, they do not directly address the hardware and application challenges in wireless sensor networks.

6. CONCLUSIONS

In this paper, we develop the first multi-frequency MAC protocol for wireless sensor network applications, in which each device adopts a single radio transceiver. The different MAC design requirements for wireless sensor networks and general wireless ad hoc networks are compared, and a complete WSN multi-frequency MAC design (MMSN) is put forth. During the MMSN design, we analyze and evaluate different choices for frequency assignments, and also discuss the non-uniform back-off algorithms for the slotted media access design. Finally, we evaluate MMSN's performance through extensive experiments, and the performance results show that MMSN exhibits the prominent ability to utilize parallel transmissions among neighboring nodes. When multiple physical frequencies are available, MMSN also achieves increased energy efficiency, demonstrates the ability to work against radio interference and the tolerance to a wide range of measured time synchronization errors.

ACKNOWLEDGEMENT

This work is supported by the National Science Foundation, under grant CNS-0435060, grant CCR-0325197 and grant EN-CS-0329609.

APPENDIX A

Channel switching time is measured as the time length it takes for motes to successfully switch from one channel to another. This parameter impacts the maximum network throughput, because motes cannot receive or send any packet during this period of time, and it also affects the efficiency of toggle snooping in MMSN, where motes need to sense through channels rapidly. In this appendix, we measure the channel switching time of Micaz [CROSSBOW 2008] sensor devices. In our experiments, one mote alternatingly switches between channel 11 and 12. Every time after the node switches to a channel, it sends out a packet immediately, and then changes to a new channel as soon as the transmission is finished. We also put two other motes in the transmission range of the sender, which are tuned to channel 11 and 12, respectively, to receive packets. These two motes are used to verify the success of channel switching and packet transmissions. During the experiments, the carrier sensing and back-off operations in the TinyOS [Hill et al. 2000]MAC layer are disabled. We measure the number of packets the test mote can send in 10 seconds, denoted as N_1. In contrast, we also measure the same value of the test mote without switching channels, denotes as N_2. We calculate the channel switching time s as:

$$s = \frac{10}{N_1} - \frac{10}{N_2}$$

By repeating experiments 100 times, we get the average channel switching time of Micaz motes: 24.3 us. We then conduct the same experiments with different Micaz motes, as well as experiments with the transmitter switching from channel

11 to other channels. In both scenarios, the channel switching time does not have obvious changes. (In our experiments, all values are in the range of 23.6 $us \sim 24.9$ us.)

We also measure the minimum time a MicaZ mote takes to send one packet, which is around 1.3 ms $\sim$ 1.4 ms. So, the channel switching time is only 1.8% of the transmission time and degrades the bandwidth by 1.77%. Therefore the channel switching time does not have a significant impact on network performance even with frequent switching.

APPENDIX B

The primary consumer of energy in WSNs is idle listening. The key to reduce idle listening is executing low duty-cycle on nodes. Two primary approaches are considered in controlling duty-cycles in the MAC layer. One primary approach is to use sleep/awake *scheduling*, such as S-MAC [Ye et al. 2002], T-MAC [Dam and k. Langendoen 2003] and TRAMA [Rajendran et al. 2003]. Another one is to use low power listening, which allows a sleeping node to check channel activity with a very short, low power "channel active" probes, such as B-MAC [Polastre et al. 2004], WiseMAC [El-Hoiydi and Decotignie 2004]and SCP-MAC [Ye et al. 2006]. We find that our MMSN can work well with these protocols. In the detail, firstly, nodes must be awake during Frequency Assignment phase, because it requires nodes to correctly exchange information among neighbors to compute a good assignment. Since Frequency Assignment is only executed once during system initialization or very infrequently at runtime, this phase does not cost too much energy even without low duty-cycle. After the frequency assignment, nodes start to execute duty-cycles. If nodes use scheduled low duty-cycle, we can add a small period of time for toggle snooping and toggle transmission at the beginning of each non-sleeping time slot, which can guarantee that MMSN operates correctly. If nodes use LPL for low duty-cycle communications, MMSN requires that on one side a sleeping node uses toggle snooping-based "channel active" probes rather than a single channel probe, and on the other side a sender uses a long preamble with the length of at least the probing period. In short, low duty-cycle techniques can be integrated into MMSN without significant modifications.

