Chameleon: On the Energy Efficiency of Exploiting Frequencies in Duty-Cycled Wireless Sensor Networks

Jing Li and Anish Arora
Department of Computer Science & Engineering
The Ohio State University
{ljing,anish}@cse.ohio-state.edu

ABSTRACT

We consider the energy efficiency of medium access control (MAC) in low power wireless communication where multiple channels are available and the duty cycle of (send, receive, and idle) channel access is controllable. We show that in this setting maximization of MAC energy efficiency reduces to maximizing the aggregate channel utilization and minimizing the aggregate duty cycle channel access. Based on the reduction, we show the theoretical existence of centralized, global information protocols which achieve optimal energy efficiency in terms of channel assignment and duty cycle scheduling. Then, towards practically realizing these protocols in a distributed fashion with local information only, we present Chameleon, which assigns channels based on on-demand estimation of channel utilization and which adapts the duty cycle of node reception relative to the incoming traffic. Chameleon notably improves energy efficiency and channel utilization among users internal to the network, but also in the presence of external users that share the spectrum. We compare Chameleon with several representative multi-channel protocols, including static channel assignment and homogeneous distribution of duty cycle schemes, under different traffic scenarios. Our simulation and experimental results demonstrate a significant performance improvement over other proposed multi-channel protocols.

1. INTRODUCTION

Energy constraints in wireless sensor networks mandate efficiency of energy spent on communication, sensing energy, as well as computing. While a good rule of thumb is to design applications whose energy consumption is equal across these three categories, communication energy has dominated in early network deployments. The motivation to particularly improve communication efficiency has only increased as the growth in application complexity to date has by far outstripped the growth in available energy.

At the MAC layer, many proposals have considered schemes for almost-always-off communication. Duty cycling is the norm in modern deployments. Ideally the duty cycle should be at a rate that is just sufficient to accommodate the traffic. The choice of the MAC protocol and the duty cycle determine the resulting energy efficiency of communication at the MAC layer. The coordination and power management design in extant MAC protocols fall into two categories: sender centricity and receiver centricity [5]. In the former, the sender wakes up all the potential receivers during the transmission even if the message is unicast. In contrast, receiver centricity means that the sender must follow the wakup schedule of receiver. In this case, only nodes that have packets towards the receiver become active to conduct transmissions. In addition, recent technologies on sensor platforms show a significant growth in listen/receiver power due to increase in receiver complexity. Thus, the receiver-centric design intrinsically yields higher energy efficiency than that of sender centricity. In this paper, we consider achievable energy efficiency of receiver-centric duty-cycled MAC operation in networks where multiple channels (equivalently, frequencies) can be exploited.

The few multi-channel protocols that have been proposed in recent years are essentially categorized into three approaches: 1) Statically partition network nodes across multiple channels so that the density of nodes on a given channel is reduced, e.g., MMSN [19] and TMCP [15]; 2) Probabilistically migrate network nodes at runtime from one channel to another so as to balance traffic load, using control theoretic techniques, e.g., [12] and [11]; and 3) Balance traffic load (deterministically or randomly) across multiple channels evenly so as to reduce potential interference, e.g., Y-MAC [10] for sensor networks and SSCH [3] (which is designed for more general wireless networks).

All of these approaches significantly improve network goodput and, in turn, energy efficiency, in comparison with MACs that use only one single channel. Several extant protocols do not per se consider duty cycling, but even if one were to include duty cycling along with these approaches, careful examination reveals an opportunity for significant further improvements in goodput and energy efficiency. In the first approach, different channels are assigned to two-hop neighbors to avoid the possibility of interference; since the actual traffic is not considered, it is possible that some channels are lightly loaded and the node partitioning is thus too conservative. The approach moreover incurs the overhead in distributed distance-2 coloring. The second approach starts off by utilizing one channel and alleviates unfairness by probabilistically allocating a fraction of nodes into the next channel. In other words, channel utilization is expanded gradually when the goodput drops to a certain empirical threshold as measured in terms of Packet Reception Ratio or percentage of successful channel accesses. Nevertheless the goodput over the available channels is not optimized, nor is the instantaneous condition of every channel is taken into account when nodes perform channel switching. As for the third approach, although splitting traffic loads evenly over multiple channels achieves fairness, the aggregate goodput of the network is again not necessarily maximized.

While we discuss related work at more length later, we note in particular that none of these approaches choose channels based on a comprehensive (albeit local) view of the current condition of all channels. Thus, the channels to which nodes are switched into may not represent the best choice. This is especially true if one would take into account the interference that would result from the concurrent operation of external networks. Selecting channels based on a locally comprehensive yet efficiently computed view serves as the starting point for our design of a multi-channel MAC protocol, Chameleon [1], which combines adaptive duty cycling and spectrum resource allocation towards the goal of maximizing energy efficiency.

The main contributions of the paper are as follows.

- We formalize the optimization of MAC energy efficiency where duty cycling and multiple channel utilization is possible. We show that the optimization reduces to maximizing the spectrum utilization over all available channels while minimizing the duty cycle.
- Assuming the existence of procedures for (a) precisely quantifying node utilization on each channel and (b) for minimizing the send-receive-idle duty cycle for a given node traffic, we provide a protocol that optimizes MAC energy efficiency.
- We design Chameleon that implements this optimal protocol but relies on approximate, lightweight implementation of the procedures for (a) and (b). The former uses a light-weight metric W which is passively computed at each receiver node. The latter uses a pseudo-random scheduling of receiver-centric MAC; it chooses the receiver duty cycle to be just enough such that the receiver experiences low sender collision rate. A side-effect of this approach is that Chameleon intrinsically accommodates external interference.
- We show, using simulation, that Chameleon is capable of maintaining high energy efficiency under varied traffic scenarios.
- We validate, using experiments on the TelosB mote platform, not only the performance of the metric W, but the overall performance of the protocol, which compare favorably with canonical multi-channel MAC protocols.

The receiver-centric operation of Chameleon merits further explanation. Each node decides its wake-up times as well as the number of slots for reception (also called the "schedule"). Receiver schedules are advertised in the neighborhood periodically by a Neighbor Discovery protocol, so potential senders know at what time(s) they may wakeup to send a message to a given receiver. Many existent neighbor discovery protocols are applicable to this end, e.g., [7, 4, 18].

The rest of this paper is organized as follows. We present, in Section 2, the system model as well as an analysis of energy efficiency optimization. We also discuss a solution approach for implementing an optimal protocol. We design our multi-channel protocol, Chameleon, in Section 3. In Section 4, we present simulation and experimental evaluations of relevant aspects. We discuss related work in Section 5 and our conclusions in Section 6.

2. PROBLEM STATEMENT AND ENERGY EFFICIENCY ANALYSIS

In this section, we first define channel utilization, spectrum utilization as well as energy efficiency. Subsequently, we discuss the maximization of energy efficiency given a network traffic load, in terms of duty cycling and expected spectrum utilization. This leads to a solution approach for designing our protocol.

2.1 System Model

The network consists of N energy-constrained half-duplex wireless sensor nodes. Radio operation of each node is represented by a contiguous sequence of frames. Each frame consists of a number of time slots; for ease of exposition, we let this number be a global constant. We define a node's *duty cycle*, implicitly over some number of frames, to be the percentage of the time slots, ψ, when its radio is active; $\psi \in [0, 1]$. A node's duty cycle is further decomposed into its transmit duty cycle, the percentage of the slots when its radio is transmitting, and its receive duty cycle, the percentage of the slots when its radio is in receive mode.

For a given node i, we refer to the packets that are sent to i as its "in-traffic", while packets that are not sent to i but are overheard by i or whose collision is overheard by i are its "interference traffic".

