O-MAC: Opportunistic MAC Protocol for M2M Communication in WiFi White Spaces

Author 1 and Author 2

***Abstract*—Machine-to-machine (M2M) communication comprises of autonomous devices communicating with each other without human intervention. The explosive growth in devices using M2M communication and the shortage of spectrum has made network access for such devices a challenging problem. While M2M communication in the unlicensed bands is attractive from an economic perspective, such bands are also under use by existing technologies such as WiFi. In this paper, we propose a new opportunistic medium access control protocol (O-MAC) to allow M2M communication within white spaces (i.e. periods without traffic) of WiFi networks. The proposed protocol increases the effective utilization of the channel without any significant impact on existing WiFi networks.**

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

I. Introduction

M2M communication is an integral part of the Internet-of-Things (IoT) and enables devices to operate and communicate without human intervention. The unlicensed industrial, scientific and medical (ISM) band is a cost effective option for communication for wireless M2M devices with limited mobility. However, this spectrum is already used by existing networks, with WiFi being a major user. WiFi is ubiquitously present in most homes and commercial infrastructure in many countries. Thus, the deployment of M2M devices in these areas can create interference, causing collisions and packet loss. However, it has been observed that WiFi networks are occasionally heavily underutilized [1]. This paper considers the problem of coexistence of M2M and WiFi communications.

The use of unlicensed spectrum for M2M devices and their coexistence with WiFi networks has been investigated in literature. In [2], the feasibility of using WiFi white spaces (defined as periods where WiFi nodes do not have any packets to send) for opportunistic M2M communication was evaluated and it was shown that sufficient number of white spaces are available even in heavy WiFi traffic. In [3] the authors proposed the sending of clear-to-send (CTS) messages to block WiFi traffic in order to facilitate data exchange between medical sensors that use ZigBee to transmit their data. However, the proposed scheme requires the M2M devices to also have the capability to send WiFi CTS packets, thereby increasing the cost of these devices. In [4] the authors proposed to send CTS packets from the access point (AP) to reserve the channel for ZigBee communication. The CTS blocks the WiFi nodes for 32 ms and allows ZigBee nodes to communicate in this interval. Such CTS packets are repeated every 200 ms. ZigBee nodes communicate using a simple TDMA procedure with pre-allocated slots. Reservation of the WiFi channel for such long duration introduces large delays in WiFi packets. Also, WiFi nodes may not honor long CTS durations if they do not hear any WiFi transmission after a CTS frame, as observed in [5]. A CTS followed by a long period with no WiFi transmission may also be interpreted as a CTS denial-of-service attack [6].

This paper proposes an opportunistic MAC protocol (O-MAC). Rather than putting M2M devices as competitors to WiFi and thus degrading both WiFi and M2M performance, we opportunistically use the white spaces in WiFi traffic for M2M communication. A white space starts when none of the WiFi devices have any packets to send. Thus, in O-MAC, WiFi packets already in queue are not interrupted. The AP reserves the channel for M2M communication by sending a modified CTS (mCTS) frame. We keep the duration reserved by mCTS frames for M2M communication small (1-3 ms) so that new WiFi packet arrivals will not experience large delays. We also show that the throughput of WiFi networks is not affected even by heavy M2M traffic as our protocol is opportunistic where WiFi devices are considered the primary users.

The rest of the paper is organized as follows. In Section II we describe the design of our MAC protocol. We present simulations results to evaluate the performance of O-MAC in Section III. Section IV presents the conclusions.

II. O-MAC: Opportunistic MAC Protocol

This section describes the proposed opportunistic M2M MAC protocol. The M2M communication model consists of one AP serving n M2M nodes and m WiFi nodes. The AP informs the M2M nodes when a white space starts. The M2M nodes then contend for the channel and successful M2M nodes transmit their data to the AP in their respective allotted slots. The following sequence describes the operation of proposed MAC protocol, as shown in Figure 1.

