Peer-to-Peer Netw. Appl. (2018) 11:504–517
DOI 10.1007/s12083-017-0548-6

Cross-layer cooperative multichannel medium access for internet of things

Ye Liu[1] · Chenglin Fan[2] · Hao Liu[3] · Qing Yang[2] · Shaoen Wu[4]

Received: 8 October 2015 / Accepted: 7 February 2017 / Published online: 8 March 2017
© Springer Science+Business Media New York 2017

Abstract Wireless networking in the Internet of Things is a challenging problem because a huge amount of devices in a relatively small region need to be interconnected. Particularly, the carrier sensing multiple access with collision avoidance (CSMA/CA) operation of IoT devices is not viable solution, since dense network leads to high channel contentions. Moreover, given an intensive network traffic load, long queues or even queue overflows are expected, which further deteriorates network performance. To address these issue, multichannel medium access is proposed and it attracts great attention recently. In this paper, we firstly establish models based on combinatorics theory to analyze the performance of dynamic multichannel medium access. Then, a Cross-layer Cooperative Multichannel Medium Access (CCMMA) is proposed to effectively avoid channel contentions by enabling neighboring devices to communicate on orthogonal channels. The CCMMA also introduces a routing-enhanced mechanism that enables relaying nodes to wake up intelligently if there is incoming traffic, that successfully mitigates delay and queue overflow problems caused by low-power operation of IoT devices. The performance of CCMMA is evaluated through extensive simulations. The results show that it provides significant improvement in terms of quality of service over existing solutions.

Keywords Internet of things · Multichannel access · Extremely dense network · Cross layer design

✉ Ye Liu
liuyefancy@seu.edu.cn

Chenglin Fan
chenglin.fan@msu.montana.edu

Hao Liu
nicky_lh@seu.edu.cn

Qing Yang
qing.yang@cs.montana.edu

Shaoen Wu
swu@bsu.edu

1 National ASIC System Engineering Research Center, Southeast University, Nanjing, Jiangsu, China

2 Department of Computer Science, Montana State University, Bozeman, MT, USA

3 Suzhou Key Laboratory of Integrated Circuit and Systems, Southeast University, Suzhou, Jiangsu, China

4 Department of Computer Science, Ball State University, Muncie, IN, USA

1 Introduction

Over the past decade, it has been witnessed the rapid development in Micro-Electro-Mechanical Systems (MEMS), wireless communication and embedded system. These technological advancements contribute to an exciting paradigm, the Internet of Things (IoT), which has drawn great attention from both academia and industry [1–4]. The IoT technology aims at a smart world where objects in our daily life will be capable of sensing environments, communicating and interacting with each other. It will essentially change people's lives, making us healthier, environment more friendly and service more convenient.

One of the most important characteristics in IoT is the huge number of devices. Thanks to the cloud computing and massive data processing [5], storage [6], and encrypted searching [7–9] techniques, it becomes feasible to deploy a large-scale IoT system where hundreds of nodes are connected, including doors, windows, lights, appliances, switches, and air conditioner. The node density of the resulting system can easily reaches one device per square meter on average. This also implies thousands of smart devices in a building and hundreds of millions of devices in a city.

In a smart building, for example, a residential sensing systems is presented in [10], where over 1200 sensors are deployed in 20 homes to form a wireless mesh network. PresenceSense [11] deploys acceleration sensors, ultrasonic sensors and WiFi access points in commercial buildings for non-intrusive detection of individual's presence. Vigil [12] is a real-time distributed wireless surveillance system that can be used in both outdoor and indoor environments. Caraoke [13] is an e-toll transponder network where readers are deployed on street lamps to query nearby transponders. GreenOrbs [14] is a large-scale wireless sensor network system for long-term ecological monitoring in the forest. The sensors collect diverse environmental information and report to the sink node. Sensors can also be deployed in an underwater environment to monitor water quality information [15, 16]. ShakeAlert [17] is deployed in the west coast of the United States for early earthquake warning. In addition, large amount of acoustic sensors are deployed to detection and track moving targets in real-time in a battle field environment [18–20].

Given the huge number of smart devices in IoT, efficient and reliable communication among devices becomes a challenging problem. The challenges are listed as follows. (1) A dense network always implies high channel contentions. Particularly, in an event-driven system where multiple nodes may detect abrupt events at the same time, and transmit a large volume of data simultaneously, resulting in a harsh communication condition. (2) It has been proved that conventional Carrier Sense Multiple Access (CSMA) based algorithms do not efficiently handle burst traffic, and thus network performance deteriorates drastically [21–23]. (3) To save energy, devices often operate in a low-power mode. This will pose challenges because duty-cycling wake-up schedules on nodes may cause extra network delay [24, 25]. When a node reports data to the sink node, for example, it has to wait for a long time until the forwarding node wakes up, causing network delay. Moreover, devices may generate and transmit a huge amount of multimedia data, which will cause the queue length on each node increases sharply, leading to a buffer overflow on queues in the network.

