Opportunistic MAC Protocol for Co-existence of M2M and Wi-Fi Network

Ajinkya Rajandekar and Biplab Sikdar

***Abstract*—Internet-of-Things is expected to increase the number of connected devices to billions. The major share of these devices will not be user operated, rather they will be machines communicating with other machines without any human intervention. In M2M communication, the network access to such high density of M2M devices is real bottleneck because of limited spectrum. The unlicensed spectrum is affordable choice for M2M networks but presence of existing technologies like Wi-Fi makes network access even more difficult problem to address. We present a MAC protocol design for coexistence of M2M and WiFi network in which high density M2M nodes can be served along with WiFi traffic. The MAC protocol uses opportunistic medium access technique which utilizes WiFi white spaces to serve M2M traffic using a hybrid contention-transmission MAC scheme. The hybrid MAC is optimized to utilize the varying lengths of white spaces effectively. We also present the effect of such opportunistic M2M communication on existing Wi-Fi network in terms of throughput and delay performance of Wi-Fi traffic. We show that proposed protocol can serve large number of M2M devices without significant impact on Wi-Fi networks.**

***Index Terms*—MAC protocols, WiFi, IoT, M2M communication, opportunistic communication.**

I. Introduction

Internet-of-Things has opened tremendous opportunities to improve the quality of life. The major differentiating factor between todays user driven networks and future Internet-of-Things will be massive number of machines operating autonomously without user intervention. The communication between autonomous machines refers to machine-to-machine (M2M) communication. The applications of M2M communication can be seen in many areas including smart homes, health care, smart grid and automation. In these and other such applications, Wireless M2M communication is especially useful and at the same time most challenging. The bottleneck for wireless M2M communication is frequency spectrum or bandwidth required for large number of M2M devices transmitting small amount of data at some interval of time. M2M communication solutions on licensed spectrum are costly. Another cost-effective option for wireless M2M communication is using the unlicensed industrial, scientific and medical band (ISM) where required coverage area is smaller. However, WiFi and many other networks are already using the ISM band and thus introducing M2M communication in such crowded spectrum increases interference, collisions and packet loss resulting in poor user experience.

Although Wi-Fi networks are deployed in most homes and commercial infrastructures, they are seldom fully utilized [1].Thus with proper coordination it is possible to use unlicensed ISM band for M2M communication. In this paper we propose a MAC protocol which coordinates M2M communication in ISM band in Wi-Fi white spaces. White spaces are time periods during which Wi-Fi network is inactive. The authors in [3] proposed a MAC protocol for ZigBee based sensors which inhibit Wi-Fi communication when sensors wishes to communicate with the ZigBee gateway. Wi-Fi communication is blocked by sending a clear-to-send (CTS) packet. To be able to send a CTS, the devices need to have Wi-Fi transceiver along with ZigBee, which increases the cost. Further improving this idea, the authors in [4] reserved the Wi-Fi channel by sending the CTS packet from the access point (AP). The AP regularly sends a CTS with duration field of 32 ms and coordinates M2M communication using time division multiple access (TDMA). Because the duration reserved by CTS is very long, the WiFi client nodes may not honor the reservation [5], especially when they observe channel not being used in between. The repeated CTS reservation without being followed by appropriate Wi-Fi transmission can be interpreted as a CTS denial-of-service (DoS) attack [6]. The TDMA or other convention MAC protocols for M2M communication are not scalable or adaptable to bursty M2M traffic as discussed in [7].

Therefore, in this paper we propose a novel M2M MAc prtoocol which utilizes the white spaces in WiFi traffic for M2M communication. The coordination between WiFi and M2M transmission is conducted by WiFi access point (AP). The WiFi AP determines when the WiFi network is having a white space and estimates the duration of the present white space based on WiFi traffic. This duration is then reserved by the AP for M2M communication by sending a modified CTS (mCTS) packet. The reserved duration of mCTS is kept within limit in proportion to the white space duration so that arriving WiFi packets are not delayed. The M2M communication is conducted using a novel hybrid MAC protocol consisting of two stages, namely, contention stage and transmission stage. The contention stage is divided in contention slots (L) of equal duration and M2M nodes contend with p-persistent ALOHA protocol. The parameters L and p are determined dynamically based on the duration of white space available thus maximizing the channel utilization. Simulation results show that the proposed protocol effectively utilizes the white spaces, allowing significant M2M data transfer with minimal effect on WiFi transmissions.

The rest of the paper is organized as follows. Section II describes the design of the proposed MAC protocol. Section III presents the estimation and optimization algorithms used to determine the parameters of the MAC protocol. In Section

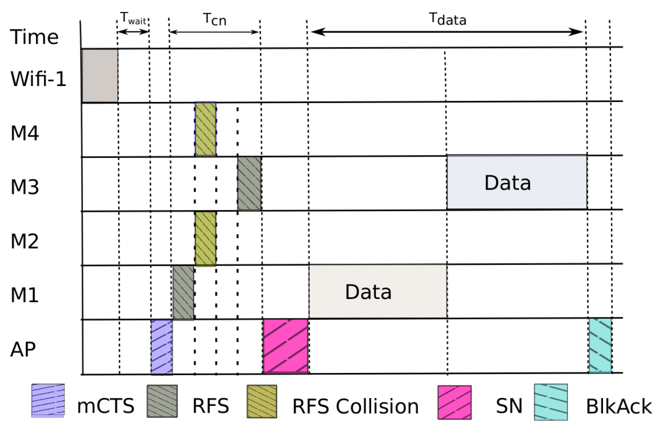

Fig. 1. Operation of the proposed MAC protocol.

