RM-MAC: A Routing-Enhanced Multi-Channel MAC Protocol in Duty-Cycle Sensor Networks

Ye Liu*, Hao Liu†, Qing Yang‡, and Shaoen Wu§
*National ASIC System Engineering Research Center, Southeast University, Nanjing, 210096, China
†Suzhou Key Lab of Integrated Circuits & Systems, Southeast University, Suzhou, 215123, China
‡Department of Computer Science, Montana State University, Bozeman, MT 59717, USA
§Department of Computer Science, Ball State University, Muncie, IN 47306, USA

Abstract—**Multi-channel media access control (MAC) is important in wireless sensor networks because it allows parallel data transmissions and resists external wireless interference. Existing multi-channel MAC protocols, however, do not efficiently support delay-sensitive applications that require reliable and timely data transmissions. In addition, multi-channel operation is inherently deficient for supporting multi-hop broadcasting, due to independent waking-up schedules on sensors. To address these issues, we present a routing-enhanced multi-channel MAC (RM-MAC) which allows nodes to coordinately select their channel polling times based on cross-layer routing information. RM-MAC also supports a ripple broadcast mechanism which achieves efficient multi-hop broadcast among sensors. Simulation results show that RM-MAC provides significant improvement over the MuCHMAC [1], in terms of end-to-end delay, under a wide range of traffic loads including both unicast and broadcast traffic.**

I. INTRODUCTION

The advancements in wireless communication and micro-electro-mechanical systems have made wireless sensor networks (WSNs) increasingly applied to a broad range of applications such as target tracking, industrial process monitoring and smart building. Typically, a WSN usually consists of a large number of battery-powered sensing nodes that are deployed in the target area over several months for reporting sensed data to sink node.

Energy efficiency is one of the primary goals when designing MAC protocols in WSNs since sensor nodes normally have very limited battery capacity. An important mechanism to mitigate energy consumption in sensor networks is duty cycling. In this technique, each node periodically alternates between active and dormant state according to a working schedule. In the active state, a node is able to transmit or receive packet, whereas in dormant state, the node completely turns off its radio to save energy, except a timer to wake itself up.

Recently, multi-channel MAC protocols have attracted considerable interests from the research community [2], since current WSN hardware such as MICAz and TelosB already support multiple channels communication. In addition, existing MAC protocols using only a single radio channel for all transmissions cannot provide efficient multi-task support or reliable and timely communication under event-driven applications. The demand for simultaneous data transmission often causes severe channel collision when using single channel MAC protocol [3], [4], [5]. Furthermore, given WSNs share the same 2.4GHz ISM band with WiFi devices capable of higher transmission power and larger coverage range, the presence of interference over the same frequency band may lead loss of data packets, transmission delay, false alarms and loss of synchronization [6], [7].

Although existing multi-channel MAC protocols obtain better communication performance than single channel MAC protocols by mitigating interference from sensor nodes in the network and other types of devices outside the network, further performance improvement are expected. On the one hand, WSNs have been extended to support delay-sensitive applications, in which multi-hop data forwarding should fast enough [8]. To design a good MAC protocol for these applications, delay and throughput must be considered as well as energy efficiency. Unfortunately, the duty cycling operation poses challenge on the sleep delay [9]. When a node transmits a packet in a multi-hop network, it must wait for a long time until the forwarding node wakes up, causing large end-to-end delivery latency. On the other hand, broadcast is a common and vital communication primitive that delivers a message from a root node to all other nodes inside the network. It is usually used for data dissemination, route discovery and network configuration [10], [11]. However, multiple channels access approaches are weak for supporting multi-hop broadcast due to frequency diversity of neighbors. A broadcast packet has to be transmitted multiple times so that the packet can be received by all nodes in the neighborhood.

To overcome the challenges above, in this paper, we present a new multi-channel MAC protocol, called Routing-enhanced Multi-channel MAC (RM-MAC). RM-MAC attempts to reduce delivery latency and provide multi-hop broadcast support in an energy efficient fashion. The key idea of RM-MAC is to assign channel polling [12] time based on cross-layer routing information. The proposed protocol effectively addresses sleep delay and multi-channel broadcast problem.