The cumulative wireless bandwidth that can be utilized by nodes denotes the network "spectrum". Spectrum is divided into several orthogonal "channels" (or "frequencies") such that communications on different channels either never or only barely interfere with each other (in practice, adjacent channels are typically not completely interference free from each other [2]). Within each channel, collisions may occur if wireless devices attempt to transmit simultaneously. Given a node, we define its interference set as the set of nodes at which a packet reception that is concurrent with the reception of another packet at the node may yield a collision at the latter (and possibly also at the former)[2]; we let η denote the size of the interference set of the given node. We let i, j, h range over nodes in the network and k range over channels of the spectrum.

[1]Recent research shows that chameleons change color not to camouflage themselves but to communicate. Their "bandwidth" of communication (aka signalling) is related to the number of colors that they use. Cf.: D. Stuart-Fox and A. Moussalli, "Selection for social signalling drives the evolution of chameleon colour change", *PLoS Biol* 6(1): e25, 2008.

[2]We note that several of our definitions are receiver-centric rather than sender-centric, as this significantly simplifies our exposition.

With respect to a given node and its interference set, we define the **channel utilization** for a given channel, k, as the ratio, $E(k)$, of the number of time slots where any one of the nodes successfully receives a packet to the total number of time slots. (The definition may be relativized to the number of frames considered in the definition of duty cycle.)

Consequently, **spectrum utilization** with respect to a node and its interference set of nodes denotes the overall successful transmissions in all channels over the total number of time slots normalized by the number of channels, M. Hence, spectrum utilization is defined as:

$$E_S = \frac{\sum_{k=1}^{M} E(k)}{M} \tag{1}$$

Our primary interest is in the metric of **energy efficiency**, which refers to the goodput for a given energy budget [5]. Basically, this metric refers to the ratio of the number of time slots with successful transmissions to the number of slots in which radios are active, albeit they are transmitting, idle, or active. Eq. (2) defines energy efficiency for a unicast scenario. Notation T in the formula is the total number of slots considered. Compared to channel and spectrum utilization, duty cycling of a node is taken into consideration in the metric.

$$E_E = \frac{\sum_{l=1}^{T}\sum_{j=1}^{N} Z_j^l}{\sum_{l=1}^{T}\sum_{j=1}^{N}(S_j^l + R_j^l)} \tag{2}$$

where

$$S_j^l = \begin{cases} 1, & \text{when node j transmits in slot l} \\ 0, & \text{when node j sleeps in slot l} \end{cases}$$

$$R_j^l = \begin{cases} 1, & \text{when node j listens in slot l} \\ 0, & \text{when node j sleeps in slot l} \end{cases}$$

$$Z_j^l = \begin{cases} 2, & \text{node j succeeds receiving its packets in slot l} \\ 0, & \text{otherwise} \end{cases}$$

2.2 Energy Efficiency Optimization

Problem Statement Given a node i, whose interference set has size η, and in-traffic p_i towards i, our goal is to schedule the in-traffic—i.e., choose channels and wakeup times for the i and the nodes sending packets to i— such that the resulting energy efficiency E_E is maximized.

We approach this problem by first simplifying Eq. (2). In Eq. (2), spectrum utilization reflects the goodput resulting from communications of the nodes in the interference range, corresponding to $\sum\sum Z_j^l$. It follows that $\sum\sum Z_j^l = 2TME_S$, where $2ME_S$ equals the aggregate spectrum utilization and the factor of 2 reflects benefits of both parties in a communication. Schemes for duty cycling control the energy consumption of nodes, which is $\sum\sum(S_j^l + R_j^l) = T\sum_{j=1}^{\eta}\psi_j$. Thus, the following equation is an equivalent representation of energy efficiency.

$$E_E = \frac{2ME_S}{\sum_{j=1}^{\eta}\psi_j} \tag{3}$$

In order to optimize E_E by maximizing E_S as well as minimizing $\sum_{j=1}^{\eta}\psi_j$, the scheduler has to choose channels and wakeup times. We will first consider channel selection that maximizes the expected E_S for traffic p_i, then we will discuss how to schedule the wakeup times of nodes to minimize $\sum_{j=1}^{\eta}\psi_j$.

Recall that $E(k)$ is the aggregate successful reception probability of each receiver in the interference set of the given node. For the purpose of analysis, in this subsection, we make two assumptions. One, the in-traffic of nodes follows a stationary process with uniform distribution of arrival times; let the in-traffic load at node i, denoted by p_i, be the probability that on average a packet is sent to i. And two, that the node and its interference set form a clique, i.e., each of these nodes can overhear each packet sent by another of these nodes; thus if packets are concurrently sent to different nodes, collisions will result at each receiver. Then, $E(k)$ at node i is computed by:

$$E(k) = \sum_j p_j \prod_{h\neq j}(1 - p_h) \tag{4}$$

where j and h range over these nodes. Initially, $E(k)$ increases as traffic loads increase. However, utilization decreases when the channel becomes overloaded, in which case collisions (or, in a contention based scheme, backoff procedures) dominate the communication.

LEMMA 1. *The expected channel utilization with respect to node i, $\hat{E}(k)$, is maximized when the aggregate traffic load in the interference set of i, $\sum_{j=1}^{\eta} p_j(k)$, increases to 1.*

PROOF. The average traffic load on channel k is calculated as $\hat{p}(k) = \sum_{j=1}^{\eta} p_j(k)/\eta$. Hence, by Eq. (4) the expected channel utilization $\hat{E}(k) = \eta\hat{p}(k)(1-\hat{p}(k))^{\eta-1}$. Since $\hat{p}(k) = 1/\eta$ implies that $\sum_{j=1}^{\eta} p_j(k) = 1$, it follows that maximal utilization occurs when $\sum_{j=1}^{\eta} p_j(k) = 1$. As the aggregate load increases up to 1, $\hat{E}(k)$ increases; after reaching 1, $\hat{E}(k)$ decreases as the aggregate load increases. It follows that the total traffic load should be 1 to achieve maximal utilization $\hat{E}(k)$. □

Next, we consider channel selection for load p_i. Theorem 1 states a sufficient condition for selecting channels for load p_i that maximizes $\hat{E_S}$.

Let $\bar{p}(k)$ be the current average load on each channel k. Given a channel k, define q as 1 minus the current total load on a given channel, i.e., $q = 1 - \eta\bar{p}(k)$. Let $\vec{q}$ be a vector of qs for all channels which is sorted in a nonincreasing order. Thus, $\vec{q}_s$ represents the sth greatest element in $\vec{q}$, corresponding to channel of index $C(\vec{q}_s)$. Let vector $\vec{\alpha} = \{\alpha(k) : k = 1, ..., M\}$ denote percentages of in-traffic allocated to each channel.

THEOREM 1. *$\hat{E_S}$ is optimized if we allocate traffic load p_i to channels according to fractions $\vec{\alpha}$ computed in Eq. (5).*

$$\alpha(C(\vec{q}_s)) = \begin{cases} \frac{\vec{q}_s}{p_i}, & p_i - \sum_{t=1}^{s-1}\vec{q}_t \geq \vec{q}_s \\ \frac{p_i - \sum_{t=1}^{s-1}\vec{q}_t}{p_i}, & 0 < p_i - \sum_{t=1}^{s-1}\vec{q}_t < \vec{q}_s \\ 0, & p_i - \sum_{t=1}^{s-1}\vec{q}_t \leq 0 \end{cases} \tag{5}$$

PROOF. $\vec{q}$ represents residual quota of load on each channel k. The essential idea in channel assignment is to fill up channels based on $\vec{q}$ sequentially, i.e., giving preference to those which have more residual capacity, until p_i has been assigned completely or all qs in $\vec{q}$ have been consumed.