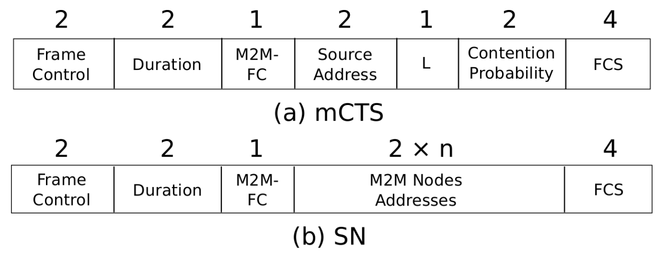

Fig. 2. Structure of (a) mCTS and (b) Slot Notification (SN) messages.

IV we develop a model to characterize the delay introduced in WiFi packets due to M2M communication. We present simulations results to evaluate the performance of the proposed MAC protocol and delay model in Section V. Section VI presents the conclusions.

II. Opportunistic MAC Protocol Design

The proposed protocol is based on following network scenario. We consider a WiFi network with one AP, n M2M nodes and m WiFi nodes. The arrival traffic at individual WiFi node is modeled using a batch Markovian arrival process (BMAP). BMAP is a versatile arrival process and can be used to model varieties of internet traffic including voice, data, video etc. The M2M nodes transmit data to WiFi AP using the proposed hybrid protocol explained below.

A. Protocol Operation

The hybrid MAC protocol is conducted in white spaces of WiFi network. The AP determines the beginning of the white space and notifies all the M2M nodes. The M2M nodes then contend for the channel in contention stage using a frame slotted p-persistent ALOHA protocol. In the transmission stage the successful M2M nodes transmit their packets in TDMA fashion. After the end of transmission stage the AP releases the reserved channel and WiFi nodes resume their operations. Figure 1 shows the operation of the protocol. Following sections describe each stage of this protocol in detail.

B. Alert Stage

The AP evaluates the start of white space at the end of a busy period when its MAC layer queue becomes empty. The AP has to check if some WiFi nodes have uplink traffic to serve before it can determine the beginning of white space. Thus, the AP begins a waiting period given by T_{wait} for uplink WiFi traffic. If no uplink packet is received in this T_{wait} period the AP concludes that no WiFi device has data to send and thus it marks the start of a WiFi white space.

At the end of a waiting period, the AP estimates the length of white space based on the WiFi traffic behavior it has observed. Also the AP estimates the number of active M2M nodes based on the results of previous contention stage. Based on the duration of white space and number of active M2M nodes the AP evaluates the length of contention stage and probability of contention. The AP sends an mCTS packet which reserves the duration of white space estimated. The mCTS packet also contains the information of number of contention slots and probability of contention for M2M nodes. The frame structure of a mCTS packet is shown in figure 2(a). The mCTS packet structure is identical to a IEEE 802.11 CTS packet and thus WiFI nodes decode the mCTS packet as a CTS packet. The mCTS packet updates the network allocation vector of WiFi nodes and they defer attempting transmission in duration reserved by mCTS. The source address field of mCTS packet is modified to hold M2M frame control sequence, number of contention slots and probability of contention. The M2M nodes use these parameters to contend for the channel.

C. Contention Stage

After receiving the mCTS packet, the M2M nodes extract the information about number of contention slots and probability of contention as it marks the beginning of a M2M communication cycle. In frame slotted p persistent ALOHA, M2M nodes select a contention slot at random among L contention slots available and transmit a request for slot (RFS) packet in that slot with probability p. To avoid errors due to lack of clock synchronization, guard times are inserted between successive contention slots.

The AP receives RFS packets from M2M nodes in each slot of a contention stage. If only one M2M node transmits RFS packet in a particular contention slot the AP denotes the slot as successful slot and records the address of that M2M node. If more than one M2M nodes choose the same slot to transmit, the AP can not receive a valid RFS packet and thus records the slot as collision. If no M2M node attempts transmission in a contention slot, the AP assumes the slot as idle slot. In the following M2M communication cycle this information of idle, successful and collision slots is used to calculate the estimated number of active M2M devices which have data to transmit. The M2M nodes whose RFS packet was successfully received by the AP are added to a list of nodes which are assigned data slots in transmission stage.

D. Notification Stage

The result of contention stage is notified to all the M2M nodes by AP by sending a slot notification (SN) message. The

SN packet assigns data slot of transmission stage to successful M2M nodes. The data slot assignment is based on the order of node address given in the SN packet as shown in Figure 2(b). For example, if a M2M node A is assigned 1st slot, its address is placed at the beginning of the list in SN packet followed by the address of the node who is assigned second slot and so on. Each node is assigned only one slot in transmission stage.

In a case when no M2M node transmit a RFS packet in contention stage, the AP understands it as absence of M2M traffic. Thus it immediately cancels the channel reservation by sending a SN-ACK packet which has the same frame structure as an IEEE 802.11 ACK packet. When WiFi nodes receive a SN-ACK packet they reset their NAV and start contending for wireless channel. In scenario where no RFS transmission was successful due to multiple collisions, the AP reduces the probability p and restarts the contention stage.