1) When the AP observes that there are no more packets to be sent in its MAC layer queue, the white space starts. To allow any pending uplink traffic from WiFi nodes, the AP waits for time T_w.
2) After waiting for T_w, the AP assumes that there are no uplink packets to be sent and it proceeds to allow M2M communication to take place. The AP sends a mCTS frame to reserve the channel for M2M communication, in which M2M nodes have to complete a communication cycle. If there are no M2M devices with data to transmit, the AP frees up the channel for WiFi transmissions.
3) When the WiFi nodes receive a mCTS frame, they defer transmission of packets for the duration communicated by the mCTS frame.

Author 1 and 2 are with the Department of XXX at Organization YYY, Country ZZZ (e-mail:author1@domain.name, author2@domain.name).

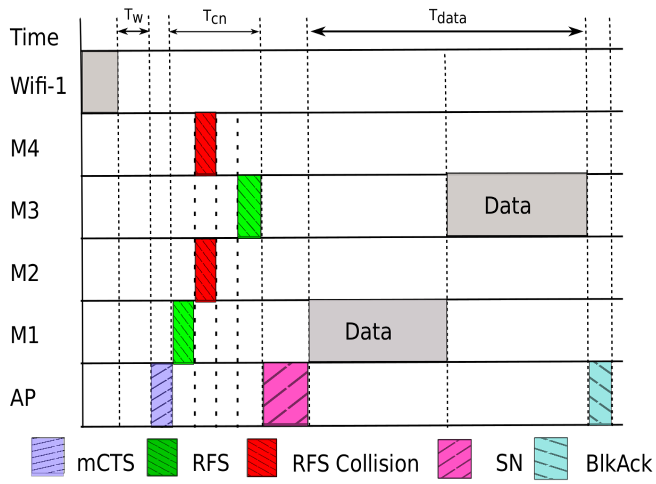

Fig. 1. Operation of the proposed MAC protocol.

4) On the other hand, M2M nodes interpret the mCTS frame as a signal to start the M2M communication cycle. A mCTS frame contains the following information: duration of the M2M communication cycle (T_{max}), the number of contention slots (L), and the contention probability (p). A cycle consists of the following five stages:
   a) Alert stage: M2M nodes receive the mCTS frame from the AP and note the values of L and p.
   b) Contention stage: M2M nodes contend for the channel by sending a Request-for-slot (RFS) frame to the AP by randomly choosing one slot from the L contention slots and transmitting in it with probability p.
   c) Notification stage: The AP notifies the result of the contention by sending a slot notification (SN) packet. This packet also specifies the slots in which successful nodes may transmit in the data transmission stage.
   d) Data transmission stage: The M2M nodes send their data in the slots assigned to them by the AP.
   e) Acknowledgment stage: The AP sends a block acknowledgment (ACK) for all the data packets received from M2M nodes in the data transmission stage.
5) The WiFi nodes resume their normal operation after the completion of the M2M communication cycle.

The various stages in the M2M communication cycle are described in detail below.

A. Alert Stage

The alert stage informs all M2M nodes about the start of the M2M cycle. This is done through the transmission of a mCTS message by the AP, whose structure is shown in Figure 2(a). The mCTS message starts with frame control and duration fields as in usual CTS messages, followed by a 1 byte mCTS frame control field that allows M2M nodes to recognize the message as the start of a M2M communication cycle. The duration field indicates the length of the M2M communication cycle, T_{max}. This allows the WiFi nodes to update their network allocation vector (NAV) so that they do not contend for the channel in this period. Further, mCTS messages contain the values of L and p. The methodology to obtain the values of L and p is described in Section II-E.

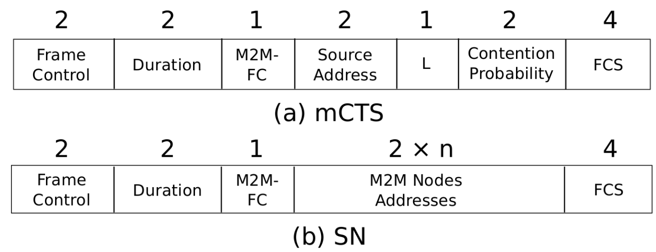

Fig. 2. Structure of (a) mCTS and (b) Slot Notification (SN) messages.