Define Δp to be the smallest unit of load that can be assigned on a channel. Hence, load p_i consists of $p_i/\Delta p$ units. Before adding a unit Δp into channel k, the expected utilization on channel k is $\hat{E}(k) = \eta\bar{p}(k)(1-\bar{p}(k))^{\eta-1}$. After adding Δp, by Eq. (4), the expected utilization becomes $\hat{E}'(k) = \eta\bar{p}(k)(1-\Delta p)(1-\bar{p}(k))^{\eta-1} + \Delta p(1-\bar{p}(k))^{\eta}$. Thus, the utilization gain, $\Delta\hat{E}(k)$, on channel k after appending each Δp would be

$$\Delta\hat{E}(k) = \Delta p\,(1-(\eta+1)\bar{p}(k))(1-\bar{p}(k))^{\eta-1} \tag{6}$$

which is a monotone decreasing function of $\bar{p}(k)$. The smaller the $\bar{p}(k)$, the higher the utilization gain will be. Since Δp is an atomic unit, assigning the channel with lowest $\bar{p}(k)$ will provide the highest $\Delta\hat{E}_S$, where $\Delta\hat{E}_S = \Delta\hat{E}(k)$ and k is the channel assigned to the Δp load.

Ideally, all $p_i/\Delta p$ units would be added into the channel with lowest $\bar{p}(k)$ to maximize total utilization gain. According to Lemma 1, however, the total load on each channel k should not exceed 1 to achieve maximal utilization. $C(\vec{q}_s)$ denotes the channel which has sth lowest $\bar{p}(k)$ and $\vec{q}_s/\Delta p$ is the number of units that can be added to a given channel before exceeding the maximum. Therefore, sequentially filling up each channel in the order of $\vec{q}$ will maximize total $\hat{E}_S$.

Consider the assignment to channel $C(\vec{q}_s)$. The number of units of p_i that are yet to be assigned is $(p_i - \sum_{t=1}^{s-1}\vec{q}_t)/\Delta p$. If this number is non-positive, indicating that all units of p_i have been assigned on channels ahead, the fraction on this channel $\alpha(C(\vec{q}_s))$ is 0. Otherwise, if the number of unassigned units is less than $\vec{q}_s$, we can allocate all of $(p_i - \sum_{t=1}^{s-1}\vec{q}_t)/\Delta p$ units on channel $C(\vec{q}_s)$. $\alpha(C(\vec{q}_s)) = (p_i - \sum_{t=1}^{s-1}\vec{q}_t)/p_i$ in this case. If the number of unassigned units is not less than $\vec{q}_s$, we can fill up this channel with $\alpha(C(\vec{q}_s)) = \vec{q}_s/p_i$. □

Finally, we consider scheduling for duty cycle minimization. To this end, a " centralized TDMA and duty cycling" scheduler that has full information of the arrival times of all packets suffices. This scheduler (having scheduled the existing traffic in the network) can schedule packet communications so that no collisions in the set of nodes and senders and receivers are scheduled to wakeup exactly at these times. Lemma 2 states that nodes running the duty-cycled TDMA will minimize gross duty cycle $\sum_{j=1}^{\eta}\psi_j$.

LEMMA 2. *Given traffic load p_i and the arrival time of the in-traffic of i, the centralized TDMA and duty cycling scheduler minimizes the total duty cycle $\sum_{j=1}^{\eta}\psi_j$.*

PROOF. Duty cycles of nodes that are neither senders nor receivers of packets in the in-traffic of i will remain unchanged. As for nodes involved in the traffic, the scheduler trivially minimizes the wakeup times, since there are no superfluous sends or receives or idle slots. The total duty cycle consumed by the load p_i is minimized to be twice of the load, i.e., $2p_i$. □

2.3 Protocol Design Approach

Theorem 1 and Lemma 2 indicate that there exists in theory a centralized, global information scheduler for maximizing energy efficiency. The scheduler deals with one node, say i, at a time, and performs three tasks: (i) determines the in-traffic for i, (ii) splits the in-traffic across the channels according to Eq. (5), and (iii) compute the times at which each in-packet to i would be sent without contending with any of the packets scheduled thus far, as well as update the sleep-wakeup schedule of the nodes to accommodate waking up only when they are involve in transmitting or receiving each in-packet to i. Note that the packet transmission time scheduling in (iii) yields an in-traffic whose arrival time may no longer satisfy a uniform distribution, as assumed in the analysis above, but since (iii) enforces collision freedom, the expected E_S and E_E are not negatively affected.

This scheduler is of high complexity.[3] Towards realizing a practical MAC solution, we now discuss a distributed, light-weight protocol wherein each node i itself performs tasks (i)-(iii) in a manner that approximates the centralized scheduler.

2.3.1 Local Metric W for Channel Utilization

In task (i), computing the in-traffic load at node i (i.e., the exact instantaneous p_i value) involves collecting information from all nodes in the interference set of i. In task (ii), splitting the in-traffic involves collecting information about q, or alternatively, about $\sum_{j=1}^{\eta}p_j$, the in-traffic load at all nodes in the interference set of i. Rather than letting i actively coordinate with all nodes in its interference set to compute p_i and $\sum_{j=1}^{\eta}p_j$ for utilization optimization, we introduce the local metric W that is passively computed by i.

$$W = \psi_r + \exp(I) \tag{7}$$

The first part of W is the receive duty cycle of node i, ψ_r; it estimates p_i. The second part of W is an exponential function of interference level I, where I is the probability that an interferer of i needs to receive; it estimates $\sum_{j=1}^{\eta}p_j$. Thus, W estimates $p_i+\sum_{j=1}^{\eta}p_j$.

As we will motivate shortly, our design realizes task (iii) via a receiver-centric MAC. In such a MAC, each node chooses its times of reception such that nodes that wish to send to it are aware of its wakeup times and other nodes in its interference neighborhood do not (whp) wakeup to receive at these times. W is designed to be more efficiently computed than $p_i+\sum_{j=1}^{\eta}p_j$ in a receiver-centric MAC. In particular, the computation of W does not involve sending any specific information, in contrast with the Channel Access Ratio message used in many multi-channel protocols, such as [12]. The value of I is calculated passively by listening to the channel for a certain period of time.

Let S be the set of nodes that send packets to i. In a receiver-centric MAC, since nodes in S are aware of the times that i wakes up to receive, they can avoid receiving packets at those times. Hence I reduces to:

$$I = 1-(1-\bar{p_I})^{\eta-|S|} \tag{8}$$

[3]In an alternative scheduler, the load at all nodes would be scheduled simultaneously, and it would be possible to achieve even higher net energy efficiency, but it would involve a combinatorial optimization.

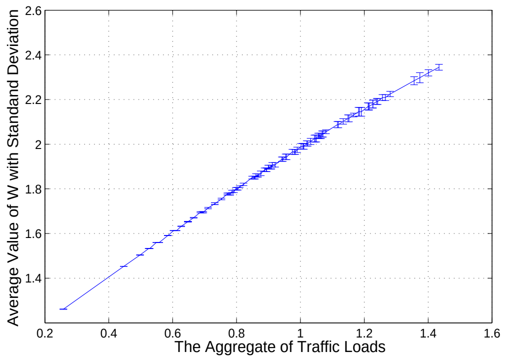


Figure 1: Mean W with std vs Metric $\sum p_i$

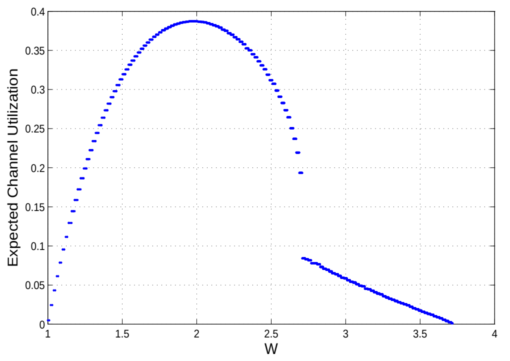


Figure 2: Expected Channel Utilization $\hat{E}(k)$ vs Metric W

where $\bar{p_I}$ denotes the average reception probability of a node whose reception can interfere at i.