To address these challenges, we present an efficient medium access protocol for a high-density IoT system. Firstly, an analytic model based on combinatorics theory is provided to understand the performance of dynamic multichannel Medium Access Contorl (MAC) protocols. We learn that random channel switching cannot make the best use of spectrum resource, especially, in a high data rate network. To this end, we introduce a cooperative multichannel access mechanism to fully utilize spectrum resource. Secondly, we design a routing-enhanced mechanism based on Time Division Multiple Access (TDMA) optimization mechanism, which not only further eliminates internal contentions but also accelerates data delivery in the network. Since nodes wake up in a ripple fashion, the routing-enhanced mechanism also efficiently addresses the issue of queue overflow problem.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 presents system model and theoretical analysis of existing random channel assignment approaches. The details of the CCMMA protocol and its theoretical performance are provided in Section 4. Simulations and results analysis are shown in Section 5. Finally, Section 6 concludes the paper.

2 Related work

In this section, we review the research works in two areas: multichannel Medium Access Control (MAC) and gradient routing protocol. For MAC, we focus on multi-channel assignment protocols applied in dense networks. For gradient routing protocols, we target at how the gradient information is used to achieve efficient routing, which will be used in the proposed routing-enhanced mechanism.

2.1 Multichannel MAC protocols

In recent years, multichannel medium access attracts growing attention from the research community [26, 27]. This is mainly because recent wireless devices already support multi-channel communications. In addition, medium access with only one single radio channel cannot provide timely and reliable data delivery.

MMSN [28] is the first multi-channel MAC protocol for sensor networks. It introduces four static channel assignment schemes, and users could choose one of them, according to their network environment. Y-MAC [29] is a control channel-based protocol that uses a dedicated control channel and several data/service channels for multi-channel transmissions. The sender and receiver first negotiate on the control channel and then both switch to a mutually agreed data channel to transmit data. The above-mentioned methods improve network performance because interference from internal nodes is avoided. Unfortunately, they are vulnerable to external interference generated by other wireless communications using the 2.4 GHz Industrial Scientific Medical (ISM) band.

In order to mitigate external wireless interference, a dynamic multi-channel control protocol is introduced in MuCHMAC [30]. In this protocol, each receiver switches its channel frequently based on an independent pseudo-random sequence. A sender can deduce its receiver's channel if the same pseudo-random sequence is used. When node density increases, neighboring nodes may choose the same channel at the same time, leading to network collisions. To address this issue, MuCHMAC presents a TDMA timing optimization mechanism in which a duty cycle period is further divided into time slots and a node randomly selects one time slot o wake up. Therefore, the probability of multiple nodes use the same channel in the same time slot reduces largely. We find the TDMA timing optimization not only can help to avoid internal contentions, but also has the potential to address the delay and overflow issues lead by duty cycling wakeup among IoT devices.

Similar to MuCHMAC, EM-MAC [31] is another dynamic multichannel protocol in which a sender predicts the receiver's channel based upon the corresponding pseudo-random function. The wake-up schedule of each node in the network is generated according to a pseudo-random number function. Furthermore, the channel blacklist mechanism in EM-MAC enables a sensor to avoid using bad channels that are interfered by other devices. MiCMAC is proposed in [32] which is a channel-hopping variant of ContikiMAC [33] and independent of upper layers of the protocol stack. Experimental results demonstrate that integrating low-power mode and channel hopping technique is an effective and practical solution to mitigating interference in large-scale sensor networks.

2.2 Gradient routing protocols

One important type of routing protocols in multi-hop wireless network is the gradient-based routing protocol [34–37]. According to the protocol, each node in the network is assigned a degree. The value of the degree is calculated based on several metrics, e.g., hop-count, expected delay, and/or residual energy. The degree value of the sink node is usually the lowest, and nodes that are farther from the sink usually have larger degree values. Therefore, in a multi-hop wireless network, data is transmitted from the nodes with higher degrees to lower degrees and finally to the sink node.

GBR [38] is a typical gradient routing protocol where a beacon message is flooded through the entire network and it keeps the number of hops in the flooding process. In this way, every node reveals its degree to the sink node, and a gradient network is formed. When a node wants to report information, it sends data to the neighboring node with the lowest degree value.

Collection Tree Protocol (CTP) [39], the de-facto standard collection protocol in TinyOS (a famous operating system for IoT), is another gradient-based routing protocol. In CTP, Expected Transmissions (ETX) is used to build the gradient network. Beacons are broadcasted in the network to update the degree value on each node, and thus maintaining a routing topology. To reduce the overhead of broadcasting beacons, an adaptive beaconing mechanism is proposed. When the network topology is relatively stable, fewer beacons are sent; when the network topology keeps changing, the adaptive beaconing mechanism is triggered to send more beacons.

The Working Group IETF Routing Over Low power and Lossy networks (ROLL) [40] concentrates on standardizing routing protocol for Low power and Lossy Networks (RPL). The fundamental building block of the RPL routing protocol [41] is the gradient routing. It supports traffic flows including multipoint-to-point, point-to-multipoint and point-to-point communications.