E. Data Transmission & Acknowledgement Stage

When the M2M nodes receive the SN packet which contains their address, the M2M nodes determine the slot in which they are supposed to transmit data based on their place in SN packet. M2M nodes wait for the assigned slot and transmit their data in that slot. M2M nodes achieve time synchronization using the SN packet reception time. To offset errors in time synchronization due to propagation delays, guard bands are inserted in between successive data slots. Data slots are kept of fixed length for simplicity and to avoid overhead.

When the AP receives all the data packets from M2M nodes, it acknowledges the reception by sending a block acknowledgement packet. The block acknowledgement packet ends the M2M communication cycle and resets the NAV of WiFi nodes. The WiFi nodes resume their transmission after the end of M2M communication cycle. This paper mainly considers the uplink data traffic of M2M nodes but it should be noted that AP can send downlink traffic by inserting addition data slots at the end. The acknowledgement for downlink packet is sent by M2M nodes immediately after receiving data packet in the same data slot.

III. O-MAC Parameter Estimation

This section presents the methodology and associated mathematical models for estimating the parameters of the proposed MAC protocol.

A. Estimation of White Space Duration

At the start of each white space, the AP needs to estimate its expected duration.

The average duration of a white space, T_w, in a $BMAP/G/1$ queue is given by [9]

$$T_w = \pi((-D_0)^{-1})^2(D(1) - D_0)e. \tag{1}$$

where D_0 is matrix containing the transition rates for which no arrivals occur, and the matrices D_i contain the transition rates for which a batch size of i occurs. $D(i)$ denotes the matrix generating function of the BMAP arrival process and it is given by

$$D(z) = \sum_{k=0}^{\infty} D_k z^k, \qquad \text{for } |z| \leq 1. \tag{2}$$

As shown in [9], the distribution of the duration of white spaces does not depend on the service time distribution. Thus one can calculate the average white space duration simply by knowing the arrival traffic characteristics. For example, for a BMAP with $m = 2$ and batch arrivals of size $k = 2$, we have

$$T_w = \pi((-D_0)^{-1})^2(D_1 + D_2)e. \tag{3}$$

The average number of white spaces in a $BMAP/G/1$ queue is given by [9]

$$N_w = \frac{p_0}{\pi((-D_0)^{-1})^2(D(1) - D_0)e} \tag{4}$$

where p_0 is the fraction of time the queue is idle. As the busy periods and idle periods strictly alternate in the queue, the average number of busy periods in one unit of time is equal to average number of idle periods. Therefore, the average number of busy periods, N_b, in one unit of time is given by

$$N_b = N_w = \frac{p_0}{\pi((-D_0)^{-1})^2(D(1) - D_0)e}. \tag{5}$$

Since there are N_b busy periods on average in an unit of time, and the fraction of time the queue is busy is $\rho = 1 - p_0$, N_b is also given by ρ/T_b where T_b is the average duration of a busy period. Equating these two expressions for N_b, we get,

$$T_w = \pi((-D_0)^{-1})^2(D(1) - D_0)e = \frac{1}{N_b} - T_b. \tag{6}$$

Thus the AP can efficiently compute the average duration of a white space, without having to calculate the matrices D_k of the BMAP, by simply observing the duration of busy periods. Note that from the AP's perspective, observing a busy period is much easier and practical than observing a white space. This is because the AP cannot accurately determine the end of a white space of the aggregate WiFi system as some WiFi nodes may have packet arrivals while M2M communication is going on. On the other hand, once a busy period begins, the AP can observe both uplink and downlink traffic and thus determine the length of the busy period more accurately.

WiFi traffic can vary over time and thus the characteristics of the white spaces also change with time. Thus the proposed protocol uses an exponentially weighted moving average to track T_b. At the end of the n-th busy period, the estimate for the average duration of a busy period is given by

$$\hat{T}_b \triangleq \hat{T}_{bn} = T_{bn}\alpha + (1 - \alpha)\hat{T}_{bn-1} \tag{7}$$

where T_{bn} is the last observed duration of a busy period. Based on our empirical results, we use α as 0.2 so that recent busy period characteristics are given more weight. To use $\hat{T}_b$ for estimating the average duration of a white space using (6), the AP also records the number of busy periods every second. Thus by observing only two quantities (T_b and N_b), the AP can estimate the average duration of white spaces. This approach is computationally efficient and allows for an updated estimated at the end of every busy period. In Section V, we use simulations to show that the estimates $\hat{T_w}$ obtained using the methodology above are accurate compared to the analytic results given by (1).

B. Estimation of Active M2M Nodes

For optimal use of white spaces, the available time needs to be divided between contention and data slots based on the M2M nodes with data to send. To achieve this, the AP estimates the number of active M2M nodes using the contention result of the previous M2M communication cycle. Let p be the probability of transmission and L be the number of contention slots in the previous contention cycle. Let $\hat{I}$ be number of idle slots observed in the previous contention period. An estimate of the number of active M2M nodes is given by

$$\hat{n} = \frac{L}{p} \ln \left(\frac{L}{\hat{I}} \right). \tag{8}$$

The expression above cannot be used for cases where none of slots were idle (i.e. $\hat{I} = 0$). For Poisson arrivals at the M2M nodes, the estimate for the number of active nodes with zero idle slots is given by [12],

$$\hat{n} = \frac{S + 2.39C}{p}. \tag{9}$$

where S and C denote the number of successful and collision slots respectively and the factor 2.39 represents the average number of M2M nodes involved in a collision. Equations (8), and (9) are closed form expressions to estimate number of active M2M nodes that can be evaluated by the AP in real time without any computational complexity.