Another problem with low duty-cycle is the loss of synchronization due to clock skew, which may happen frequently with cheap sensor devices. In order to overcome this, we can periodically run the time synchronization protocol, RTAS [Ganeriwal et al. 2007], to maintain synchronization. As mentioned in Section 4, RTAS can be used for an ultra low duty-cycle system, and is reported to maintain a 225 us error bound in an ultra low duty-cycle system. In Section 4.6, our experiment results have shown that MMSN can achieve good performance with such synchronization errors.

REFERENCES

Adya, A., Bahl, P., Padhye, J., A.Wolman, and Zhou, L. 2004. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *IEEE Broadnets 2004*.

Ahn, G.-S., Miluzzo, E., Campbell, A. T., Hong, S. G., and Cuomo, F. 2006. Funneling-MAC: A Localized, Sink-Oriented MAC for Boosting Fidelity in Sensor Networks. In *ACM SenSys*.

Akyildiz, I. F., Melodia, T., and Chowdhury, K. R. 2007. A Survey on Wireless Multimedia Sensor Networks. *Computer Networks*.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. 2002. Wireless Sensor Networks: a Survey. *Computer Networks*.

Bahl, P., Chancre, R., and Dungeon, J. 2004. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In *ACM MobiCom*.

Bao, L. and Garcia-Luna-aceves, J. J. 2001. A New Approach to Channel Access Scheduling for Ad Hoc Networks. In *ACM MobiCom*.

Battiti, R., Bertossi, A. A., and Brunato, M. 2001. Cellular Channel Assignment: A New Localized and Distributed Strategy. In *Mobile Networks and Applications*.

Buettner, M., Yee, G. V., Anderson, E., and Han, R. 2006. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. In *ACM SenSys*.

Caccaco, M., Zhang, L. Y., Sha, L., and Buttazzo, G. 2002. An Implicit Prioritized Access Protocol for Wireless Sensor Networks. In *IEEE RTSS*.

Cao, Q. and Abdelzaher, T. 2006. Scalable logical coordinates framework for routing in wireless sensor networks. *ACM Transactions on Sensor Networks*.

Cerpa, A., Wong, J. L., Kuang, L., Potkonjak, M., and Estrin, D. 2005. Statistical Model of Lossy Links in Wireless Sensor Networks. In *ACM/IEEE IPSN*.

Chang, J. and Maxemchuk, N. F. 1984. Reliable Broadcast Protocols. In *ACM Transactions on Computer Systems*.

Crescenzi, P., Gambosi, G., and Penna, P. 2004. On-line algorithms for the channel assignment problem in cellular networks. In *Discrete Applied Mathematics*.

CROSSBOW 2008. XBOW Sensor Motes Specifications. http://www.xbow.com.

Culler, D., Estrin, D., and Srivastava, M. 2004. Overview of Sensor Networks. In *IEEE Computer, Special Issue on Sensor Networks*.

Dam, T. and k. Langendoen. 2003. An Adaptive Evergy-Efficient MAC Protocol for Wireless Sensor Networks. In *ACM SenSys*.

Deng, J. and Haas, Z. 1998. Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks. In *IEEE ICUPC*.

Dust Networks. Technical Overview of Time Synchronized Mesh Protocol (TSMP). http://www.dustnetworks.com.

El-Hoiydi, A. and Decotignie, J.-D. 2004. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of InfrastructureWireless Sensor Networks. In *IEEE ISCC)*.

El-Hoiyi, A., Decotignie, J.-D., and Hernandez, J. 2004. Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. In *The Fifth European Wireless Conference*.

Elson, J., Girod, L., and Estrin, D. Fine-Grained Network Time Synchronization using Reference Broadcasts. In *USENIX ODSI), year = 2002*.

Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. 1999. Next Century Challenges: Scalable Coordination in Sensor Networks. In *ACM MobiCom*.

Fitzek, F., Angelini, D., Mazzini, G., , and Zorzi, M. 2003. Design and Performance of an Enhanced IEEE 802.11 MAC Protocol for Multihop Coverage Extension. In *IEEE Wireless Communications*.

Ganeriwal, S., Ganesan, D., Sim, H., Tsiatsis, V., and Srivastava, M. 2007. Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks. In *ACM SenSys*.

Ganeriwal, S., Kumar, R., and Srivastava, M. B. 2003. Timing-sync Protocol for Sensor Networks. In *ACM SenSys*.