3 System models

This paper focuses on efficient and reliable communications among devices in a dense IoT system. Specifically, we study multichannel medium access under TDMA-based scheme. The objective is to efficiently allocate multiple wireless channels to devices because multichannel rendezvous is the fundamental problem in multichannel MAC protocol design. As discussed in the related work section, existing multichannel MAC protocols can be roughly categorized into two groups: static and dynamic multichannel MACs. The dynamic approaches can be further categorized into protocols with control channels (such as Y-MAC) and protocols without control channels (such as MuCHMAC). We are focusing on the dynamic multichannel protocols without control channel because this type of approach is more suitable for Internet of Things applications. In the dynamic multichannel MAC protocols, time division multiple access (TDMA) plays a critical role in channel management. Therefore, in this section we first introduce the TDMA-based network model, and then analyze the performance of dynamic multichannel protocols without control channel.

3.1 Network model

We consider the scenario where a great amount of devices are densely deployed in a target field. Each node is equipped with a radio that can be tuned into different channels. As shown in Fig. 1, the lifetime of the entire network is divided into several fixed-length duty-cycle periods. Each duty-cycle period is further divided into several time slots, e.g., four time slots in a duty-cycle period in Fig. 1.

With the TDMA timing optimization scheme, a device working in in low-power mode wakes up briefly for channel sensing in a randomly chosen time slot of a duty-cycle period. If traffic is present, it remains awake for this time slot to receive data. Otherwise, it goes back to the

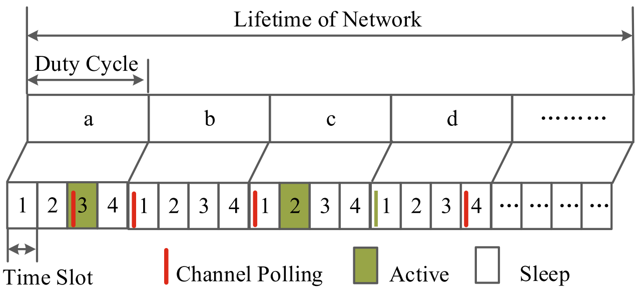


Fig. 1 The lifetime of a network is divided into several blocks and each block corresponds to a duty-cycle period. Each duty-cycle period is further divided into several time slots. In a duty-cycle period, a node randomly selects one time slot for channel polling

sleep model after channel polling, using the Clear Channel Assessment (CCA) technique. In the example shown in Fig. 1, a node continues polling the channel in the duty-cycle periods *a*, *b*, *c* and *d* at their *3rd*, *1st*, *1st*, and *4th* time slots, respectively. During the *3rd* time slot of duty-cycle period *a*, the node wakes up and performs CCA. The CCA result indicates the channel is busy, so it switches to the reception mode to receive data. For the other three cases, as CCA results indicate the channel is clear, i.e., no traffic, the node switches to sleep mode.

When a node intends to send packets, it competes the channel with others. If it wins, e.g., at the *2nd* time slot of duty-cycle period *c*, the node first sends short preambles to the intended receiver, and starts data transmission if an acknowledgement (ACK) packet is received from the receiver [42]. If it fails due to others transmitting packets, e.g., at the *1st* time slot of duty-cycle *d*, the node goes to sleep mode and attempt to retransmit the packets in the next duty-cycle period.

3.2 Analyzation of dynamic multichannel MAC

Existing dynamic multichannel protocols without control channel usually switch channels based on a pseudo-random sequence. In the following, we mathematically analyze its performance. Table 1 shows the nomenclature of variables used for analyzing the network performance of multichannel access protocols. Here, we consider the cases where there are n nodes (within the communication range) competing for m channels.

Case (I) We first consider a low data rate situation. $S(n,k)$ is used to denote the number of ways assigning k channels to n nodes, i.e., number of ways of partitioning a set of n elements into k nonempty subsets.

$$S(n,k) = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}C_k^j j^n \tag{1}$$

Table 1 Nomenclature of the multichannel access model

Notation	Description
n	Number of nodes within the communication range
m	Number of channels available on a node
$S(n,k)$	Number of assignments of k channels on n nodes
$P(m,n,k)$	Probability that exactly k nodes successfully access a channel given n nodes and m channels
$N(m,n)$	Expected upper bound of the rounds needed for n nodes finishing their transmission given m channels available
$E(m,n)$	Expected times of a node attempting to access the channel until it succeeds
$P(A_i)$	Probability that a node obtains a channel in the *ith* round

Let $P(m,n,k)$ denote the probability that exactly k nodes obtaining the channels given that n nodes competes for m channels. There are m^n possible ways of assigning m channels to n different nodes, and C_m^k possible choices of k channels from m channels. Therefore, we obtain the following equation

$$P(m,n,k) = \frac{S(n,k)C_m^k k!}{m^n} \tag{2}$$

We use $N(m,n-k)$ to denote the expected maximum number of rounds needed for the remaining $(n-k)$ nodes to finish their transmission. We know $N(m,0) = 0$ as the all nodes had finished their transmission. As a result, the expected upper bound of all nodes finishing their transmission can be calculated as

$$N(m,n) = 1 + \sum_{k=1}^{\min\{m,n\}} P(m,n,k)N(m,n-k) \tag{3}$$

On the other hand, let $E(m,n)$ denote the expected times that a node competes until it obtains the channel:

$$E(m,n) = \sum_{k=1}^{\min\{m,n\}} \frac{P(m,n,k)Q(m,n,k)}{n} \tag{4}$$

where $Q(m,n,k) = (n-k)*(E(m,n-k)+1)+k$. In the extreme case where $n=0$, we have $E(m,0)=0$.