The remainder of this paper is organized as follows: Section II describes related work in multi-channel MAC protocol for wireless sensor networks. In section III, the detailed design of RM-MAC is presented. Evaluation results from simulations are shown in Section IV. Finally, in Section V, we present conclusions.

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

II. Related work

Several multi-channel MAC protocols for wireless sensor networks have been proposed in the literature. These approaches can be roughly categorized into three categories: static, control channel-based and dynamic multi-channel MAC protocols.

MMSN [13] is the first multi-channel MAC protocol for WSNs. Four static frequency assignment schemes are introduced. In order to support broadcast communications, each time slot is divided into a broadcast contention period and a transmission period. Y-MAC [14] is a control channel-based protocol which uses a dedicated control channel and several data channels for multi-channel transmissions. In a broadcast period, all nodes wake up and listen on the control channel for broadcast and control message. Each node, according to a distributed time slot assignment method, picks one of unicast slots to receive packets.

EM-MAC is presented in [15]. Instead of using control channel as rendezvous for data exchange, every node in EM-MAC dynamically selects one of available channels and wakes up to send beacon based on its own duty cycle schedule. To communicate, senders predict the receivers wake-up channel and wake-up time by using pseudo-random number sequence. MuCHMAC [1] is a TDMA-based dynamic multi-channel MAC protocol. Like EM-MAC, each node with MuCHMAC uses an independently generated pseudo-random sequence to control its receiving channel for each slot. It also introduces a TDMA timing optimization mechanism to overcome channel collision problem for dense network, in which a slot is divided into several subslots and each node randomly selects one of them for low-power listening. In order to provide broadcast support, the same channel number is periodically generated for all nodes in every $u : b$ unicast slots.

Most recently, B. A. Nahas et al. [16] propose MiCMAC, which is a channel hopping variant of ContikiMAC and independent of upper layers of the protocol stack, as well as operates in a distributed way. Experimental results demonstrate that extending low-power listening with channel hopping is an effective and practical solution to mitigating interference in large wireless sensor networks.

III. RM-MAC protocol design

A. System model

We consider a wireless sensor network with *n* nodes arbitrarily distributed in the sensing field. Each node is equipped with a single half-duplex radio to transmit or receive packet. The lifetime of the entire network is divided into several fixed-length frames, and each frame contains a number of time slots with equal length. Inspired by MuCHMAC, each time slot in RM-MAC is further split into multiple subslots, which is long enough for a round-trip packet transmission.

Sensor nodes in the network operate at a low-duty-cycle model and alternate between the active state and the dormant state [17]. In the active state, a node opens its radio to receive or transmit data packets. In the dormant state, it turns off all its function modules expect a timer for waking itself up. Each node picks one of subslot in every slot to poll channel for possible traffic. The remaining subslots serve for the dormant state. So a node can receive a packet only in its channel polling time unit and wake up to transmit packet according to working schedule of the intended receiver.

B. Frequency selection in each time slot

RM-MAC adopts hybrid design (combining FDMA and TDMA) to avoid interference between transmissions of neighboring nodes under high loads and burst traffic. Every node independently decides its receiving channel for each time slot. Similar to EM-MAC and MuCHMAC, pseudo-random channel-switching scheme is used in RM-MAC, so that a sender can accurately predict the listening channel of a receiver as well as reduce memory overhead. Equation (1) is taken as our pseudo-random function to generate channel hopping sequence for a node.

$$C_{n+1} = (aC_n + b) \mod n. \tag{1}$$

where n is the number of available channels, C_n is the current listening channel, each C_{n+1} generated can be used as listening channel in the next time slot and becomes the new seed. Parameter a and b are the $multiplier$ and $increment$, respectively.

C. Channel polling subslot assignment

After frequency selection, all nodes get a channel for data reception in every time slot. In order to reduce collisions caused by nodes waking up on the same channel, each node independently selects its channel polling subslot.