Fig. 1 shows an instance of the relation between $\sum p_i$ and W. We consider a clique network where six pairs of nodes communicate independently with arbitrary traffic load in the range [0,1]. Each receiver locally computes the metric W according to Eq. (7) for various traffic scenarios. Fig. 1 plots the mean W and the standard deviation among six receivers versus the total traffic load $p_i+\sum_{j=1}^{\eta} p_j$. Note that the same value $p_i+\sum_{j=1}^{\eta} p_j$ corresponds to a few different sequences of traffic loads $\vec{p}$. It is observed that value W is approximately linear with $p_i+\sum_{j=1}^{\eta} p_j$, which verifies that W is a feasible indicator of $\hat{E}(k)$ in lieu of metric $p_i+\sum_{j=1}^{\eta} p_j$. In addition, the locally computed deviation of the Ws is very small, i.e., the average standard deviation shown in the figure is around 0.005.

Fig.2 shows the relation between W and $\hat{E}(k)$. In the equation for expected channel utilization, $\hat{E}(k) = \eta\bar{p}(1-\bar{p})^{\eta-1}$, (cf. Lemma 1) $\bar{p}$ also equals

$$\bar{p} = \frac{(\eta - |S|)\bar{p_I} + \psi_r}{\eta} \quad (9)$$

Since I estimates the aggregate receive duty cycle of nodes in the interference set and ψ_r estimates p_i, we expect that the construction of W will estimate $p_i+\sum_{j=1}^{\eta} p_j$ accurately. To validate this, we populate ψ_r and $\bar{p_I}$ with randomly chosen values in the range of [0,1] in formula W and $\hat{E}(k)$, and average the channel utilization over metric W. In Fig. 2, η and $|S|$ are configured to be 10 and 2, respectively. The figure shows that there exists a peak where expected channel utilization is maximized as W increases. We refer to the peak value of W as W^*, which corresponds to $p_i+\sum_{j=1}^{\eta} p_j = 1$. Thus, $W^* - \exp(I)$ estimates $p_i+\sum_{j=1}^{\eta} p_j - \exp(I)$, i.e., $1 - (\sum_{j=1}^{\eta} p_j)$, which means that $W^* - \exp(I)$ is useful as an estimator of q.

(We note that W^* is close to 2 in both figures. Another relevant observation from our analysis is that even when parameters such as η and $|S|$ change, W^* remains unchanged in the different configurations.)

2.3.2 *Receiver-Centric MAC with Duty Cycle Scheduler*

The algorithm associated with Task (iii) ensures interference freedom (collision avoidance) and sleep-wakeup schedule update at high cost. We adopt the receiver centric scheduling approach (as exemplified by O-MAC [5] and Crankshaft [8]) to locally and more efficiently avoid collision and schedule sleep-wakeup.

The basic idea that we borrow is this. Each receiver has a pseudo-random scheduler which determines its wakeup slots. The wakeup schedule is advertised to neighbors, compactly since essentially the pseudo-random seed needs to be shared, via a neighbor discovery process. When a node discovers this receiver, it also obtains this receiver's state (of pseudorandom generation), and thus the node is can generate the receiver's wakeup schedule. When the node wishes to send to the receiver, it wakesup at the next slot at which the receiver will be awake and attempts to communicate. Two basic modules, neighbor discovery and time synchronization, are used and in turn the module offers Send and Receive interfaces. Chameleon adopts these basic interfaces from those in O-MAC. This decentralized pseudo-random scheduling staggers nodes' wakeup times whp and has been proven to achieve higher energy efficiency than other sender-centric protocols [5], such as S-MAC, B-MAC, and others, for a variety of traffics.

In other words, in this receiver-centric scheme, the senders' wakeup times are implicitly scheduled. Since receivers wakeup at random times in each frame, the likelihood that two interfering receivers will simultaneously receive is low. However, if two or more senders attempt to simultaneously send to a common receiver, collisions may occur and goodput/energy efficiency will be reduced. In order to deal with this problem, we enhance the duty cycle scheduling of the receiver centric scheduler, as follows.

To respond to potential changes in its in-traffic, each receiver periodically updates its receive duty cycle to be consistent with its in-traffic load. To determine its in-traffic load, we let senders, 1..$|S|$ where S denotes the set of senders, piggyback (1 byte) information about their expected flow rates, $\lambda_1, \lambda_2, ...\lambda_{|S|}$, along with the packets they send to the receiver. Based on this information, the receiver scheduler calculates the expected number of slots, T_d, it needs to receive data during the next period. Since radio operation is a sequence of contiguous frames, letting T_f denote the length of terms of slots.

$$T_d = T_f \sum_{i=1}^{|S|} \lambda_i \tag{10}$$

In other words, each receiver only listens for T_d out of T_f time slots to receive its in-traffic energy-efficiently. If no packets come in, the receiver remains in its initial duty cycle ψ_{ini}, as specified by the application, in anticipation of potentially incoming data.

3. ENERGY EFFICIENT MULTI-CHANNEL PROTOCOL DESIGN

In this section, we detail the design of our energy efficient multi-channel access protocol, Chameleon. Since Section 2.3.2 has already overviewed the underlying receiver-centric MAC protocol and how the receiver duty cycle is computed (to minimize node duty cycle), we focus here on the protocol that realizes the channel assignment algorithm and that coordinates receivers with their senders. We conclude with an overview of our TinyOS implementation of Chameleon.

3.1 Channel Assignment Scheduler

The task of this scheduler is to distribute the traffic load, ψ_r, according to the algorithm described in Theorem 1. To this end, we explain, first, how interference level I is measured, then, how the local metric W is used in channel assignment to load fractions, and, lastly, how multiple channels are applied to receive wakeup schedule.

3.1.1 Channel Interference Level Measurement

Every receiver maintains a vector $\vec{I} = \{I(k) : k = 1, 2, ..., M\}$, representing the measured interference level on each channel. It measures in two ways: cumulatively and incrementally.

- The cumulative measure is made to determine the interference level on every channel. The receiver passively monitors a channel for T_m slots contiguously, where $T_m \geq T_f$. The measure is performed initially to support channel initialization, but also to support channel reassignment, as and when it needs to be performed.
- The incremental measure is made when receiving a packet on an assigned channel. This has no overhead, but it does not yield information about non-assigned channels.

The measurement of I may itself be performed in various ways, i.e., using the collision/overhearing ratio at the receiver and backoff ratio at the sender. Let C be the event that collision or overhearing happens at a receiver. We measure the probability of the event, $P(C)$, by the ratio of the number of slots with collided and overheard packets to the number of slots where the receiver listens. Note that the probability of collision from in-traffic senders is comparatively small due to the double backoff scheme explained in section 3.2. Hence, $I \approx P(C)$.

In our implementation, the receiver cumulatively records the number of interfered slots, which equals the total number of busy CCA (Clear Channel Assessment) slots minus the total number of slots where it received packets. In other words, it performs continuous CCA checks along with listening. After T_m slots, it computes I as the fraction of total number of interfered slots over the total number of checked slots.

3.1.2 Channel Assignment to Load

Algorithm 1, given $\vec{I}$, first computes the number of acceptable units on each channel, in $\vec{q}$ (lines 3 to 9). (Recall Δp is the smallest unit of load that can be assigned; it is set to 0.01 in our implementation.) Given ψ_r, lines 18 to 27 assign units to each channel according to Theorem 1, which results in a vector of size M, $\vec{V}$, e.g., $\vec{V} = (3, 7, 1, 0, ..., 0)$, where each element represents the units allocated to the channel. Thus, ψ_r is split to each channel in proportion to $\vec{V}$. (Which channels to use in which frame is discussed in the next subsection.) If the sum of the available capacity, $\sum_{k=1}^{M} q(k)$, is less than the total receive duty cycle ψ_r required, cf. line 11, senders are notified to reduce their outgoing traffic if possible.