Case (II) Next, we consider a high data rate situation. Let $N(m,n,a)$ denote the remaining expected number of turns when a new nodes (nodes that winning in contention and generating new packets after a while) had finished transmitting data.

For $(n-a)$ old nodes (nodes that holding the old packet due to fail in contention) and a new nodes, they compete for channels and suppose k of them obtain the chance to transmit, then probability that i old nodes obtain the chance to transmit is equal to $\frac{C_{n-a}^i C_{k-i}^a}{C_n^k}$.

Finally, We obtain Eq. 5. Initially, $N(m, n, n) = 0$ means all the n nodes had sent packets, and we need to compute $N(m, n, 0)$ as the expected upper bound of all n nodes finishing their transmission.

$$N(m, n, a) = 1 + \sum_{k=1}^{\min\{m,n\}} Q(k, a)P(m, n, k) \tag{5}$$

where,

$$Q(k, a) = \sum_{i=\max\{0,k-a\}}^{\min\{k,n-a\}} \frac{C_{n-a}^{i} C_{k-i}^{a}}{C_n^k} N(m, n, a+i) \tag{6}$$

Similarly, let $E(m, n, a)$ denote the expected times needed to wait for each remaining node after a nodes had finished transmitting data. Initially, $E(m, n, n) = 0$ means all the n nodes had finished transmitting data. We need to compute $E(m, n, 0)$ of Eq. 7:

$$E(m, n, a) = \sum_{k=1}^{\min\{m,n\}} R(k, a)P(m, n, k) \tag{7}$$

here,

$$R(k, a) = \sum_{i=\max\{0,k-a\}}^{\min\{k,n-a\}} \frac{C_{n-a}^{i} C_{k-i}^{a}}{C_n^k} * Q(m, n, a) \tag{8}$$

$$Q(m, n, a) = \frac{(E(m, n, a+i) + 1) * (n - a - i) + i}{n - a} \tag{9}$$

Table 2 gives the upper bound of random channel switching in low data rate and high data rate (corresponding to Eq. 3 and $N(m, n, 0)$ of Eq. 5). The more the contending nodes are, the severer the congestion become, especially in high traffic loads. For example, when there are 6 contending nodes with 3 available channels, the expected bound is 3 and 5 in case I and II, respectively. This means it needs three time periods for all the six nodes finishing packet transmission in low traffic loads and five time periods in high traffic loads.

Table 2 Expected bound of random channel switching

Nodes Channels	1	2	3	4	5	6	1	2	3	4	5	6
1	1	2	3	4	5	6	1	3	6	9	12	15
2	1	2	3	3	4	4	1	2	3	5	6	8
3	1	2	2	3	3	**3**	1	2	3	3	4	**5**
4	1	2	2	2	3	3	1	2	2	3	4	4
5	1	2	2	2	3	3	1	2	2	3	3	4
6	1	2	2	2	2	3	1	2	2	2	3	3
	Case I						Case II					

4 Cross-layer cooperative multichannel medium access (CCMMA)

Based on the previous analysis, we find random channel access scheme has some limitations in effectively assigning multichannel to different nodes in a network. We address this issue by proposing a cooperative multichannel medium access (CMMA) scheme. When the number of nodes increases or the available wireless channels becomes less, it is possible for multiple nodes to choose the same channel in CMMA. To achieve this goal, we adopt the TDMA timing optimization technique and leverages the routing information to efficiently assign channel polling time slot to nodes. With the innovative cross layer cooperative multichannel medium access (CCMMA) scheme, many key metrics of QoS (Quality of Service) are significantly improved, such as network congestion, throughput, fairness, delivery rate, delay, and jitter.

4.1 Cooperative multichannel access

To take full advantage of the scarce spectrum resource, we present a cooperative multichannel access mechanism, where neighboring nodes select listening channel in each duty-cycle period according to Latin rectangle. Let $\mathbf{Z}_n$ presents a set consisting of n elements. A $k \times n$ Latin rectangle is a $k \times n$ matrix ($k \leq n$) that chooses entries from the set $\mathbf{Z}_n$, and any element occurs no more than once in each row and column [43]. For example

$$L_{5\times8} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 1 \\ 3 & 4 & 5 & 6 & 7 & 8 & 1 & 2 \\ 4 & 5 & 6 & 7 & 8 & 1 & 2 & 3 \\ 5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 \end{pmatrix}$$

is a Latin rectangle with 5 rows and 8 columns. Every row contains all elements of set $\mathbf{Z}_n$, but no column contains a duplicate element. This feature is quite attractive for medium access using multiple channels. Let we consider columns in Latin rectangle L as time periods in a frame, and regard the $\mathbf{Z}_n$ as the set of available channels in the network. If each node selects one of rows in L as its channel switching sequence, no one would listen a repeat channel in the same duty-cycle period. Therefore, the spectrum utilization reaches maximum. Analytical formulations in aforementioned two traffic situations are presented in the following.

Case (I) The expected upper bound and the expected number of a node wins the channel can be calculated by Eqs. 10 and 11 respectively.

$$N(m, n) = \left\lceil \frac{n}{m} \right\rceil \tag{10}$$

$$E(m,n) = \frac{\sum_{i=1}^{\lceil \frac{n}{m} \rceil} i}{\lceil \frac{n}{m} \rceil} \tag{11}$$

Case (II) As m channels are assign to n nodes, hence at most $\lceil \frac{n}{m} \rceil$ nodes compete for the same channel, then we analyze the expected number of turns needed for $\lceil \frac{n}{m} \rceil$ nodes to finish transmitting data.