C. Determination of Contention Parameters

Using the estimates of the white space duration available ($\hat{T_w}$) and number of active M2M devices ($\hat{n}$), the AP has to select the protocol's parameters to maximize the channel utilization. The parameters to be decided are the number of data slots, n_d, number of contention slots, L, and the contention probability p. The parameters assumed to be known (and fixed) are the length of a contention slot, T_c, and the length of a data slot, T_d. Data slot lengths are assumed to be equal for simplicity. Let T_{CL} denote the entire duration of a M2M communication cycle, and it is given by

$$T_{CL} = LT_c + n_d T_d + T_{SN} + T_{BACK} \tag{10}$$

where T_{SN} and T_{BACK} are the time taken by a slot notification message and block acknowledgement message, respectively.

The channel utilization, η, during a M2M communication cycle is given by

$$\eta = \frac{n_d T_d}{T_{CL}}. \tag{11}$$

The optimization problem that maximizes this utilization can be expressed as

$$\begin{aligned} \max_{L, n_d, p} \quad & \eta \\ \text{subject to} \quad & T_{CL} \leq \max\{\hat{T}_w, T_{min}\}, \\ & n_d \leq L, \\ & 0 \leq p \leq 1. \end{aligned}$$

where T_{min} is the time required to transmit one M2M packet and one contention slot. The first constraint reflects that, in general, the M2M communication cycle should be smaller than the white space duration. However, if the white space duration is too small, this constraint ensures that the M2M communication cycle is long enough to accommodate at least one transmission. The second constraint reflects the fact that data slots (i.e., n_d) are needed only for successfuly nodes, and this number is bounded by the number of contention slots, L. As the contention stage uses p-persistent frame slotted ALOHA, the probability of success in a contention slot, when $\hat{n}$ nodes contend in L slots with probability p, is given by

$$P[\text{Success}] = \frac{p\hat{n}}{L} \left(1 - \frac{p}{L}\right)^{\hat{n}-1} \tag{12}$$

and the expected number of successful slots, $E[S]$, is given by

$$E[S] = p\hat{n} \left(1 - \frac{p}{L}\right)^{\hat{n}-1}. \tag{13}$$

To select the parameter p such that the number of successful slots is maximized for a given L, we differentiate the expression above with respect to p and equate to zero. After some algebra, the optimal contention probability is given by

$$p^* = \frac{L}{\hat{n}}. \tag{14}$$

Substituting this value of p^* in (13), we get

$$E[S] = \left(1 - \frac{1}{\hat{n}}\right)^{\hat{n}-1} L \approx \frac{L}{e}. \tag{15}$$

Since only successful contention slots result in data slots, we use $E[S]$ from (15) as the number of data data slots in the M2M communication cycle. Using these values of n_d and p^*, the optimization problem can be expressed as

$$\begin{aligned} \min_{L} \quad & 1 + \frac{T_c}{e^{-1}T_d} + \frac{T_{SN} + T_{BACK}}{e^{-1}LT_d} \\ \text{subject to} \quad & L \leq \frac{\max\{\hat{T_w}, T_{min}\} - T_{SN} - T_{BACK}}{e^{-1}T_d + T_c}. \end{aligned}$$

The objective function is monotonic in L, and thus the optimal solution is given by

$$L^* = \frac{\max\{\hat{T_w}, T_{min}\} - T_{SN} - T_{BACK}}{e^{-1}T_d + T_c}. \tag{16}$$

To select the parameters for the proposed protocol, the AP first uses (16) to determines the parameter L using the estimated white space duration and number of active M2M devices. Next, to select n_d, we first note that if the number of

M2M devices with data to transmit is smaller than the number of data slots that can be accommodated in the current white space, then the M2M communication cycle does not have to occupy the entire white space. Thus, the AP selects n_d as

$$n_d = \begin{cases} \delta L, & \text{if } e^{-1}L < \hat{n} \\ \hat{n} & \text{otherwise} \end{cases}. \tag{17}$$

Finally, the contention probability is selected as

$$p = \begin{cases} \frac{L}{\hat{n}}, & \text{if } L < \hat{n} \\ 1 & \text{otherwise} \end{cases}. \tag{18}$$

D. Frequency of M2M Communication Cycles

In the absence of M2M traffic, sending mCTS packets at the end of every busy period is an unnecessary excercise which can degrade the performance of the WiFi network. Thus, mCTS packets should be sent only when required. In the proposed protocol, the AP limits the transmission of mCTS packets based on the observed traffic in M2M communication cycles using the methodology described below.

Let R denote the average number of M2M communication cycles that had at least one active contention slot (i.e. at least one active M2M node) in a time interval. R depends on the number of mCTS packets sent during the time interval and the rate at which packets are generated by M2M devices. If M2M traffic is low then the number of M2M communication cycles with activity will also be low, even if many mCTS packets were sent during this period. Now consider the scenario where a mCTS packet is sent but no M2M node attempted a RFS transmission. The AP takes this as an indication that M2M traffic is low and thus it should wait for some time (denoted by T_{mCTS}) before sending the next mCTS packet, even if the WiFi network is idle. To determine the value of T_{mCTS}, the AP records the number of M2M communication cycles with active M2M nodes in each second. The average duration between active M2M communication cycles is then used as the mCTS waiting time. Thus we have

$$T_{mCTS} = \frac{1}{R}. \tag{19}$$

Note that this waiting time is invoked only when the AP observes that there was no activity from M2M devices in response to a mCTS packet. In such cases, the AP waits for T_{mCTS} before sending the next mCTS packet. In cases where M2M devices respond to a mCTS packet, the AP sends another mCTS packet at the next white space and does not have to wait for T_{mCTS}. This ensures that M2M devices are not starved of opportunities if they have data to send, while avoiding unnecessary mCTS packet transmissions when they are idle.