We first analyze the multi-hop transmission probability in a single time slot in the MuCHMAC protocol. Let $P(n)$ denote the probability that a packet can be delivered at least n-hop within a single time slot. The number of subslots divided in a time slot is denoted as N_s. The chance that a packet can be delivered in no less than 1-hop is 1, and more than N_s-hop is 0. Therefore,

$$P(n) = \begin{cases} 1 & \text{when } n = 1 \\ 0 & \text{when } n > N_s \end{cases} \tag{2}$$

In other cases, when relaying nodes wake up after the packet is received by their downstream nodes, the packet can be forwarded multiple hops in one time slot. For example, when $N_s = 3$, if the three successive upstream nodes(*C*, *B* and *A*) of sender(node *D*) wake up at the $1st$, $2nd$, and $3th$ subslot respectively, a data can be transmitted three hops from node *D* to node *A* in one slot. If node *C* wakes up at the $3th$ subslot, node *D* will transmit data at that time, which cannot be sent to the next node during the same time slot. Therefore, we can derive the following equation:

$$P(n) = \begin{cases} \dfrac{\sum_{i=1}^{N_s-1} i}{{N_s}^2} & when\ n = 2 \\ \dfrac{\sum_{j=1}^{N_s-n+1} \sum_{i=1}^{j} i}{{N_s}^n} & when\ 2 < n \le N_s \end{cases} \tag{3}$$

where ${N_s}^n$ is the number of whole possible outcomes that all nodes on a multi-hop path pick randomly out of subslots. The numerator is the number of permutations that a packet can be delivered at least n-hop within a single time slot. Therefore,

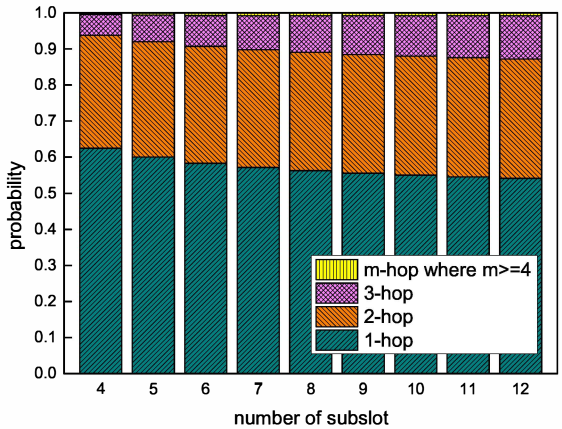
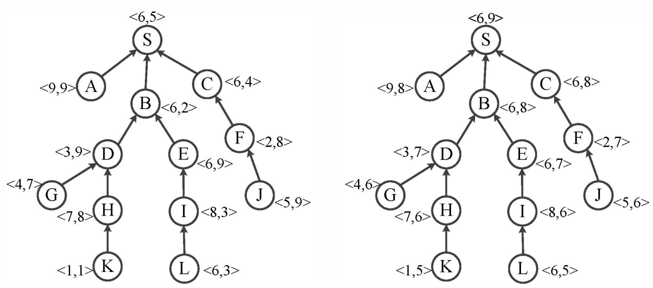


Fig. 1. Multi-hop probability within a single time slot of MuCHMAC.



(a) MuCHMAC working schedule (b) RM-MAC working schedule

Fig. 2. An example of convergecast traffic in a multi-hop sensor network.

the probability of a packet forwarded only n-hop can be captured by $P(n) - P(n+1)$. Fig. 1 provides information on MuCHMAC's capability of multi-hop delivery within a single time slot. Overall, the probability varies slightly. More than 50 per cent of packets are forwarded only 1-hop in all situations. The percentage of packets forwarded less than 4-hop is up to 99.9% even though each time slot is divided into twelve subslots.

To address the above-mentioned problem, RM-MAC exploits a cross-layer design to mitigate sleep latency introduced by duty cycling as well as maximizing parallel transmissions. Every node in RM-MAC calculates channel polling subslot assignment based on its hop distance to the sink node. The detail process is presented with the following example. In this manner, an intermediate relaying node does not have to wait for a long time to forward the packet to its next upstream node. On the other hand, horizontal interference is eliminated by channel assignment, and subslot assignment eliminates vertical interference.