Let $F = \lceil \frac{n}{m} \rceil$, and then $N(a, F-a)$ denote the expected number of turns needed to make the remaining a nodes finish transmitting data after $(F-a)$ nodes had already obtained the chances to transmit data. Then we have the recurrence relationship as follows:

$$\begin{aligned} N(a, F-a) &= 1 + \frac{F-a}{F} N(a, F-a) \\ &\quad + \frac{a}{F} N(a-1, F-(a-1)) \end{aligned} \tag{12}$$

At last we obtain the recurrence relationship as follows:

$$N(a, F-a) = \begin{cases} 0 & a = 0 \\ \frac{F}{a} + N(a-1, F-a+1) & a \le F \end{cases} \tag{13}$$

We need to compute $N(F, 0)$ and the result is

$$N(F, 0) = F \sum_{i=1}^{F} \frac{1}{i} = \left\lceil \frac{n}{m} \right\rceil \sum_{i=1}^{\lceil \frac{n}{m} \rceil} \frac{1}{i} \tag{14}$$

With respect to the expected counts that a node competes for channel until it finishes packet transmission, the result can be obtained through Eq. 15:

$$E(m,n) = \begin{cases} 1 & n \le m \\ \sum_{i=1}^{+\infty} P(A_i | \prod_{j=1}^{i-1} \bar{A}_j) \times i & n > m \end{cases} \tag{15}$$

where, $P(A_i)$ represents a node wins the channel at ith contention. Its value can be obtained by Eq. 16:

$$P(A_i) = \frac{\lceil \frac{n}{m} \rceil}{n} \times \frac{1}{\lceil \frac{n}{m} \rceil} + \frac{n - \lceil \frac{n}{m} \rceil}{n} \times \frac{1}{\lfloor \frac{n}{m} \rfloor} \tag{16}$$

Table 3 Expected bound of cooperative channel switching

Nodes / Channels	1	2	3	4	5	6	1	2	3	4	5	6
1	1	2	3	4	5	6	1	3	6	9	12	15
2	1	1	2	2	3	3	1	1	3	3	6	6
3	1	1	1	2	2	2	1	1	1	3	3	3
4	1	1	1	1	2	2	1	1	1	1	3	3
5	1	1	1	1	1	2	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1
	Case I						Case II					

Table 3 gives the expected upper bound of cooperative channel access in low data rate and high data rate (corresponding to Eqs. 10 and 14). In either case, all nodes can finish their packet transmission at the first duty-cycle period as long as the number of available channels is no less than the number of contending nodes.

4.2 Potential benefit of TDMA timing optimization

The TDMA timing optimization technique allows nodes to randomly choose time slots to transmit and/or receive data. Therefore, channel contention is further reduced when node density is high. However, it does not consider that the routing information is critical in determining time slot assignments to further improve QoS. In the following, we analyze the limitations of exiting TDMA timing optimization techniques, and then propose a solution in the next subsection.

Let $P(n)$ denote the probability that a packet can be delivered at least n-hop within a single duty-cycle period. The number of slots divided in a duty-cycle period is denoted as N_s. Obviously, 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,

$$P(n) = \begin{cases} 1 & n = 1 \\ 0 & n > N_s \end{cases} \tag{17}$$

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 duty-cycle period. Figure 2 show an 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 $3th$ slot respectively, a data can be transmitted three hops from node D to node A in one

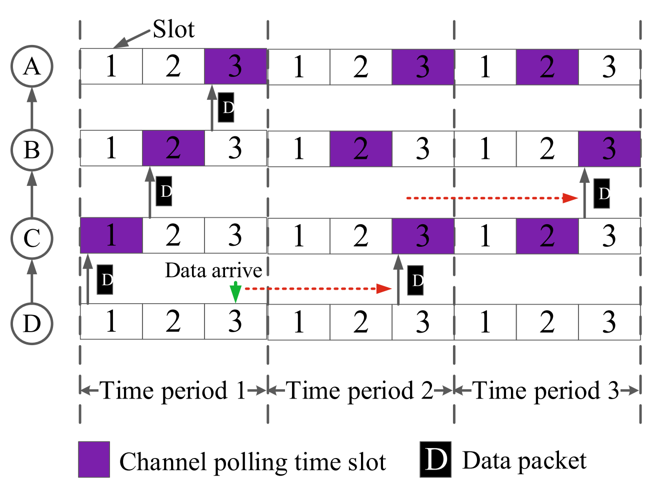


Fig. 2 An example of how nodes poll channels in a 3-hop linear network topology

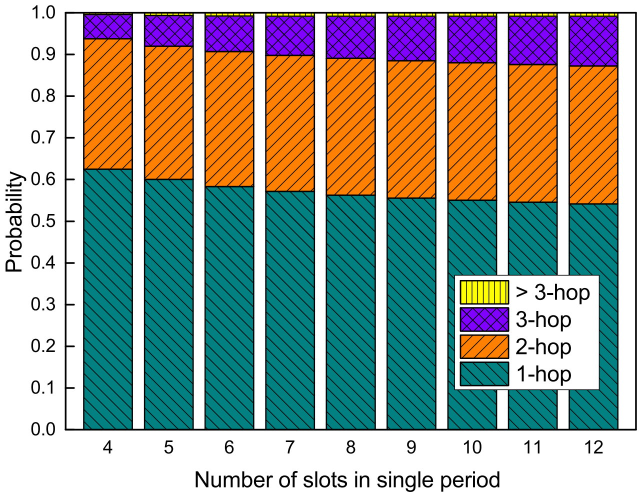


Fig. 3 Multi-hop transmission probability in a single duty-cycle period with the random TDMA timing optimization technique