IV. WiFi Delay Model

M2M communication in WiFi white spaces does introduce delay in WiFi packets. This is because when the M2M communication cycle is going on the WiFi packets which arrive in that period will have to wait until the M2M communication cycle is going on. This introduces additional delay in WiFi packets. There are two factors which influence this additional delay in WiFi packets 1) T_W, duration of white space AP reserves for M2M communication and 2) Arrival traffic at M2M nodes. The duration of white space reserved increases the delay in WiFi packets as longer reservation implies longer waiting time for WiFi packets. Also the packets which arrive after completion of M2M communication cycle may experience additional delay due to increased waiting time in queue. As for the M2M arrival traffic, higher M2M arrival traffic will utilize the white spaces aggressively and all the M2M communication cycles will be at their maximum lengths thus adding the WiFi packets delay. We present a mathematical model in this section to characterize the increase in delay experienced by WiFi traffic.

Let us assume that M2M traffic is saturated and all M2M devices are always ready to send data. This scenario will give highest delay in WiFi packets due to M2M communication. The WiFi network is modeled using a $BMAP/G/1$ queue with server going for vacations at the end of busy periods. In the queue, the aggregated arrivals at all WiFi nodes are modeled as a BMAP. For the service distribution we use the IEEE 802.11 MAC protocol. The wireless channel is assumed to be the server serving wireless transmission from all the WiFi traffic. The reservation period by AP for M2M communication at the end of busy period is modeled as server going for a vacation. Any WiFi packet arrival during these periods has to wait until the server comes back from vacation. Once the M2M communication cycle ends and the vacation is over, the server transmits all waiting packets as well as those that arrive during this busy period. The server goes on the next vacation when there are no WiFi packets left to be served and the busy period ends. For analytical tractability, the model does not account for the waiting time of T_{wait} in the proposed protocol before the M2M communication cycle begins and the server goes on vacation.

Let W_v be the random variable denoting the waiting time of WiFi packets in the $BMAP/G/1$ queue with vacations. At the end of a busy period, the server goes on vacation for period V. We denote the Laplace-Stieltjes transform (LST) of the probability density function of V by $P^*(s)$ and its first and second moment by $E[V]$ and $E[V^2]$, respectively. Also, let W_{nv} be the random variable denoting the waiting time of WiFi packets in a $BMAP/G/1$ queue without vacations (i.e. a traditional WiFi network without the proposed M2M communication protocol). From [13], we know that the expected waiting time in the queue with vacations is the sum of the expected waiting time in a queue without vacations and the expected residual vacation period. Thus,

$$W_v = W_{nv} + \frac{E[V^2]}{2E[V]}. \tag{20}$$

Since the duration of the M2M communication cycle is chosen as T_w in the mCTS packet, we have

$$E[V] = T_w = \pi((-D_0)^{-1})^2(D(1) - D_0)e. \tag{21}$$

To evaluate $E[V^2]$, we first define $u^*(t,j|i)$ as the probability that the duration of an idle period (T) of the $BMAP/G/1$ queue is less than t and the phase of arrival process at the start of subsequent busy period is j, given that phase of the arrival process at the end of preceding busy period was i:

$$u^*(t,j|i) = P(T < t, j|i) \qquad \forall\ i,j \in 1,2,\cdots,m. \tag{22}$$

Let $U^*(t)$ denote a $m \times m$ matrix with elements $u^*(t,j|i)$, $1 \le i,j \le m$. The transform of $U^*(t)$ is given by [14]:

$$\mathrm{U}^*(s) = [sI - D(0)]^{-1}(D(1) - D(0)) \tag{23}$$

where I is a $m \times m$ identity matrix. The second moment of $U^*(t)$ can be obtained from the matrix

$$E[U^*(t)^2] = (-1)^2 \left.\frac{d^2(U^*(s))}{ds^2}\right|_{s=0} = 2\left(-D_0^{-1}\right)^3 (D(1) - D_0)\,.$$

$E[U^*(t)^2]$ represents a matrix of conditional expectations. Unconditioning, we have

$$E[V^2] = 2\pi((-D_0)^{-1})^3(D(1) - D_0)e. \tag{24}$$

Substituting these values of $E[V]$ and $E[V^2]$ in (20), the increase in the WiFi packet delays due to the proposed protocol is thus

$$W_v - W_{nv} = \frac{\pi((-D_0)^{-1})^3(D(1) - D_0)e}{T_w}. \tag{25}$$

The increase in the delay for the saturated case as given by (25) serves as an upper bound on the delay introduced to WiFi nodes due to the proposed opportunistic MAC protocol.