Fig. 2 shows an example of unicast traffic, where node S and others are the sink and sensing nodes, respectively. Suppose the working schedule of nodes is represented by $< C_i, S_j >$, where C_i means that a node wakes up at channel C_i, and S_j is the channel polling subslot of this node. As aforementioned, the wake-up channel is determined by a pseudo-random generator in both MuCHMAC and RM-MAC. Instead of randomly choosing one of subslots, the channel polling subslot of each node in RM-MAC is calculated by

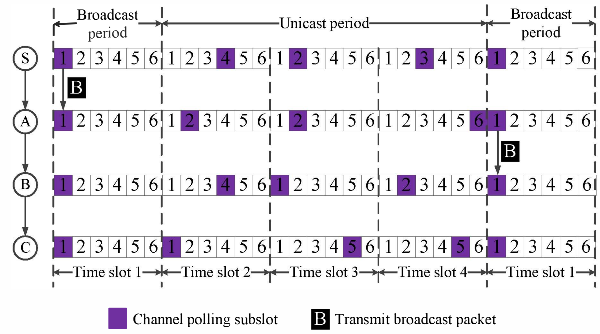


Fig. 3. Broadcast and unicast period assignment in MuCHMAC.

the following equation:

$$S_j = N_s - \ell \bmod N_s \tag{4}$$

where N_s is the number of subslots divided in a time slot as mentioned above, ℓ is the hop distance to the sink, which can be easily obtained from gradient-based routing protocols. It is worthwhile to note that hop distance information can be also obtained from other kinds of routing protocols like on-demand and geographical routing protocols. Another method to achieve hop distance information can be done by broadcasting beacon frame. Sink node broadcasts beacon frame with *hop 0* periodically. Its neighbors receive this packet and forward the beacon frame with *hop 1*, and so forth.

Assuming node *K* has a packet to send to the sink node *S*. With working schedule of MuCHMAC, it wakes up to send the packet to node *H* at subslot 8, and this packet can be forwarded to node *D* at subslot 9. However, node *D* has to wait for its next upstream node *B* to wake up to receive the packet in the next time slot. Finally, the packet arrives on sink node *S* at subslot 5 of next time slot. In RM-MAC, all node along the data forwarding path wake up sequentially from subslot number 6 to 9. It is easy to find that data packet with RM-MAC can be forwarded multiple hops within a single time slot and thus additional latency caused by duty cycling is reduced. Moreover, when burst or high traffic load occur in the network, multiple nodes send data to the sink node simultaneously. As shown in Fig. 2(b), nodes of the same depth to the sink(such as node *K* and node *L*) send packet at the same time, but their intended receiver node *H* and node *I* wake up on channel 7 and 8, respectively. Although both node *B* and node *E*'s wake-up channel are channel 6, they receive packet from node *D* and node *I* at different time. Therefore, the probability of network congestion can be mitigated.

D. Ripple broadcast mechanism

In MuCHMAC, a broadcast period is inserted in every $u : b$ unicast slots, in which all nodes inside network switch to a same frequency at the same time for transmitting or receiving a broadcast packet. As shown in Fig. 3, one broadcast slot is inserted in every three unicast slots and node *S*, *A*, *B* and *C* all wake up at the first subslot on a broadcast channel. Since one broadcast transmission of a sender could reach all its neighbors, broadcast communications do not have to be achieved through redundant unicast transmissions. However, the large interval between two broadcast periods increases delivery latency substantially. In addition, the whole

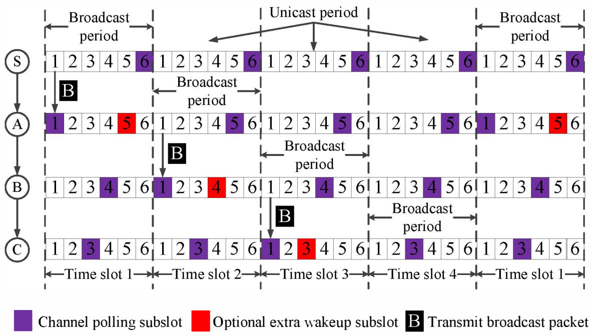


Fig. 4. Broadcast and unicast period assignment in RM-MAC.

nodes wake up simultaneously, causing severe broadcast storm problem [18] when network density increases.