period. If node C wakes up at the $3th$ slot, node D will transmit data at that time, which cannot be sent to the next node during the same duty-cycle period. Therefore, we can derive the following equation:

$$P(n) = \begin{cases} \dfrac{\sum_{i=1}^{N_s-1} i}{{N_s}^2} & n = 2 \\ \dfrac{\sum_{j=1}^{N_s-n+1} \sum_{i=1}^{j} i}{{N_s}^n} & 2 < n \leq N_s \end{cases} \tag{18}$$

where ${N_s}^n$ is the number of whole possible outcomes that all nodes on a multi-hop path pick randomly out of slots. The numerator is the number of permutations that a packet can be delivered at least n-hop within a single duty-cycle period. Therefore, the probability of a packet forwarded only n-hop can be captured by $P(n) - P(n+1)$. Figure 3 provides information on random TDMA schedule's capability of multi-hop delivery within a single duty-cycle period. Overall, the probability varies slightly. More than 50 % 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 duty-cycle period is divided into twelve slots.

4.3 Routing-enhanced mechanism

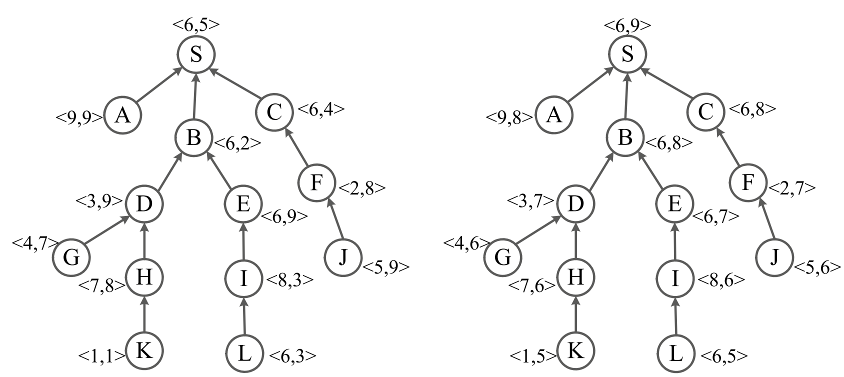
We exploit a cross-layer design to mitigate the delay introduced by duty cycling techniques as well as maximize parallel transmissions. Every node calculates channel polling slot assignment based on its hop distance to the sink node. The detail process is presented with the following example. In this 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 slot assignment eliminates vertical interference.

Figure 4 shows an example of convergecast 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 slot of this node. Instead of randomly choosing one of slots, the channel polling slot of each node in our approach is calculated by the following equation:

$$S_j = N_s - \ell \bmod N_s \tag{19}$$

where N_s is the number of slots divided in a duty-cycle period as mentioned above, ℓ is the hop distance to the sink, which can be easily obtained from gradient-based routing protocols. The IETF ROLL [40] has based its future standard on gradient-based routing, since it is easy to implement for real-world deployments and suitable for applications in low power and lossy networks [34]. 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

Fig. 4 An example of convergecast traffic in a multi-hop network



(a) Random TDMA timing optimization

(b) Routing-enhanced TDMA timing optimization 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 random working schedule, it wakes up to send the packet to node H at slot 8, and this packet can be forwarded to node D at slot 9. However, node D has to wait for its next upstream node B to wake up to receive the packet in the next duty-cycle period. Finally, the packet arrives on sink node S at slot 5 of next duty-cycle period. In our approach, all node along the data forwarding path wake up sequentially from slot number 6 to 9. It is easy to find that data packets can be forwarded multiple hops within a single duty-cycle period 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. 4b, nodes of the same depth to the sink node (e.g., 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.

Unfortunately, when there are not enough available channels, two nodes (node B and node C in Fig. 4b) maybe wake up to receive packet at the same time on the same channel. It is also possible that two nodes (node G and node H in Fig. 4b) want to transmit packet to the same receiver at the same time. So collision avoidance and retransmission must be considered in such cases. In the proposed approach, nodes with pending data to send wake up and perform carrier sense at a random point within backoff-window. The node that detects channel idle sends small preamble messages to occupy wireless medium. Upon receiving the preamble, the intended receiver sends ACK frame for acknowledgement and starts to exchange data. If a busy channel is detected by the node, it aborts transmission until the next duty-cycle period and goes back to sleep for high energy efficiency.