B. Contention Stage

After receiving the mCTS message, M2M nodes contend for channel access in the contention stage. This stage is divided into L slots of equal duration, called RFS duration (RFSD). M2M nodes wait for a period of SIFS (short interframe space) after receiving the mCTS message and then contend for the channel as per a modified slotted ALOHA protocol. A M2M node with data to send randomly chooses one of the L slots, and with probability p, sends a RFS message to the AP in that slot. If there is no collision due to simultaneous RFS transmissions from other nodes, the AP receives the RFS successfully and adds the M2M node to the list of successful nodes who will be allotted data slots. If there is a collision, then the AP records it as such to adjust its estimate of the number of M2M nodes with data to transmit.

C. Slot Notification Stage

If there is any successful RFS transmission during the contention stage, the AP sends the list of successful nodes in a slot notification (SN) message. The order in which successful nodes are listed in the SN message implicitly indicates the order of data slots assignment to the nodes. Figure 2(b) shows the structure of a SN frame. The SN messages have a format similar to IEEE 802.11 BLK-ACK packets which have variable payloads. The payload contains the list of successful M2M nodes and the duration field is updated to communicate the remaining reserved time in the M2M communication cycle.

If RFS contention is not successful because no M2M node has any data to send, then the AP sends a SN-ACK packet which is identical to an IEEE 802.11 ACK packet, and the duration field is set to zero. This resets the NAV of all the WiFi nodes and they resume their data transmission.

If the contention fails due to collision between M2M nodes, the AP sends another mCTS packet with updated T_{max}, L and p values to adjust for the reduced duration available. The frame control field of this SN packet is the same as that of IEEE 802.11 CTS packets and it allows M2M nodes to contend again for channel access. After an unsuccessful contention period, the AP dynamically decides whether to continue the contention period by sending another mCTS packet or to quit and hand over the channel to the WiFi network based on the remaining reserved time and WiFi packet arrivals at the AP.

D. Data Transmission Stage

In the data transmission stage, nodes that successfully contended transmit data in their respective data slots in TDMA fashion. Data slots are separated by a guard time and each slot is of fixed length for simplicity. After the data packets are received by the AP, the AP sends a block ACK (BlkAck). This is the last stage and called the acknowledgment stage (AN). The BlkAck is sent by the AP for the uplink packets received successfully. The AP may send downlink packets at the end of data transmission stage by adding extra data slots. The acknowledgment for downlink packets is sent immediately after receiving the data packets.

E. Parameter Estimation

The AP sends the values of L and p to the M2M nodes in the mCTS messages. These values should be chosen so as to maximize the channel utilization, which is achieved when all possible data slots in the data transmission stage are occupied. This in turn requires that an adequate number of M2M nodes successfully contend for the channel. The parameter estimation process at the AP starts with the number of possible data slots, N_D, which is given by

$$N_D = \frac{T_{max} - T_{SN} - T_{ACK} - LT_C}{T_D} \tag{1}$$

where T_C, T_{SN}, T_{ACK} and T_{DS} are the time taken for a contention slot, slot notification packet, acknowledgment and a data slot, respectively. Then, L and p should be chosen such that the expected number of successful M2M nodes in the contention stage should be at least N_D. In a contention stage with L slots and n active M2M nodes, the expected number of successful slots, $E[S]$, for a given p is [8]:

$$E[S] = pn\left(1 - \frac{1}{L}\right)^{pn-1} \approx pne^{-(pn/L)}. \tag{2}$$

For a given value of L, the optimum value of p that maximizes the number of successes can be obtained by differentiating (2) with respect to p and equating to 0. Thus we get,

$$p = \frac{L}{n} \implies E[S] = Le^{-1}. \tag{3}$$

Since our objective is to ensure $E[S]$ equals the number of available data slots, N_D, we have, $L = eN_D$. Substituting this value of L in (1), we have

$$N_D = \left\lfloor \frac{T_{max} - T_{SN} - T_{ACK}}{T_D + eT_C} \right\rfloor. \tag{4}$$

Thus The AP starts with an estimate of N_D using (4) and the contention stage consists of $L = eN_D$ slots with $p = eN_D/n$. To estimate n, the AP can use the methodology of [7] which is based on measuring the success probability in the contention stage of the previous cycle [7]. Finally, if a contention stage only contains collisions and the AP proceeds with another contention stage, the process starts again with (4) but with T_{max} updated as $T_{max} = T_{max} - T_{SN} - LT_C$.