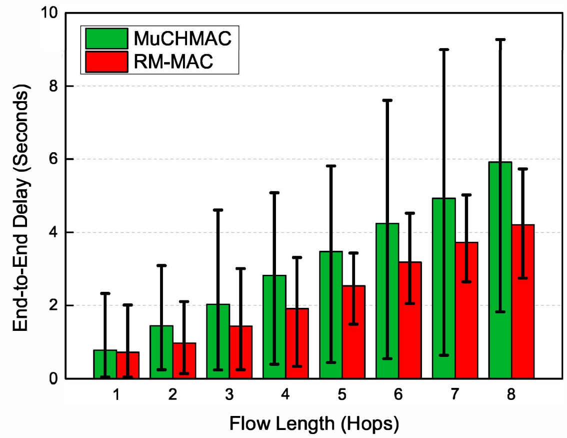
RM-MAC introduces a ripple broadcast mechanism that can effectively accelerate multi-hop forwarding of broadcast packet and mitigate the broadcast storm issue. Fig. 4 demonstrates the principle of ripple broadcast mechanism, in which each node selects broadcast period according to its hop distance to the sink node. For instance, the broadcast period of node *S*, *A*, *B* and *C* are the *1st*, *2nd* , *3rd* , and *4th* time slot, respectively. The sink node *S* sends a broadcast packet to node *A* at time slot 1. This packet then can be forwarded from node *A* to node *B* at time slot 2. After that, node *B* transmits it to node *C* at time slot 3. In this way the interval between consecutive broadcasts decreases, since a sender need not to wait for a long time until its neighbors wake up for broadcast reception again. As neighbor nodes in different layers do not wake up at the same time, RM-MAC also mitigates redundant reception and medium contention.

IV. Performance Evaluation

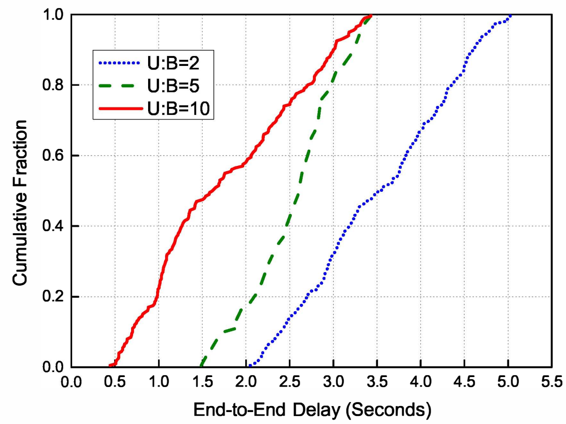
In this section, we compare RM-MAC with MuCHMAC in NS2. Table I summarizes the key parameters we used in our simulations. The duration of time slot is set to one second, which is further divided into 10 subslots with equal length in both RM-MAC and MuCHMAC. We varied the total number of available channels to illustrate how it affects the performance of RM-MAC and MuCHMAC. The ripple broadcast scheme has a chance to abort sequential transmission of unicast packet, and the simulation results show that different settings of $u : b$ have a great impact on end-to-end unicast delivery latency. For fair comparison with MuCHMAC, extra wakeup subslot that ensure sequential transmission is not added in RM-MAC and each node with RM-MAC and MuCHMAC does not send unicast packet during broadcast period.

TABLE I. Simulation Parameters

Tx range	250 m
Carrier sensing range	550 m
Backoff window	31-255
Retry limit	5
Time slot	1 s
Number of subslots	10
Number of channels	4 or 8
Unicast to broadcast periods	2,5 or 10



(a) End-to-end delay versus flow length



(b) CDF of end-to-end delays

Fig. 5. Performance for time-driven traffic in linear network.