Ganti, R. K., Jayachandran, P., Abdelzaher, T. F., and Stankovic, J. A. 2006. SATIRE: A Software Architecture for Smart AtTIRE. In *ACM MobiSys*.

Harvard CodeBlue 2008. CodeBlue: Sensor Networks for Medical Care. http://www.eecs.harvard.edu/ mdw/proj/codeblue/.

He, T., Krishnamurthy, S., Stankovic, J. A., Abdelzaher, T. F., Luo, L., Stoleru, R., Yan, T., Gu, L., Hui, J., and Krogh, B. 2004. Energy-Efficient Surveillance System Using Wireless Sensor Networks. In *ACM MobiSys*.

He, T., Stankovic, J. A., Lu, C., and Abdelzaher, T. F. 2003. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *IEEE ICDCS*.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. 2000. System Architecture Directions for Networked Sensors. In *The Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*.

IEEE 802.11 1999. IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. ANSI/IEEE Std. 802.11.

Jain, N. and Das, S. R. 2001. A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks. In *IEEE IC3N*.

Janssen, J., Krizanc, D., Narayanan, L., and Shende, S. 2000. Distributed Online Frequency Assignment in Cellular Networks. In *Journal of Algorithms*.

Karp, B. 2000. Geographic Routing for Wireless Networks. Ph.D. thesis, Harvard University, Cambridge, MA.

Kidd, C. D., Orr, R. J., Abowd, G. D., Atkeson, C. G., Essa, I. A., Macintyre, D., Mynatt, E., Starner, T. E., and Newstetter, W. 1999. The aware home: A Living Laboratory for Ubiquitous Computing Research. In *CoBuild*.

Klues, K., Hackmann, G., Chipara, O., and Lu, C. 2007. A Component Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks. In *ACM SenSys*.

Le, H. K., Henriksson, D., and Abdelzaher, T. F. 2007. A Control Theory Approach to Throughput Optimization in Multi-Channel Collection S ensor Networks. In *ACM/IEEE IPSN*.

Li, J., Haas, Z. J., Sheng, M., , and Chen, Y. 2003. Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Network. In *IEEE AINA*.

Li, Q., Stankovic, J. A., Hanson, M., Barth, A., Lach, J., and Zhou, G. 2009. Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information. In *BSN*.

Li, X., Kim, Y. J., Govindan, R., and Hong, W. 2003. Multi-dimensional Range Queries in Sensor Networks. In *ACM SenSys*.

Li, Y., Wu, H., Perkins, D., Tzeng, N.-F., and Bayoumi, M. 2003. MAC-SCC: Medium Access Control with a Separate Control Channel for Multihop Wireless Networks. In *IEEE ICDCSW*.

Lu, C., Blum, B., Abdelzaher, T., Stankovic, J., and He, T. 2002. RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks. In *IEEE RTAS*.

Lucarelli, D. and Wang, I.-J. Decentralized Synchronization Protocols with Nearest Neighbor Communication. In *ACM SenSys), year = 2004*.

Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson., J. 2002. Wireless Sensor Networks for Habitat Monitoring. In *ACM WSNA*.

Maróti, M., Kusy, B., Simon, G., and Á. Lédeczi. 2004. The Flooding Time Synchronization Protocol. In *ACM SenSys*.

MIThril 2008. MIThril. http://www.media.mit.edu/wearables/mithril/.

Nasipuri, A. and Das, S. R. 2000. Multichannel CSMA with Signal Power-based Channel Selection for Multihop Wireless Networks. In *IEEE Vehicular Technology Conference*.

Nasipuri, A., Zhuang, J., and Das, S. R. 1999. A Multichannel CSMA MAC Protocol for Multihop Wireless Networks. In *IEEE WCNC*.

Natarajan, A., Motani, M., de Silva, B., Yap, K., and Chua, K. C. 2007. Investigating Network Architectures for Body Sensor Networks. In *HealthNet*.

Polastre, J., Hill, J., and Culler, D. 2004. Versatile Low Power Median Access for Wireless Sensor Networks. In *ACM SenSys*.

Polastre, J., Szewczyk, R., and Culler, D. 2005. Telos: Enabling Ultra-Low Power Wireless Research. In *ACM/IEEE IPSN/SPOTS*.

Rajendran, V., Obraczka, K., and Garcia-Luna-Aceves, J. 2003. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. In *ACM SenSys*.