For unsaturated traffic conditions at M2M nodes, the delay introduced is lower because the server does not go on a vacation at the end of every busy period. Thus we introduce a factor ν, $0 \le \nu \le 1$, which denotes the fraction of white spaces that result in M2M communication cycles with traffic. ν is then given by

$$\nu = \frac{R}{N_w}. \tag{26}$$

Let the aggregate rate (i.e. sum of the individual rates) of packet arrivals at the M2M nodes be denoted by Λ_M. Since the traffic conditions at the M2M nodes are unsaturated, we have $\Lambda_M T_{min} < 1$. Also, the number of M2M packets transmitted in communication cycle, N_m, is upper bounded by T_w/T_{min}. Thus the number of active M2M communication cycles per second satisfies $R > \Lambda_M/(T_w/T_{min})$. Substituting this in (26), we get

$$\nu > \frac{\Lambda_M T_{min}}{N_w T_w} = \frac{\Lambda_M T_{min}}{p_0}. \tag{27}$$

Substituting this expression for ν in (20), the increase in the average waiting time for WiFi packets can be expressed as

$$W_v - W_{nv} > \frac{\Lambda_M T_{min}}{2p_0}\frac{\pi((-D_0)^{-1})^3(D(1) - D_0)e}{T_w}. \tag{28}$$

The average duration of white space, T_w, required for evaluating (25) and (28), is given by (1). Also, p_0 for a WiFi network modeled using a $BMAP/G/1$ queue can be calculated using the method described in [9].

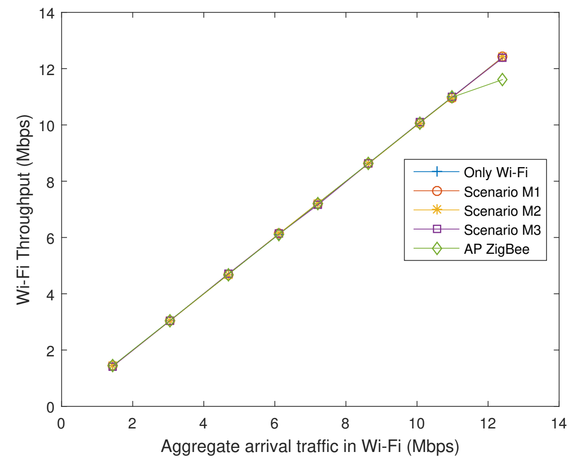

Fig. 3. Effect of M2M traffic on the throughput of WiFi network.

V. Simulation Results

This section presents simulation results to evaluate the performance of the proposed MAC protocol. The simulations were performed using the NS-3 simulator. The simulation setup consisted of one AP, m WiFi nodes, and n M2M nodes. The number of WiFi nodes was fixed at $m = 5$ and the number of M2M nodes was varied to create three different scenarios. Scenario M1 had $n = 100$ M2M nodes and M2M devices generated 3.45 packets per second per node. The size of each M2M packet was 85 bytes. Scenario M2 had $n = 100$ and M2M nodes had saturated conditions by generating 10 packets per second per node. In scenario M3, n was kept at 4 and M2M devices generated 55 packets per second per node. The WiFi nodes IEEE 802.11g as the protocol and a data rate of 18 Mbps while the M2M nodes transmitted at 1Mbps. The value of T_{wait} was set to $2CW_{min}$ where CW_{min} is the minimum contention window for the WiFi nodes. The number of contention slots, probability of contention, and number of data slots was chosen dynamically by the AP. As a benchmark, we compare our results with the protocol described in [4] and its results are denoted by the curves labeled "AP ZigBee" in the figures.

The arrival traffic at WiFi nodes was generated using a BMAP. The aggregate WiFi traffic arrival rate was varied from 1.4 to 12.4 Mbps (leading to a variation in the MAC layer utilization factor from $\rho = 0.1$ to $\rho = 0.9$ when there is no M2M traffic). To keep the comparison of results meaningful, the results are plotted as a function of the WiFi traffic in the network.

Figure 3 shows the WiFi throughput in Mbps for the various scenarios. It can be seen that there is no change in the throughput of the WiFi network even with saturated M2M traffic for scenario M2. This is because the protocol utilizes the white spaces in WiFi traffic for M2M packet transmissions. The proposed protocol always checks for the presence of WiFi traffic before commencing a M2M communication cycle, thereby ensuring that WiFi devices remain the primary users of

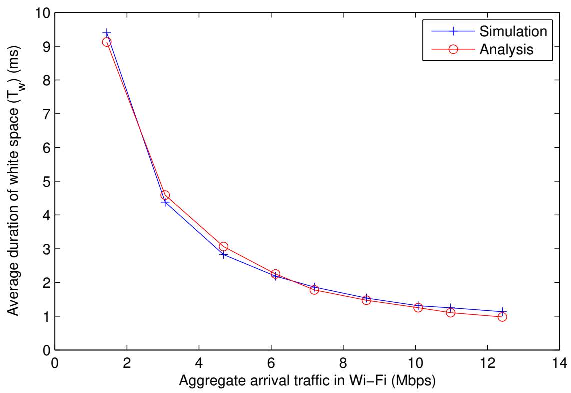

Fig. 4. Average duration of estimated white space (T_w) based on Wi-Fi traffic.

the channel and there is no difference in their throughput. On other hand, the protocol in [4] reserves the channel for M2M communications irrespective of WiFi network conditions. Thus at higher WiFi traffic rates, packets drops and network congestion are experienced.

To validate the methodology for estimating white space durations (T_w) at the AP, Figure 4 shows the white space duration estimated during the simulations with those obtained analytically. In the simulations, the estimation method described in section III-A was used to estimate T_w. The analytical values of T_w were obtained using (1). It can be seen that the estimated values are very close to the analytical values of T_w. As expected, the average white space duration decreases with increase in WiFi traffic, giving smaller opportunities for M2M communication.