Algorithm 1 *Duty Cycle Coloring*

```
Require: $\psi_r, \vec{I}, \vec{W}$
 1: if $\exists l : W(l) > W^{**}$ then
 2:     $Measure\ \vec{I}$;
 3:     for $k = 1$ to $M$ do
 4:         if $W^{*}- \exp(I(k)) \leq 0$ then
 5:             $q(k) \leftarrow 0$;
 6:         else
 7:             $q(k) \leftarrow \lceil \frac{W^{*} - \exp(I(k))}{\Delta p} \rceil$;
 8:         end if
 9:     end for
10:
11:     if $\sum_{k=1}^{M} q(k) < \lceil \frac{\psi_r}{\Delta p} \rceil$ then
12:         $Inform\ senders\ (optional)$;
13:     end if
14:
15:     $Sort\ \vec{q}\ in\ non-increasing\ order$
16:     $\vec{q} = (q_1, q_2, ..., q_M)$
17:     $the\ channel\ index\ of\ q_s\ is\ C(q_s)$;
18:     for $s = 1$ to $M$ do
19:         if $\lceil \frac{\psi_r}{\Delta p} \rceil - \sum_{t=1}^{s-1} q_t \geq q_s$ then $V(C(q_s)) \leftarrow q_s$;
20:         else
21:             if $0 < \lceil \frac{\psi_r}{\Delta p} \rceil - \sum_{t=1}^{s-1} q_t < q_s$ then
22:                 $V(C(q_s)) \leftarrow \lceil \frac{\psi_r}{\Delta p} \rceil - \sum_{t=1}^{s-1} q_t$;
23:             else
24:                 $V(C(q_s)) \leftarrow 0$;
25:             end if
26:         end if
27:     end for
28: end if
```

Theoretically, color reassignment should be conducted by a receiver as long as the incoming load changes. However, in a network with dynamic traffic loads, frequently performing cumulative measurements involves significant amount of energy consumption. To avoid the inefficiency, we enforce the condition that only when the estimated W of an existing channel (during an incremental measure) exceeds W^{**} will the receiver perform a cumulative measure and then color reassignment. Otherwise, it would continue to use the current channels. Thus, receivers only need to conduct a periodic checking on the channels currently in use. Line 1 states the applied condition, which is checked by nodes. W^{**} is an empirical upper bound of capacity range (e.g., $W^{**} = 2.3$) compared to the lower bound W^* (e.g., $W^* = 2$) for stabilization in implementation. If $W(l) > W^{**}$, indicating that the capacity on channel l has been exceeded, channel reassignment. Initially, every receiver starts with a duty cycle of ψ_{ini}. After conducting a cumulative measurement, each node allocates its load to the corresponding channels.

Dwell Phase

The dwell phase is introduced to alleviate fluctuations caused by channel switching. The receiver postpones updating its wake-up schedule for a randomly chosen number of frames (T_{dwell}). If monitored channels still satisfy the condition during the whole dwell phase, the node will finally carry out the change. Otherwise, it will not perform channel reassignment.

The function of dwell phase is to break possible synchrony of channel adaptation in the neighborhood. For example, two nodes might attempt to extend their duty cycle on frequency k simultaneously, resulting in both Ws exceeding the threshold. With the help of dwell phase, one node will probably update first and the other one will find out change of channel quality during its dwell time and recompute the coloring. Only if W remains less than W^{**} during the whole dwell phase, will the node eventually make the change. With this scheme, channel expansion would occur gradually and smoothly.

3.1.3 Assigning Channels to Frames

The scheduler also associates a channel with each frame. This channel will be used by the receiver in all slots in which it wakesup during that frame. We implement the association using a vector of units assigned to each channel, $\vec{V}$. Given an assignment $\vec{V}$, the receiver maintains a shadow copy $\vec{V'}$, which is initially set to $\vec{V}$. In each frame, it checks the next k in $\vec{V'}$. If the value of $V(k) > 0$, then channel k is used in the next frame and the current value in $\vec{V'}$ is decremented; otherwise, the next channel is checked until all values become 0. Then, $\vec{V}$ is copied to $\vec{V'}$ again and the above procedure repeated. In this way, nodes uses multiple channels in proportion to $\vec{V}$.

3.2 Sender-Receiver Coordination

There are two ways in which the receiver shares its updated channel assignment with senders: asynchronously, through the neighbor discovery process and, synchronously, through beaconing in the first wakeup slot at the beginning of each frame. In the former case, nodes independently compute each other's wake-up slot and channel. The updated colored wake-up schedule V has to be notified to neighbors via the discovery module within certain amount of time. Each sender keeps its own updated $\vec{V'}$ and the current index of the receiver, generating future wake-up slots and colors independently. We chose to implement the asynchronous neighbor discovery protocol, Disco [7], which schedules radio wake times at multiples of prime numbers, ensuring deterministic pairwise discovery and rendezvous latencies. Disco operates on a wellknown channel, called the home channel. We add

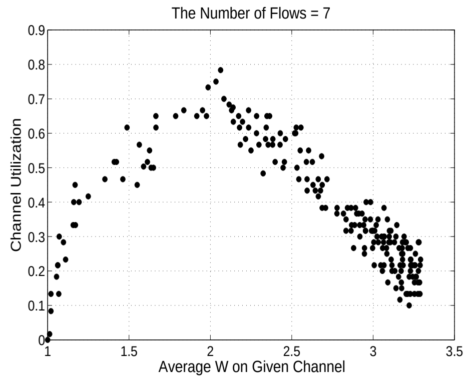


Figure 3: Channel Utilization vs. W in a Network of 7 Flows

several small pieces of information to the packets sent out by Disco, related to time synchronization, channel assignment, wakeup schedule and T_d. When a receiver starts to change its channel assignment schedule, it may accelerate propagating an channel update, by increasing the duty cycle of Disco. After exceeding the deterministic rendezvous period, Disco will go back to previous low duty cycle. However, the energy cost of updating schedules through Disco is nontrivial especially when frequent updates exist. Moreover, the discovery schedule may interrupt with node's listen schedule more frequently in this case.

In the latter case, status is updated by advertising receiver's current color and wake duration T_d at the beginning of each frame, using the home channel. Senders do not maintain any color schedule, instead they listen to the home channel at the wake slot of receiver. The receiver broadcasts its current T_d and the channel ID it is going to use in one short beacon on the home channel. After that potential senders and receiver all switch to the chosen channel for the rest of communication. Based on T_d information, senders further stagger their transmissions by randomizing competing slots during the listen period of receiver. Within each slot, senders utilize a small fixed backoff window. When a node fails in competition, it would continue to compete for the next slot. This "double backoff" scheme further reduces the chance of collision among in-traffic nodes. The total of beaconing and switch time takes approximately 5 ms.

3.3 Implementation

We implemented Chameleon using the UPMA framework in TinyOS on a network of TelosB sensor motes. The composition of Chameleon in the UPMA framework is shown in Fig. 4. We implemented Chameleon for the CC2420 radio platform, which is a packetizing radio used in popular TelosB and MicaZ motes; the code is readily ported to motes with streaming radios such as CC1000.