5 Performance evaluation

The CCMMA is implemented in the *ns-2* network simulator, and Table 4 summarizes the configuration parameters used in the simulations. The parameters for radios on nodes are based upon the datasheet of CC2420 radio [44]. Transmission range and carrier sensing range are configured to simulate several state-of-art sensor nodes [45]. These values are also the typical setting used in *ns-2*. The duration of a duty-cycle period is set to be one second that is further divided into 5 (or 10) time slots. We change the number of channels from 5 to 10 to evaluate how it affects network performance. Because various network topologies have different impacts on network performance, we start our evaluations from a single-hop scenario and then extend it to bipartite and multi-hop topologies. In addition to network delay, we also evaluate the throughput, the queue length, and queue overflow of CCMMA and existing solutions.

Table 4 Simulation parameters

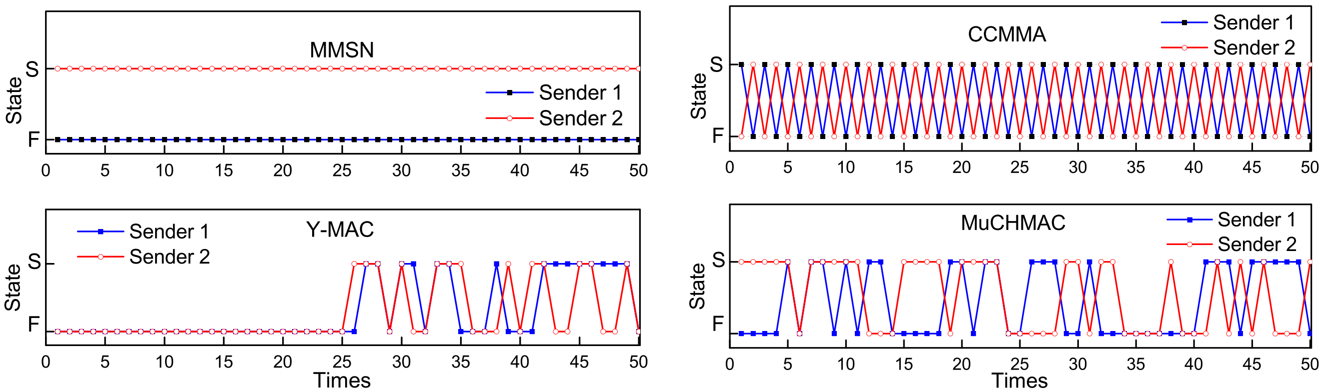
Parameter	Value
Tx range	250 m
Carrier sensing range	550 m
Bandwidth	250 Kbps
Duty-cycle period	1 second
Slots divided in single period	5 or 10
Number of available channels	5 or 10
Size of data packet	251 Bytes
Queue length	50 or 200 pkts
Backoff window	[31,255]

5.1 Single-hop topology

Due to the limited amount of spectrum resources, many wireless technologies use the same band. For example, ZigBee, Bluetooth and WiFi technologies all operate in the 2.4 GHz ISM band. Therefore, IoT devices are susceptible to external interference, a major cause of QoS (Quality of Service) degradation. Considering a single-hop topology with an external interfering source, we compare the performance of CCMMA and several existing solutions, e.g., MMSN, Y-MAC and MuCHMAC.

In this scenario, two senders and two receivers are deployed in a small area where *Sender 1* transmits packets to *Receiver 1*, and *Sender 2* transmits packets to *Receiver 2*. There are two channels, *Channel_0* and *Channel_1*, available in MMSN, MuCHMAC and CCMMA. In Y-MAC, there are on control channel, *Channel_0*, and two data channels, *Channel_1* and *Channel_2*. The external wireless interference source continues transmitting data on *Channel_0*. In the simulations, we record the transmission states of *Sender 1* and *Sender 2* in Fig. 5. The state Success (S) denotes that a sender successfully transmits packets, and state Failure (F) means the sender cannot transmit data.

Figure 5 shows the transmission states of *Sender1* and *Sender2* that run MMSN, Y-MAC, CCMMA and MuCHMAC protocols. With the MMSN protocol, because static channel assignment is adopted, we see *Channel_0* is assigned to *Sender 1* and *Channel_1* is assigned to *Sender 2*. In this case, only *Sender 2* works and *Sender 1* cannot transmit any packets. With the Y-MAC protocol, nodes need to



(a) (b)

Fig. 5 Transmission states of senders in a single-hop topology under external wireless interferences. State S and F denote that a sender successfully or fails to transmit data, respectively

negotiate on the control channel before any data transmission. Because the control channel *Channel_0* is occupied by the external interference source both senders are in F states from time 0 to 25. When the external interference is turned off at time 26, both *Sender 1* and *Sender 2* could transmit packets if *Channel_2* is assigned for transmissions.