A. *Under time-driven traffic*

In the first group of simulations, we compare RM-MAC with MuCHMAC in a linear network with 9 nodes. Each node is 200 meters away from its neighbors, and the last node is the sink. There is a single CBR (constant bit rate) data flow starting from other sensor nodes to the sink node in every 20 seconds. The length of flow varies from 1 hop to 8 hops. Eight available channels are used, and the value of $u : b$ is set to 5.

Fig. 5(a) shows the average and maximum/minimum end-to-end latency achieved by MuCHMAC and RM-MAC with increasing number of hops. The latency of both protocols increases linearly with the number of hops. However, MuCHMAC has a much bigger rate of increase than does RM-MAC. When the path length is increased to 8 hops, the average and maximum latency of packets in MuCHMAC are 5.92s and 9.28s respectively. The values in RM-MAC are 4.21s and 5.73s respectively. RM-MAC reduces average end-to-end delay by 29%, and reduces delay bound by 38%. The reason is that each packet with MuCHMAC has to wait for half of one time slot on average in each hop, while RM-MAC is able to forward a packet multiple hops within a single time slot. It is also found that the lower delay bound in RM-MAC increases suddenly when the flow length is 5 hops. This sharp rise is due to ripple broadcast scheme. Sequential transmission must be suspended as the value of $u : b$ is chosen to be 5. No matter when a source

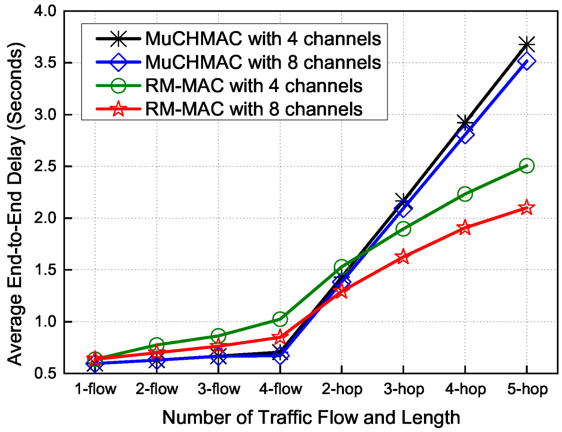


Fig. 6. Performance for event-driven traffic in grid network.

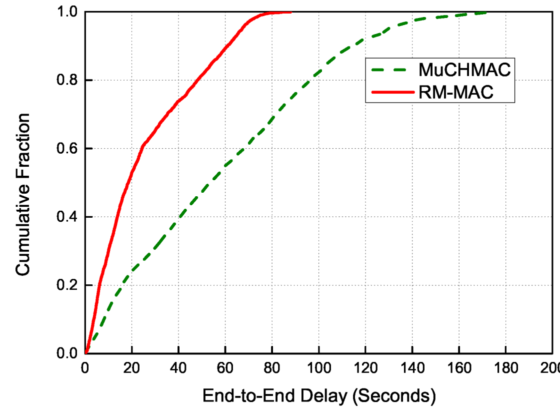
node sends a packet, one of relaying node must wake up for potential broadcast data, and does not forward the receiving data in its broadcast period.

To gain insight into how the packet delivery latency varies with different values of $u : b$, we evaluated latency again on the 5-hop linear network scenarios. Fig. 5(b) shows the CDF of end-to-end latency with RM-MAC under different situations. Average end-to-end delay under the ratio of unicast to broadcast periods 10, 5, and 2 are 1.77, 2.54, and 3.53 seconds, respectively. This is because when the ratio decreases, broadcast period status rises, and thus the phenomenon of transmission interruption happened frequently.