III. Simulation Results

This section presents simulation results to evaluate the performance of the proposed O-MAC protocol. The protocol was implemented using the NS3 simulator. The simulated network consisted of one AP, m WiFi nodes, and n M2M nodes. Keeping the number of WiFi nodes fixed at $m = 5$,

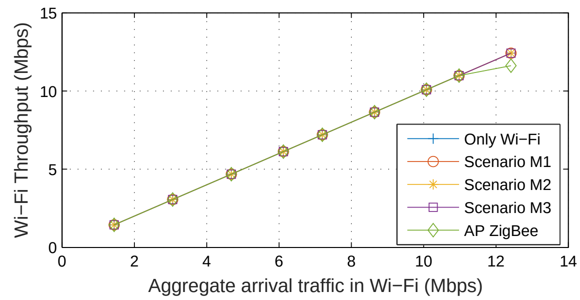

Fig. 3. Effect of M2M traffic on the throughput of WiFi network.

three scenarios were studied. Scenario M1 had $n = 4$ M2M nodes and M2M traffic was kept at 55 packets per second per node. The size of each M2M packet was 85 bytes. In scenario M2, $n = 4$ and M2M nodes generated 222 packets per second per node. In scenario M3, n was kept at 100 and M2M devices generated 3.45 packets per second per node. T_{max} was set to 2.5 ms to keep the delay small. The value of T_W was set to $2CW_{min}$ where CW_{min} is minimum contention window for backoff mechanism in WiFi. We compare our results with the protocol described in [4] where ZigBee devices communicate in their reserved period of 32 ms.

The arrival traffic at WiFi nodes was generated using a batch Markovian arrival process (BMAP) since BMAP is a general arrival process that may be used to model various kinds of traffic including voice, data and video [9]. The 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 for various scenarios. It can be seen that there is no change in the throughput of the WiFi network even with heavy M2M traffic. This is because O-MAC utilizes the white spaces or idle periods in WiFi traffic for transmitting M2M packets. On other hand, [4] reserves the channel exclusively and thus at higher WiFi traffic, packet drops and network congestion is experienced.

The delay experienced by WiFi packets is shown in Figure 4. The increase in delay due to the addition of M2M traffic is of the order of 1 ms even at high M2M and WiFi loads, when T_W is at least $2CW_{min}$. 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 6.3 ms. On other hand at $\rho = 0.9$ the delay in [4] is 400 ms.

Figure 5 shows the effective utilization of M2M communication opportunities by plotting the number of mCTS packet transmissions and the number of M2M communication cycles with data transmission. M2M communication cycles may not have any data transmission when the M2M nodes do not have any data to send, as can be seen in scenarios M1 and M3 where M2M traffic is lower. On the other hand in scenario M2 where M2M traffic is heavier, all mCTS transmissions resulted in communication cycles with M2M data transmission. Note

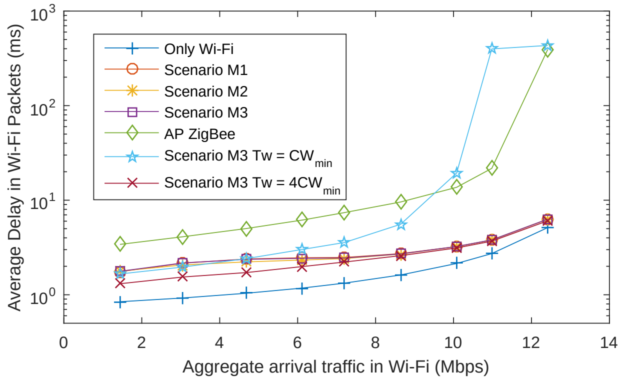

Fig. 4. Effect of M2M traffic on delay of WiFi packets.

that when M2M nodes do not have any data to send, the AP sends an empty SN packet immediately after the contention period, resetting the NAV of all WiFi nodes and thus releasing the reserved channel. This helps to reduce the delay of WiFi packets and effective utilization of channel resources. As the delay introduced by a mCTS and a contention period is around 100-150 μs, the effect of sending a mCTS with no M2M traffic on WiFi delay is of the order of a fraction of a millisecond.