The delay experienced by WiFi packets in the presence of M2M communication is shown in Figure 5. The increase in the delay of WiFi packets is of the order of $1-4$ milliseconds. When the WiFi traffic is small, the average duration of white spaces is large. Thus, if many M2M nodes are active, a large duration is reserved for M2M communication. Any WiFi packet that arrives during such periods will experience additional delays. On the other hand, when WiFi traffic increases, the average duration of white spaces decreases. Consequently, the reserved white space is also small, which in turn limits the increase in delay of WiFi packets. The WiFi delay when ρ is small may be reduced by allowing only small durations for M2M communication cycles. However, this approach would limit the opportunities available for M2M communication. The proposed protocol outperforms the protocol in [4] in terms of the delay experienced by WiFi packets. For example, for WiFi arrival rate of 12.4 Mbps (corresponding to $\rho = 0.9$ in the scenario without M2M traffic) the WiFi delay in the absence of M2M traffic is 5.1 ms while in scenario M3, the average delay of WiFi packets is 6.5 ms. On other hand at $\rho = 0.9$ the delay in the protocol from [4] goes to 400ms.

Figure 6 shows the utilization of M2M communication opportunities by plotting the number of mCTS packet transmissions and the number of M2M communication cycles the actually resulted in M2M transmissions. Using the estimation model in Section III-D, the AP can limit the sending of unnecessary mCTS packets in the absence of M2M traffic. The curve for mCTS packets sent labeled "M3W" denotes the number of mCTS packets sent for scenario M3 if the AP does not use the transmission limiting methodology presented in Section III-D. In scenario M3, M2M traffic is low and thus the number of mCTS communication cycles required is smaller than the number of available white spaces. Therefore, using a real-time estimation for M2M cycle requirement avoids unnecessary overhead by sending mCTS packets only when M2M traffic is likely to be present.

Figure 7 shows the number of M2M packets successfully transmitted per second. We see that the number of packets transmitted is constant when the network is unsaturated (e.g. scenarios M1 and M3). Under saturated traffic (scenario M2) the number of M2M transmissions decreases with an increase in the WiFi traffic since the available white space duration decreases. Increase in the WiFi traffic results in long busy periods in the WiFi network, thereby decreasing both the average duration and number of white spaces. As a result, the number of M2M transmissions also decreases. However, even with heavy WiFi traffic (12.4 Mbps, with $\rho = 0.9$ in the purely WiFi scenario) the number of M2M transmissions is around 100 per second in all scenarios.

Figure 8 shows the increase in WiFi packet delay due to the addition of M2M traffic. The figure also compares the model proposed in Section IV to estimate the increase in WiFi packet delay against simulation results. In all scenarios, the delay obtained using the bound provided by the analytic model is higher than delay observed in simulation. This may be partially attributed to the fact that in the proposed queuing model the vacation period starts immediately after a busy period ends. However, in the proposed protocol, the AP waits for time T_{wait} before sending a mCTS packet that prevents WiFi packets from accessing the channel. If there is any arrival within this period, the M2M communication cycle is not initiated. The proposed model serves well as an upper bound and is failry accurate at high loads.

VI. Conclusions

This paper proposed an opportunistic MAC protocol for M2M communication in the unlicensed band which effectively exploits the white spaces in WiFi networks. The proposed protocol efficiently utilizes the unused channel resources in the network while treating WiFi devices as the primary users. A real time optimization methodology is proposed to estimate the parameters of the proposed MAC protocol such that network utilization is maximized. A model to estimate the impact of introducing M2M communications on the delay experienced by WiFi packets is also proposed. The proposed protocol is a cost effective solution as existing infrastructure can be readily used for M2M communication without affecting WiFi traffic, with only minor changes to the MAC layer of the AP.

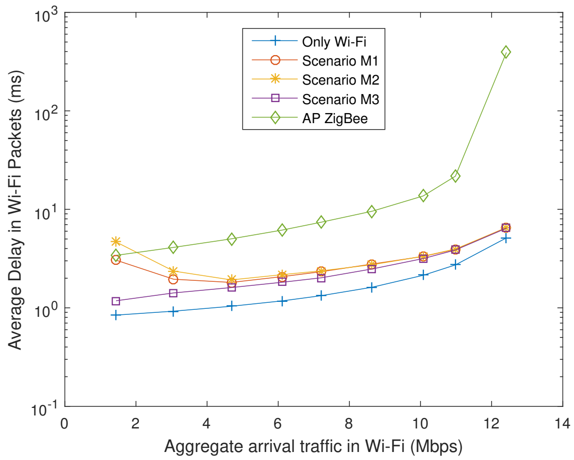

Fig. 5. Effect of M2M traffic on delay of WiFi packets.

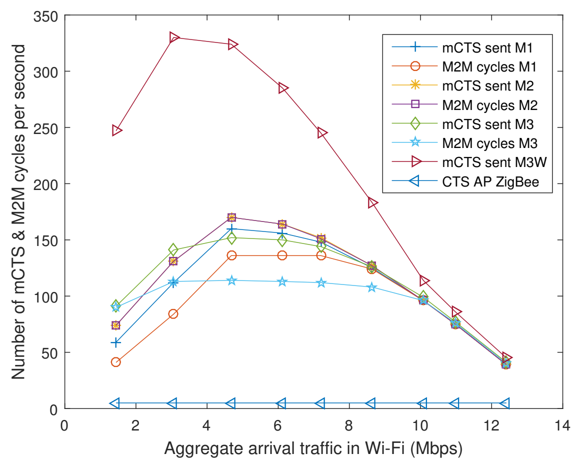

Fig. 6. Effective white space utilization: number of mCTS sent and successful M2M communication cycles.