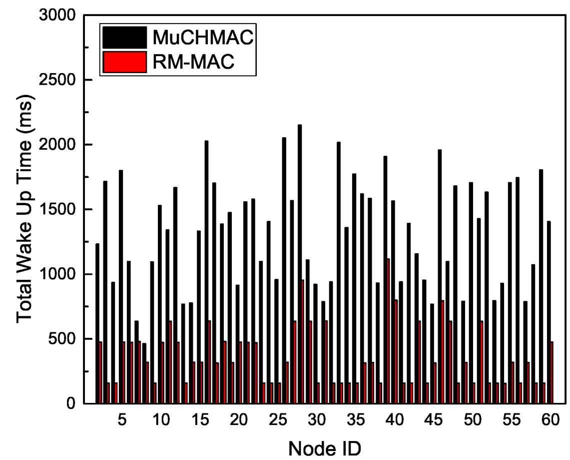
B. Under event-driven traffic

In the second group of simulations, we compare RM-MAC with MuCHMAC in grid network, each node is 200 meters away from its neighbors at different layers and 100 meters from nodes at the same layer. Such that all nodes in the same layer are within the radio interference range of each other. The node at the bottom of each flow is the source node, with hop count ranging from 1 to 5. When an event is detected, the source generates packet and sends it to destination. We varied interference intensity by varying the number of flows. The ratio of unicast to broadcast periods is 10 to 1. Different total number of available channels is used to show how it affects the performance of RM-MAC and MuCHMAC.

The left part in Fig. 6 shows the performance of the two protocols when one-hop traffic flow number increases from 1 to 4. The right part shows the performance of both protocols when there are 4 contending traffic flows, with lengths ranging from 1 to 5 hops. With increasing number of contending flows in the one-hop scenario, delivery latency in RM-MAC increases at a faster rate than does MuCHMAC. This is mainly because when sensing an interesting event, all source nodes in RM-MAC send the data to their destination at same subslot, while source nodes in MuCHMAC wake up at different subslots, maximizing parallel transmission among nodes of the same layer. On the other hand, when the number of available channels increases from 4 to 8, RM-MAC's packet delivery latency declines, since it utilizes multiple orthogonal radio channels to enhance the amount of parallel traffic in the same layer. However, the packet delivery latency of MuCHMAC degrades as the flow



(a) CDF of end-to-end delays



(b) Energy consumption for receiving broadcast packets

Fig. 7. Performance for broadcast traffic in 60-node random network.

length increases. The reason is an intermediate relaying node needs to wait for a long time until its upstream node wakes up to receive the packet. RM-MAC outperforms MuCHMAC in multi-hop traffic flows because RM-MAC is able to forward a data multiple hops within a single time slot.

C. Under broadcast traffic

In the last group of simulations, we compare RM-MAC with MuCHMAC in a 60-node random network where 60 nodes are randomly distributed in a 1000m $\times$ 1000m square area. One of these nodes is selected as the sink, which is then located at the top right corner of the square. The sink node generates a total of 50 broadcast packets, one every 200 seconds so that all forwarding for one packet completes before the next packet is generated. On receiving a broadcast packet for the first time, a sensor node rebroadcast it to neighbors. The ratio of unicast to broadcast periods is 5 to 1.

The CDF of end-to-end delay for all packets is shown in Fig. 7(a). The average and upper bound for broadcast latency with MuCHMAC are 58.22 and 174.74 seconds, respectively. These values with RM-MAC are 25.87 and 88 seconds, respectively. RM-MAC shows an average end-to-end latency that is around 44% of that with MuCHMAC, and delay bound that is around 50% of that with MuCHMAC. The reason is that RM-MAC's ripple broadcast mechanism allows each node selects broadcast period based on its hop distance to the sink node, and thus a broadcast packet can be forwarded over multiple hops in a single frame, whereas with MuCHMAC, upon receiving a message, senders have to wait for a long time until the downstream nodes wake up at the next broadcast period.

Fig. 7(b) shows the energy consumption of each sensor node for receiving broadcast packets. Total wake up time is used to measure energy consumption. Each packet transmission time is 3.2 *ms*. The sink node originates 50 broadcast packets during the run. However, due to the whole nodes in the network wake up simultaneously, the average wake up time to receive broadcast packet with MuCHMAC is 1328 *ms*, which means each node received a same broadcast packet on average 8 times. In the worst case, node 28 received a total of 672 broadcast packets in the run. Since neighbors in different layers do not wake up at the same time, RM-MAC considerably reduces the redundant reception, in which the total wake up time for receiving broadcast packets by most nodes are no more than 640 *ms*. Overall, RM-MAC achieves lower end-to-end delays and much higher energy efficiency for broadcast traffic.