The SlotGenerator module in Fig. 4 provides the basic functionality of generating slotted frames continuously and the signalling slot event handler for other modules. The functionality was realized using the GenericSlotterC

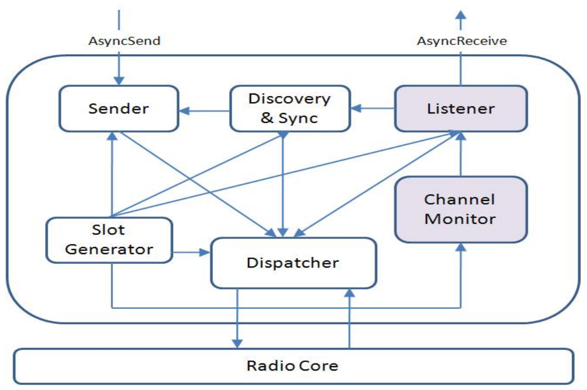


Figure 4: Composition of the Chameleon Protocol

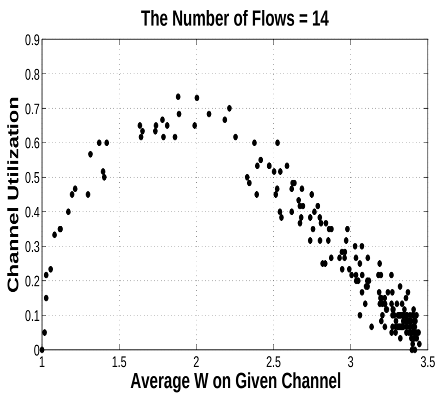


Figure 5: Channel Utilization vs. W in a Network of 14 Flows

provided by UPMA itself. The Listener module decides node wake-up times and durations, while Sender determines when to transmit application packets given the state maintained in the neighborhood table. The Discovery & Sync module performs relative slot synchronization on basis of asynchronous discovery (we implemented the Disco protocol); these processes have rather low overhead.

The ChannelMonitor module realizes the bulk of the functionality of Chameleon, including the periodic checking of W. and the channel selection (or, coloring). It also generates the colored wake-up schedule, which is input to the Listener module which implements the desired channel switching upon wakeup. Requests from all of these components are arbitrated by the Dispatcher according to priorities, via a state machine. In the diagram, colored components represent Chameleon modules which are new with respect to the original O-MAC protocol.

4. PROTOCOL EVALUATION

We evaluated Chameleon both via simulation (in Matlab) and experimentation (with an implementation in TinyOS 2.0.2 on TelosB motes [1] to understand and compare its performance.

4.1 Simulation Evaluation

We present four groups of results. The first evaluates the relation between channel utilization and capacity indicator W at different node densities, to corroborate the trend predicted by earlier analysis (see Fig. 1). The second evaluates the effectiveness of $W^{*}-\exp(I)$ as an estimate the residual channel capacity q in choosing how to distribute loads over channels. The third and the fourth respectively compare the energy efficiency and tolerance to interference from external networks of Chameleon versus benchmark protocols. Specifically, our comparison is based on selecting and implementing improved versions of the Y-MAC [10] and the MMSN [19] protocols, and simulations of various traffic load scenarios.

In all simulations, the frames consist of 100 slots. The amount of data that can be transmitted in a slot is fixed (at 80 bytes), hence, senders first accumulate data till it reaches that size and then start to transmit them when possible. All data communications performed are in unicast. The size of backoff window in each slot is configured as 4. Traffic rates at senders change periodically.

4.1.1 *Channel Utilization vs Metric W*

Our simulations used a clique network with several overlapping pairs of senders and receivers. After every X frames ($X = 50$), receivers calculated the average W based on their receive duty cycle and measurements of I. We plot the relation between the average value of Ws in the network and aggregate utilization on the given channel. In Fig. 3, there are seven pairs of senders and receivers, and each sender transmits at certain arbitrary low rate initially. Similarly, Fig. 5 shows a scenario where the number of receivers is 14. Traffic load increases at each node but at different rates as time passes.

The trend is consistent with our analysis: At both scales, when W goes up and reaches the range [2, 2.1], the overall successful transmission probability is maximized, after which channel utilization drops quickly. (Hence, in the rest of our evaluations, each channelŠs lower bound W^{*} and upper bound W^{**} were chosen to be 2 and 2.3, respectively.)

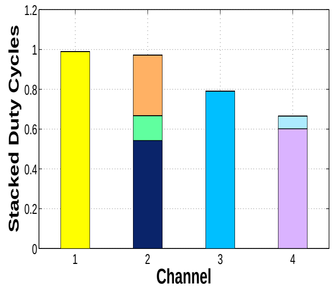
4.1.2 *Effectiveness of Channel Selection*

In all simulations henceforth we typically considered 4 channels and again considered 7 pairs of senders and receivers whose traffics increased at different rates. We tabulate the fractions of the duty cycle of each receiver assigned to each channel.

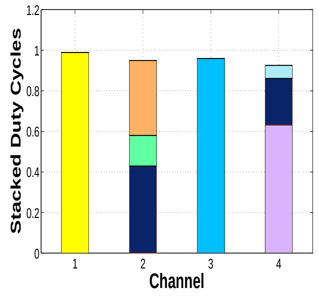
Figs. 6(a)-(c) validates that Chameleon's fraction computation using W^{*}–$\exp(I)$ to estimate q respects the goal of approaching but not exceeding the capacity of each channel. Each figure is derived from a snapshot taken at different times, and shows the load fraction of each receiver in a different color and stacks up the fractions per channel.

In Fig. 6(a), for instance, the aggregate load assignment on channel 2 approaches one; the node indicated by the dark blue bar in the channel has 54.3% load on the channel 2. In Fig. 6(b), as the load of the node increases, its load is split over channels 2 and 4. Fig. 6(c) presents the distribution of duty cycles as traffic loads continue to rise. When all channels were full, Chameleon stopped reassigning channels. (Note that we did not enforce a traffic policing component in the simulation, which is why the overall traffic load exceeded the threshold).

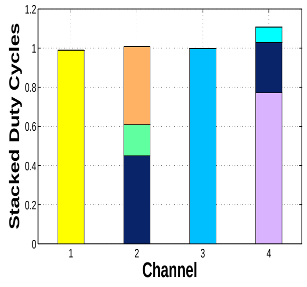
4.1.3 *Energy Efficiency Comparison*



(a) Time 1

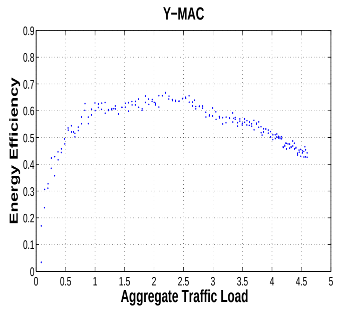


(b) Time 2

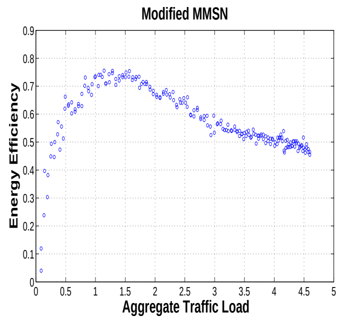


(c) Time 3

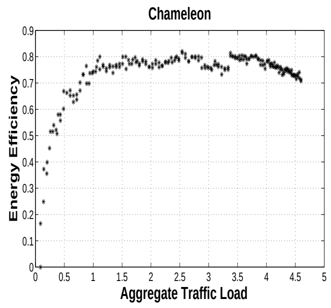
Figure 6: Duty Cycle Distribution in Three Periods



(a)



(b)



(c)

Figure 7: (a) Y-MAC (b) Modified MMSN (c) Chameleon

Towards comparing the energy efficiencies of the three multi-channel protocols, i.e., MMSN, Y-MAC, and Chameleon, in a fair manner we made several necessary modifications to the other two.

MMSN per se does not consider duty cycling. However, in our improved version, we used the same receive duty cycle scheduler on MMSN as in Chameleon, to reduce energy consumption of idle listening. Moreover, we let each receiver wake up at its own slot as in Chameleon. Therefore, the original expensive frequency toggle preamble provided in MMSN is avoided given that senders are aware of receiver schedule. In other words, we modified MMSN to be an adaptive duty-cycled protocol while reducing its protocol overhead.

In Y-MAC, both senders and receivers keep switching their channel from slot to slot. We let the channel stabilization window be rather short (2ms), less than the value originally specified in Y-MAC [10]. Backoff periods and data communication periods in every slot were specified to be 2ms and 10ms, respectively.

In the comparison, we did not let Chameleon enforce restrictions on incoming traffic even if all channel capacities had been exceeded. Such policing would, however, help the performance of Chameleon.