Raniwala, A. and Chiueh, T. 2005. Architecture and Algorithm for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In *IEEE INFOCOM*.

Shah, R. C., Nachman, L., and Wan, C.-Y. 2008. On the performance of Bluetooth and IEEE 802.15.4 radios in a body area network. In *BSN*.

Simon, G., Maróti, M., Á. Lédeczi, Balogh, G., Kusy, B., Nádas, A., Pap, G., Sallai, J., and Frampton, K. 2004. Sensor Network-Based Countersniper System. In *ACM SenSys*.

So, J. and Vaidya, N. 2004. Multi-Channel MAC for Ad-Hoc Networks: Handling Multi-Channel Hidden Terminal Using A Single Transceiver. In *ACM MobiHoc*.

Tang, Z. and Garcia-Luna-Aceves, J. 1999. Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks. In *IEEE INFOCOM*.

Tay, Y. C., Jamieson, K., and Balakrishnan, H. 2004. Collision-Minimizing CSMA and Its Applications to Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications*.

Tourrilhes, J. 1998. Robust Broadcast: Improving the Reliability of Broadcast Transmissions on CSMA/CA. In *IEEE PIMRC*.

Tzamaloukas, A. and Garcia-Luna-Aceves, J. J. 2000. Channel-Hopping Multiple Access. In *IEEE ICC*.

UFL Smart House 2008. UFL Smart House. http://www.rerc.ufl.edu.

UVA Smart House 2008. UVA Smart House. http://marc.med.virginia.edu/.

Vollset, E. and Ezhilchelvan, P. 2003. A Survey of Reliable Broadcast Protocols for Mobile Ad-Hoc Networks. In *Technical Report CS-TR-792, University of Newcastle upon Tyne*.

WernerAllen, G., Tewari, G., Patel, A., Welsh, M., and Nagpal, R. 2005. FireflyInspired Sensor Network Synchronicity with Realistic Radio Effects. In *ACM SenSys*.

Woo, A. and Culler, D. 2001. A Transmission Control Scheme for Media Access in Sensor Networks. In *ACM MobiCom*.

Woo, A., Tong, T., and Culler, D. 2003. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *ACM SenSys*.

Wood, A. D., Stankovic, J. A., and Zhou, G. 2007. DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks. In *IEEE SECON*.

Wu, S.-L., Liu, C.-Y., Tseng, Y.-C., and Shen, J.-P. 2000. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *I-SPAN*.

Xu, N., Rangwala, S., Chintalapudi, K. K., Ganesan, D., Broad, A., Govindan, R., and Estrin, D. 2004. A Wireless Sensor Network For Structural Monitoring. In *ACM SenSys*.

Xu, W., Trappe, W., and Zhang, Y. 2007. Channel Surfing: Defending Wireless Sensor Networks from Interference. In *ACM/IEEE IPSN*.

Ye, W., Heidemann, J., and Estrin, D. 2002. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *IEEE INFOCOM*.

Ye, W., Silva, F., and Heidemann, J. 2006. Ultra-Low Duty Cycle MAC with Scheduled Channel Polling. In *ACM SenSys*.

Zeng, X., Bagrodia, R., and Gerla, M. 1998. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *The 12th Workshop on Parallel and Distributed Simulations*.

Zhao, J. and Govindan, R. 2003. Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. In *ACM SenSys*.

Zhou, G., He, T., Krishnamurthy, S., and Stankovic, J. A. 2004. Impact of Radio Irregularity on Wireless Sensor Networks. In *ACM MobiSys*.

Zhou, G., He, T., Krishnamurthy, S., and Stankovic, J. A. 2007. Models and Solutions for Radio Irregularity in Wireless Sensor Networks. In *ACM Transactions on Sensor Networks*.

Zhou, G., He, T., Stankovic, J. A., and Abdelzaher, T. F. 2005. Radio Interference Detection in Wireless Sensor Networks. In *IEEE INFOCOM*.

Zhou, G., Huang, C., Yan, T., He, T., Stankovic, J. A., and Abdelzaher, T. F. 2006. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks. In *IEEE INFOCOM*.

Zhou, G., Lu, J., Wan, C.-Y., Yarvis, M. D., and Stankovic, J. A. 2008. BodyQoS: Adaptive and Radio-Agnostic QoS for Body Sensor Networks. In *IEEE INFOCOM*.