V. Conclusion

This paper presented the design and evaluation of RM-MAC (Routing-enhanced Multi-channel MAC), a multi-channel MAC protocol for duty-cycle sensor networks. Nodes in RM-MAC calculate channel polling subslot based on its hop distance to sink node. The ripple broadcast mechanism makes broadcast period of nodes sequentially distributing in a frame. We conducted a comprehensive evaluation through NS2, which shows that RM-MAC significantly outperformed MuCHMAC. For example, in the 8-hop flow scenario under time-driven traffic, RM-MAC reduces average end-to-end delay by 29%, and reduces upper delay bound by 38%. Under broadcast traffic in the 60-node random network scenario, RM-MAC achieves average delay that is around 44% and upper bound that is around 50% of that with MuCHMAC. In addition, RM-MAC considerably reduces redundant broadcast reception.

Acknowledgment

This work was partly supported by the National Science Foundation of China (Grant No.61176031) , Natural Science Foundation of Jiangsu (Grant No.BK2011018), the Fundamental Research Funds for the Central Universities and Graduate Research and Innovation Projects of Universities in Jiangsu Province (KYLX_0130).

References

[1] K. S. Joris Borms and B. Lemmens., "Low-overhead dynamic multi-channel mac for wireless sensor networks," in *Proceedings of 7th European Conference on Wireless Sensor Networks (EWSN 2010)*, February 2010, pp. 81–96.

[2] P. Huang, L. Xiao, S. Soltani, M. Mutka, and N. Xi, "The evolution of mac protocols in wireless sensor networks: A survey," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 101–120, First 2013.

[3] M. Ringwald and K. Romer, "Burstmac: An efficient mac protocol for correlated traffic bursts," in *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, June 2009, pp. 1–9.

[4] Y. Liu, F. Jiang, H. Liu, and J. Wu, "Sc-mac: A sender-centric asynchronous mac protocol for burst traffic in wireless sensor networks," in *Communications (APCC), 2012 18th Asia-Pacific Conference on*, Oct 2012, pp. 848–853.

[5] P. Huang, C. Wang, L. Xiao, and H. Chen, "Rc-mac: A receiver-centric medium access control protocol for wireless sensor networks," in *Quality of Service (IWQoS), 2010 18th International Workshop on*, June 2010, pp. 1–9.

[6] L. Angrisani, M. Bertocco, D. Fortin, and A. Sona, "Experimental study of coexistence issues between ieee 802.11b and ieee 802.15.4 wireless networks," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 8, pp. 1514–1523, Aug 2008.

[7] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving wi-fi interference in low power zigbee networks," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10, 2010, pp. 309–322.

[8] P. Suriyachai, U. Roedig, and A. Scott, "A survey of mac protocols for mission-critical applications in wireless sensor networks," *Communications Surveys Tutorials, IEEE*, vol. 14, no. 2, pp. 240–264, Second 2012.

[9] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1567 – 1576 vol.3.

[10] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Achieving efficient flooding by utilizing link correlation in wireless sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 21, no. 1, pp. 121–134, Feb 2013.

[11] Z. Li, M. Li, J. Liu, and S. Tang, "Understanding the flooding in low-duty-cycle wireless sensor networks," in *Parallel Processing (ICPP), 2011 International Conference on*, Sept 2011, pp. 673–682.

[12] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06, 2006, pp. 321–334.

[13] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic, and T. Abdelzaher, "Mmsn: Multi-frequency media access control for wireless sensor networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–13.

[14] Y. Kim, H. Shin, and H. Cha, "Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, ser. IPSN '08, 2008, pp. 53–63.

[15] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "Em-mac: A dynamic multichannel energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '11, 2011, pp. 23:1–23:11.

[16] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt, "Low-power listening goes multi-channel," in *Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on*, May 2014.

[17] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*, Oct 2011, pp. 383–392.

[18] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99, 1999, pp. 151–162.