It follows that in our simulations the maximal efficiency of slot utilization, β, in the modified MMSN and Chameleon is around $10/(2+10) \approx 83.3\%$ whereas for Y-MAC, it is around $10/(2+2+10) \approx 71.4\%$. We calculate the energy efficiency to be β times the ratio of the total number of slots where packets were successfully received to the total number of active slots at receiver side, including the slots for channel activity monitoring. (The monitoring overhead is: zero for MMSN, since channel assignment is done a priori; one per frame if no packets arrive when the receiver wakes up for YMAC; and an entire frame if the measured W exceeds W^* for Chameleon.)

Fig. 7 compares the energy efficiency for different aggregate traffic loads. We see that as load increase, Y-MAC improves data reception ratio by deterministically switching channels and gradually distributing receive duty cycle into multiple channels. It performs gracefully when the number of extended slots does not surpass the number of available frequencies. Nevertheless, when traffic continues to grow, the probability of collisions increases accordingly, leading overall efficiency drops slowly to 40%.

Since nodes in are assigned to channels statically in MMSN, some channels are underutilized while others are overloaded due to imbalance of traffic in the network (see Plot (b)). Hence, the trend of system efficiency declines quickly than that of Y-MAC.

Plot (c) illustrates that Chameleon is typically 5% to 25% (and typically more energy efficient than the other two MACs). Notably, it maintains around 75%-80% efficiency even when capacity of all four channels is exceeded. The observed small fluctuations in the efficiency are due to the process of reassigning colors based on channel monitoring.

4.1.4 Coexistence with External Interference

Even or static assignment of load to channels as in YMAC and MMSN is inherently inefficient if the utilization of shared spectrum by external systems is not monitored. Since Chameleon monitors channels comprehensively, it is intrinsically adaptable to dynamic and un-

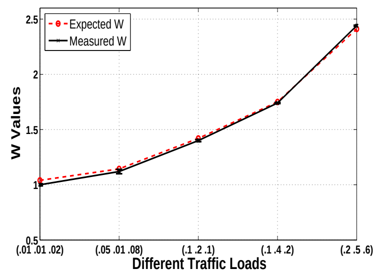


Figure 10: Comparison of Mean W under Different Traffic Loads

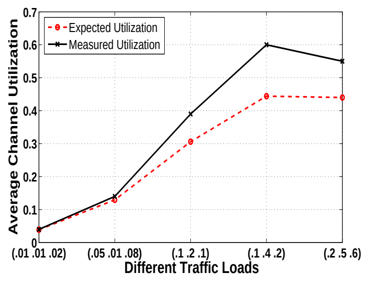


Figure 11: Comparison of Mean Channel Utilization under Different Traffic Loads

known wireless environments.

We studied the effects of external interference in two ways, letting the interferers use only 2 of the 4 available channels. In the first setting, we let the external traffic grow while holding the internal traffic steady. In the second setting, we explored the effect of growing the internal traffic, while the external traffic exhibited dynamic variation in terms of both channel selection and load.

Fig. 8 illustrates the first setting with 7 light internal loads. The energy efficiency of Y-MAC and modified MMSN deterministically degrades as the two channels became more crowed. Y-MAC is affected more strongly since its home was occupied by one of interferers. Chameleon's adaptation leads to substantially higher energy efficiency in this setting.

Fig. 9 illustrates the second setting. Fluctuations observed in these three plots indicate the effects of variation of external traffic. Chameleon is again least affected, but is also able to maintain its high energy efficiency when the channels start filling up.

4.2 Experimental Evaluation

To gain higher fidelity validation in the presence of environment and (TelosB) platform effects, we experimentally evaluated the accuracy of the measured W with respect to the analytically predicted W, as well as corroborated Chameleon's ability to tolerate external traffic and its relative improvement over MMSN and YMAC.

A first set of experiments involved three pairs of nodes executing Chameleon on channel 10, with frames consisting of 200 slots of size 15ms. Every receiver calculated its W periodically. Figure 10 plots the mean of measured Ws compared to expected value of W for five load combinations. The errors are rather small, within 0.05.

Figure 11 plots the average of the measured channel utilization compared to the expected value for the five load combinations. While the trend is similar as the loads increase, the measured value exceeds the expected value indicating that in measuring the in-traffic load of nodes in their interference set, nodes missed observing some packets, which is likely a result of slots being misaligned or the effect of local backoffs.

The second set of experiments involved inducing different changes in external traffic. Nodes used three channels (6, 12, and 18) on the TelosB mote with low internal traffic arriving at two receivers (Rx 1 and Rx 2 with 30% and 40% in-traffic load, respectively) from multiple senders apiece. External traffic on the three channels was raised on the middle channel from a low interference traffic load

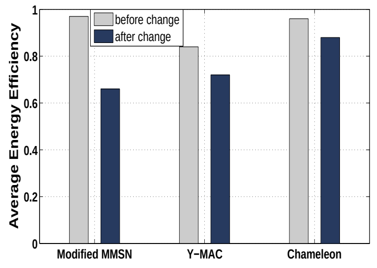


Figure 12: Comparison of Energy Efficiencies in Dynamic Channel Conditions

of 40%, 0%, and 20% on the three channels initially, respectively, to 70% (on channel 12).

Figure 12 shows the performance of each protocol before and after interference changes the channel condition. The energy efficiency of modified MMSN (which had high efficiency initially since all traffic to Rx1 was on channel 18 and to Rx2 was on channel 12, so there was essentially no contention) drops significantly due to increased interference on channel 12.

Y-MAC's even splitting of the traffic implied it was less impacted, but its slot-to-slot switching of channels in our experiments showed degraded coordination between senders and receivers, resulting in the average number of average number of received packets for YMAC being the lowest.

As for Chameleon, when the channel condition changed, it switched Rx 2's channel to channel 6, which yielded the highest energy efficiency among three cases.

5. RELATED WORK

The design of extant WSN MAC protocols focuses on scheduling as well as duty cycling to achieve high energy efficiency. There are a few works on adjusting duty cycle to the traffic load. Most of their mechanisms are implicit (e.g. [16],[6] and etc.): nodes remain active when they sense communication on the radio. In SCP [16] and T-MAC protocols [6], nodes increase their duty cycle if they overhear potential traffic towards them in the neighborhood. There are also schemes, such as D-MAC [13], that explicitly control duty cycle by assigning staggered transmission schedules to nodes in the collection tree to reduce delay of forwarding. Therefore, latency minimization in duty-cycled network is the focus of these works. MAC protocols like Y-MAC [10] and RI-MAC [14] dynamically

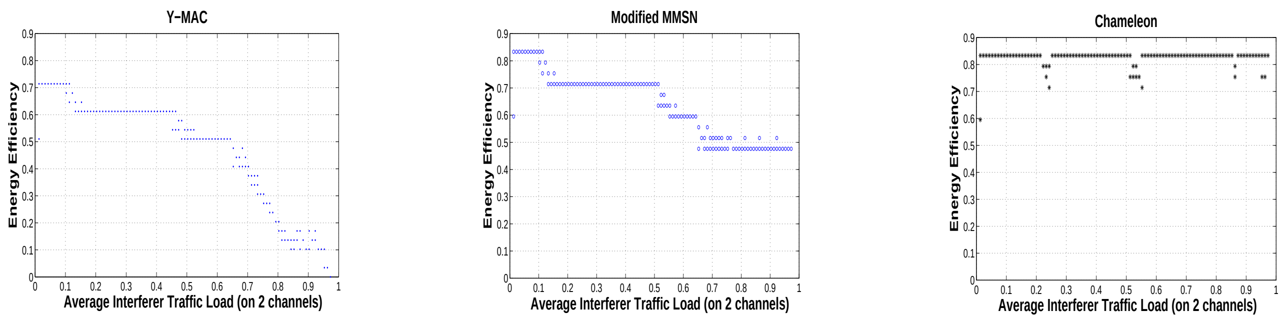


Figure 8: Scenario I: Internal Network Loads Remain While Interference Load Increases Gradually