Figure 6 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. scenario M1 and M3). Under saturated traffic (scenario M2) the number of M2M transmissions initially increases with the WiFi traffic and then decreases. This is because at low WiFi traffic, the number of white spaces is small but of relatively longer duration [2]. As the WiFi traffic increases, the idle periods in WiFi network are interrupted by frequent arrival of WiFi packets which decreases the average duration of the white spaces but increases their number. This increase in the number of white spaces allows more opportunistic M2M transmissions. However, a further 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.

The effect of T_W on the performance of O-MAC is shown in Figures 4 and 6. Smaller values of T_W increase the WiFi packet delay, since the AP may prematurely start M2M communication cycles even though nodes have uplink traffic. On the other hand, larger values of T_W waste the white space is wasted while the AP is waiting, and consequently, the number of M2M packets transmitted is smaller. A choice of $T_W = 2CW_{min}$ provides a compromise which does not significantly degrade WiFi performance and at the same time, gives sufficient opportunities for M2M communication.

IV. Conclusion

This paper proposed an opportunistic MAC protocol for M2M communication that exploits the white spaces in WiFi networks. It is a cost effective solution as existing infrastructure is used for M2M communication without affecting WiFi traffic, with only minor changes to the MAC layer of the AP.

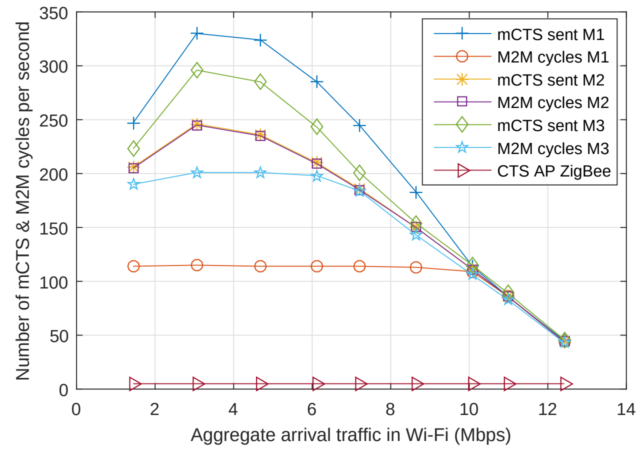

Fig. 5. Effective white space utilization: number of mCTS sent and successful M2M communication cycles.

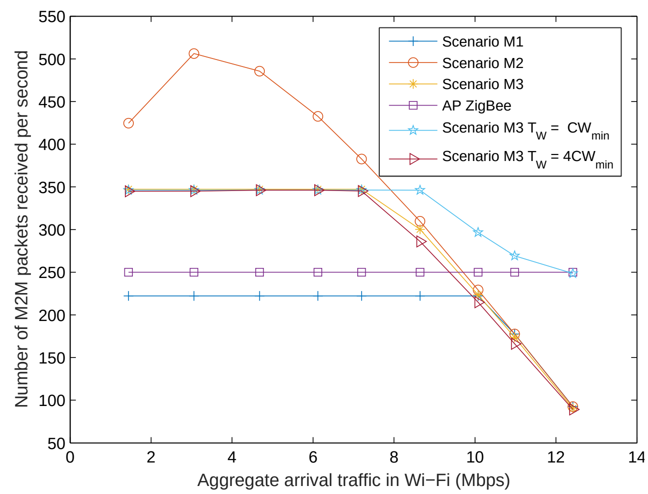

Fig. 6. Number of M2M packets received per second.

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] 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.

[8] 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.

[9] 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.