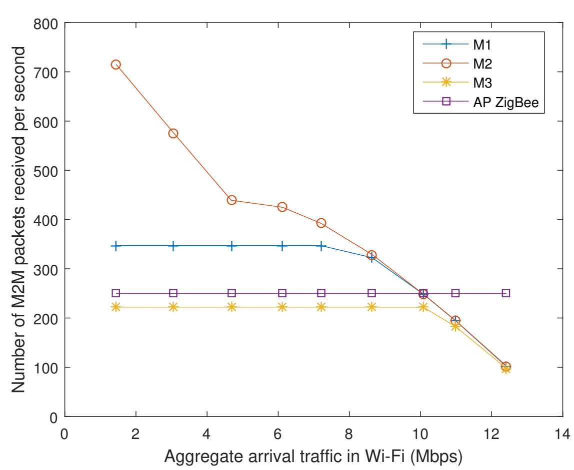

Fig. 7. Number of M2M packets received per second.

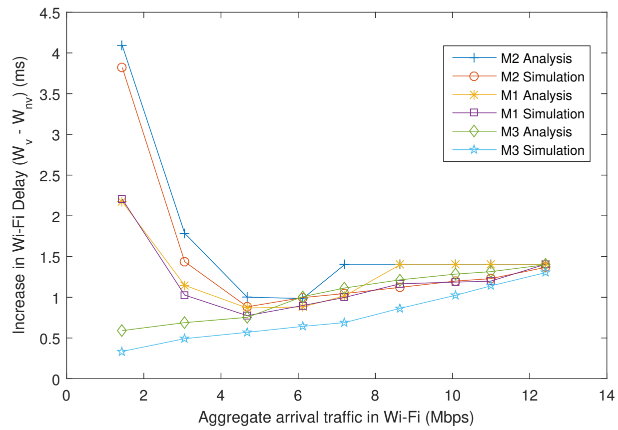

Fig. 8. Average increase in WiFi delay due to M2M communication.

References

[1] R. Raghavendra, J. Padhye, R. Mahajan and E. Belding, "Wi-Fi networks are underutilized," *MSR Technical Report*, 2009.

[2] A. Rajandekar and B. Sikdar, "On the Feasibility of Using WiFi White Spaces for Opportunistic M2M Communications," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 681-684, December 2015.

[3] J. Hou, B. Chang, D. Cho and M. Gerla, "Minimizing 802.11 Interference on Zigbee Medical Sensors," *Proc. ICST BodyNets*, pp. 18, Los Angeles, CA, April 2009.

[4] S. Ishida, S. Tagashira and A. Fukuda, "AP-Assisted CTS blocking for WiFi-ZigBee Coexistance," *Proc. IEEE ISCN*, pp. 110-114, Sapporo, Japan, December 2015.

[5] K. Gopinath, P. Bhagwat and K. Gopinath, "An empirical analysis of heterogeneity in IEEE 802.11 MAC protocol implementations and its implications," *Proc. ACM WiNTECH,* pp. 80-87, Los Angeles, CA, October 2006.

[6] D. Chen, J. Deng and P. Varshney,"Protecting wireless networks against a denial of service attack based on virtual jamming." *Proc. ACM MobiCom*, pp. 14-19, San Diego, CA, September 2003.

[7] A. Rajandekar and B. Sikdar,"A survey of MAC layer issues and protocols for machine-to-machine communications." *IEEE Internet of Things Journal*, vol. 2(2), pp. 175-186, 2015.

[8] D.M. Lucantoni,"The BMAP/G/1 queue: a tutorial." *In Performance Evaluation of Computer and Communication Systems, Springer*, pp. 330-358, Berlin Heidelberg, 1993.

[9] A. Rajandekar, B. Sikdar, "Characterization of White Spaces in Wi-Fi Networks for Opportunistic M2M Communications," *IEEE Transactions on Communications* , vol.PP, no.99, pp.1-1.

[10] A. Klemm, C. Lindemann and M. Lohmann, "Modeling IP traffic using the batch Markovian arrival process," *Performance Evaluation*, vol. 54, no. 2, pp. 149-173, 2003.

[11] J. R. Cha and J. H. Kim, "Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system", *Proc. IEEE CCNC,* pp. 768-772, January 2006.

[12] F. Schoute, "Dynamic Frame Length ALOHA," *in IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565-568, Apr 1983.

[13] D. M. Lucantoni, S. M. Kathleen and M. F. Neuts,"A single-server queue with server vacations and a class of non-renewal arrival processes", *Advances in Applied Probability*, 22.03 (1990): 676-705.

[14] M. Neuts, *Structured stochastic matrices of M/G/1 type and their applications,* Marcel Dekker New York, 1989.

[15] Y. Liu, C. Yuen, X. Cao, N.U. Hassan, and J. Chen, "Design of a scalable hybrid MAC protocol for heterogeneous M2M networks", *IEEE Internet of Things Journal*, vol. 1(1), pp.99-111 , 2014.

[16] C. Park, D. Hwang and T. Lee, "Enhancement of IEEE 802.11ah MAC for M2M Communications", *IEEE Communication Letters*, vol. 18, no.7, pp. 1151-1154, July 2014.