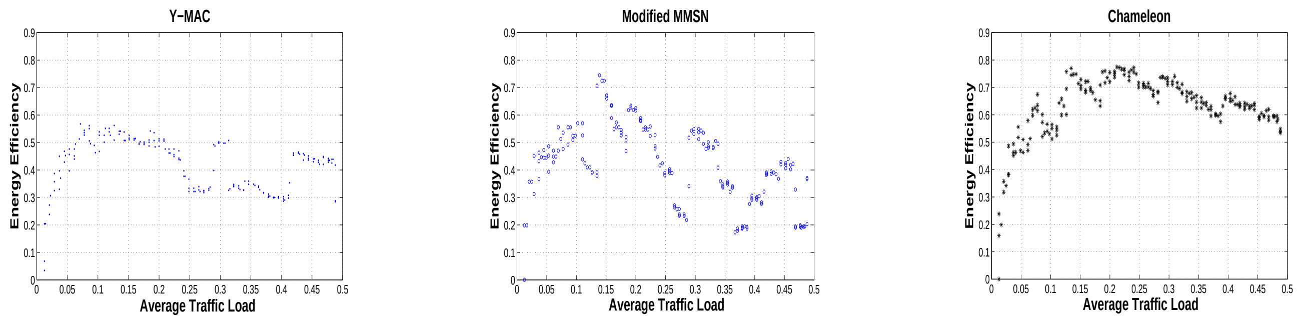


Figure 9: Scenario II: Internal Network Load Grows Gradually While Interference Changes Dynamically

extend receive duty cycle implicitly by monitoring channel activity for a short period of time. However, without an explicit knowledge of how long the receiver would be active, senders have to compete for every slot, which leads to high contention. By way of contrast, adaptive duty cycling in Chameleon emphasizes the maximizing energy efficiency, which explicitly minimizes awake duration of receivers based on incoming traffic.

In the state-of-the-art research, a significant number of multi-channel protocols have been proposed for sensor networks. Per our earlier classification, the first category statically assigns multiple frequencies to nodes in the network as a way of topology control in order to reduce potential interferences. Channel allocation is carried out beforehand, and is independent of real traffic conditions, such as in [19][15]. In [19], every node is assigned a color for data reception such that most of two-hop neighbors do not communicate on the same channel. The TMCP protocol [15] divides nodes into several subsets of different colors, wherein nodes only communicate within their subset for simplicity of implementation. These schemes require a centralized channel assignment algorithm to execute in the beginning and the channel utilization is not adjusted according to communication load or interference on each channel.

The second approach expands channels when contention on current channel become higher than an empirically chosen threshold. A distributed protocol in [12] lets all nodes in the network start in their home channel. When the channel becomes overloaded, a fraction of the nodes migrate to the next one. Channel switching is performed with a probability such that while alleviating congestion, it avoids having all nodes jump to the new channel. However, this protocol does not have a global view of quality on each channel, thus, channel switching need not result in higher efficiency. Another work [11] presents a centralized protocol for load balancing across channels for throughput maximization. Each node periodically decides which channel to use based on measurements from the base station. The authors assume that network throughput is optimized as long as loads are distributed equally on each channel, and they do not consider the channel goodput.

There are also schemes based on frequency hopping [3] which are designed mainly for wireless ad hoc networks involving continuously switching channel slot to slot even when there is no need for transmission. TMMAC [17] is a TDMA based multi-channel MAC protocol, which consists of a ATIM window where each node negotiates which time slot and channel it will use for transmission, and a communication window in which scheduled communications would be performed. A drawback of this for sensor networks is that it is heavily loaded, since every data transmission has to be scheduled in the ATIM window.

The most recent multi-channel protocol, Y-MAC [10], exploits both duty cycling and multi-channel utilization. Every receiver wakes up at its non-overlap slot within each frame on home channel. If more packets need to be received, the receiver will stay awake but hop to the next channel for reception. In this way, traffic is shifted to multiple channels concurrently and deterministically. The merit of this scheme lies in its staggered non-overlapping channel utilization over the extended M slots, while its weakness is that contiguous channel switching is expensive and the non-overlapping is guaranteed only within the M slots.

There exists other work on exploiting frequency diversity based on devices with multiple radio interfaces or with interfaces which can listen simultaneously on different channels [9]. To our knowledge, most hardware platforms for WSN, however, do not support these techniques.

6. CONCLUDING REMARKS

This paper has presented a novel multi-channel MAC protocol, Chameleon, for duty-cycled wireless sensor networks. Chameleon betters the energy efficiency of existing protocols by adapting both the duty cycle and the channels that are being used. On one hand, it attempts to maximize spectrum utilization, via a light-weight channel utilization metric W that lets it split loads across channels effectively. On the other hand, it uses a receiver-centric approach to minimize on-duty time at the receiver, while letting senders wakeup only when they need to send and know the receiver is awake.

Simulation and experimental results confirm that Chameleon enhances energy efficiency significantly compared to other multi-channel protocols under various internal traffic scenarios. Related experiments have shown show us that external interference in long-lived WSNs is nontrivial, and is also typically unpredictable. Chameleon naturally coexists with dynamic conditions in spectrum and improves energy efficiency to a large extent in a practical deployment.

Future work will examine the dynamics of Chameleon in large scale setting. We seek to address potential stabilization issues in maximizing utilization among different interference sets.

7. REFERENCES

[1] *CC2420 Datasheet*. http://www.ti.com.

[2] N. Ahmed, S. Kanhere, and S. Jha. Multi-channel interference measurements for wireless sensor networks (poster). In *IPSN*, 2008.

[3] P. Bahl and R. Chandra. Ssch: Slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *The Tenth Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 216–230, 2004.

[4] H. Cao, A. Arora, K. W. Parker, and T. H. Lai. Continuous asynchronous discovery with efficient synchronous communication for mobile networks. Technical Report OSU-CISRC-4/07–TR34, 2007, The Ohio State University, CSE.

[5] H. Cao, K. W. Parker, and A. Arora. O-mac: A receiver centric power management protocol. In *The 14th IEEE International Conference on Network Protocols (ICNP)*, 2006.

[6] T. V. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys*, pages 171–180, 2003.

[7] P. Dutta and D. Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *SenSys*, pages 71–84, 2008.

[8] G. Halkes and K. Langendoen. Crankshaft: An energy-efficient mac ptorotol for dense wireless sensor networks. In *The European Conference on Wireless Sensor Networks(EWSN)*, 2007.

[9] N. Jain, S. R. Das, and A. Nasipuri. A multichannel csma mac protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of IEEE (IC3N)*, 2001.

[10] Y. Kim, H. Shin, and H. Cha. Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In *The 7th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 53–63, 2008.

[11] H. K. Le, D. Henriksson, and T. Abdelzaher. A control theory approach to throughput optimization in multi-channel collection sensor networks. In *IPSN*, 2007.

[12] H. K. Le, D. Henriksson, and T. Abdelzaher. A practical multi-channel media access control protocol for wireless sensor networks. In *IPSN*, pages 70–81, 2008.

[13] G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy efficient and low-latency mac for data gathering in wireless sensor networks. In *Parallel and Distributed Processing Symposium*, 2004.

[14] Y. Sun, O. Gurewitz, and D. B. Johnson. Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *The 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 1–14, 2008.

[15] Y. Wu, J. A. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *Infocom*, pages 1193–1201, 2008.

[16] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle mac with scheduled channel polling. In *SenSys*, pages 321–334, 2006.

[17] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic. Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks. In *ICC*, pages 3554–3561, 2007.

[18] R. Zheng, J. C. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. In *The 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 35–45, 2003.

[19] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. Abdelzaher. Mmsn: Multi-frequency media access control for wireless sensor networks. In *The 25th IEEE International Conference on Computer Communication (Infocom)*, 2006.