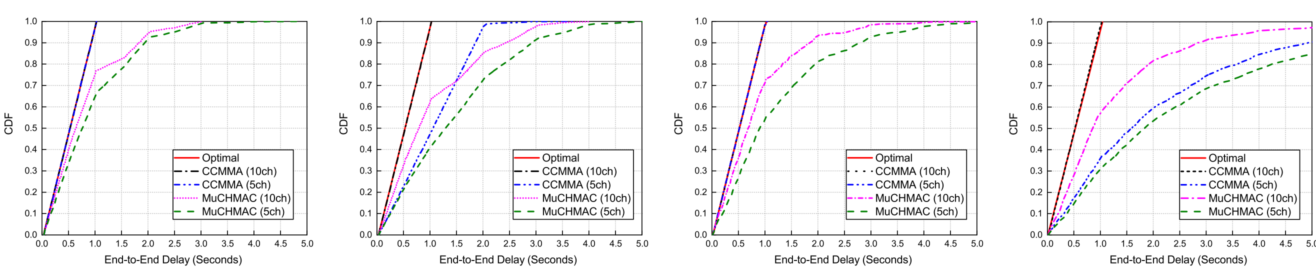
In CCMMA, both senders could send packets successfully as they always use different channels for transmissions. If a sender detects *Channel_0* is occupied, it will switch to *Channel_1* next time. Similar to CCMMA, senders with MuCHMAC dynamically change their communicating channels. However, random channel selection cannot efficiently avoid wireless interference, especially, in the situation of less channels available. For example, *Sender 1* fails 4 times and finally transmits successfully at 5th time in MuCHMAC. Among all existing solutions, we find MuCHMAC gives the best network performance. Therefore, we only compare the performance of CCMMA to MuCHMAC and simply omit the other protocols.

5.2 Bipartite topology

In a bipartite network, nodes are divided into two sets such that a node in the first set only connects to a node in the second set. This network topology is more complicated than the single-hop topology because parallel transmissions from several pairs of nodes may interfere with each other. In the simulations, we change the number of source nodes to simulate various network densities. Each source node generates a total of 200 packets. All source nodes are closely deployed and send packets to corresponding receivers simultaneously, i.e., they are within the interference range of each other. To focus on the channel assignment efficiency, we disable TDMA timing optimization mechanism on both CCMMA and MuCHMAC in low data generation rate. When data rate increases, media would be occupied constantly. So the performance are similar with or without timing division mechanism.

5.2.1 End-to-end delay

Figure 6a and b show the Cumulative Distribution Function (CDF) of end-to-end delays of CCMMA and MuCHMAC protocols in a low data rate situation. The optimal result is obtained from the simplest bipartite topology, i.e., one source transmits data to one receiver. Since the duty-cycle period is 1 second, the optimal end-to-end delays are distributed uniformly between 0 and 1 seconds. In a moderate



(a) Low data rate in a moderate dense network (b) Low data rate in a dense network (c) High data rate in a moderate dense network (d) High data rate in a dense network

Fig. 6 End-to-end delays of the CCMMA and MuCHMAC and the optimal result under various node densities and data rates dense network where 5 nodes are within the transmission range, there are 66.7 % packets delivered to the receivers by MuCHMAC within one duty-cycle period, given 5 channels available. The maximum end-to-end delay of MuCHMAC is about 4.87 seconds. On the other hand, the end-to-end delay achieved by CCMMA with 5 and 10 channels is very close to the optimal values. The reason is that CCMMA makes use of most spectrum resources available in the network, thus reaching maximum spectrum utilization.

The CDF of end-to-end delays in high data rate situations are shown in Fig. 6c and d. In a dense network where 10 contending nodes are within the transmission range, only 41 % and 63 % packets are transmitted during the first duty-cycle period by MuCHMAC with 5 channels and 10 channels available, respectively. However, all packets in CCMMA (10 channels) are delivered within the first duty-cycle period. Similar to the low data rate situation, CCMMA achieves much smaller end-to-end delay in both moderate dense and dense networks than MuCHMAC.

5.2.2 Throughput

Network throughput is a critical metric if applications feature burst traffic loads. In the following, we evaluate network throughput achieved by CCMMA and MuCHMAC. A moderate dense bipartite network topology is used here with 5 transmitting nodes and 5 channels. The queue length on each node is set to be 200 to obtain the maximum throughput. Figure 7 show the throughput of CCMMA, MuCHMAC and the optimal result. Because the bandwidth is 250 Kbps and the size of a packet is 251 *Bytes*, the maximum throughput is 124.5 packets/second. The actual optimal value is 107 packets/second because additional preambles are added to each packet. We see that the throughput of both CCMMA and MuCHMAC gradually increases as the data rate increases. After reaching the highest point, the throughput goes down due to network congestions. We observe that the throughput achieved by CCMMA is almost identical to the optimal one. However, the throughput achieved by MuCHMAC is lower and decreases quickly when more and more packets are injected into the network. This is because the same channel is chosen by multiple nodes in MuCHMAC, and only one node wins the competition and transmit packets.

5.2.3 Fairness

Fairness is another important metric in wireless networks, especially when smart devices are interconnecting under the TCP/IP network architecture. We use the well-known Jain's fairness index to capture the fairness achieved by CCMMA and MuCHMAC. The network setting is same as previous evaluations, where five nodes transmit packets to their destinations with five channels available. The parameter n in Eq. 20 is 5, and x_i is the throughput of the ith node.

$$f(x_1, x_2, ..., x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} {x_i}^2} \tag{20}$$

Figure 8 shows the result from 50 runs. The fairness index of CCMMA is always 1 (the best case). This is because the cooperative multichannel access enables all nodes to receive the same allocation. However, the fairness index of MuCHMAC fluctuates between $\frac{2}{5}$ and $\frac{4}{5}$.

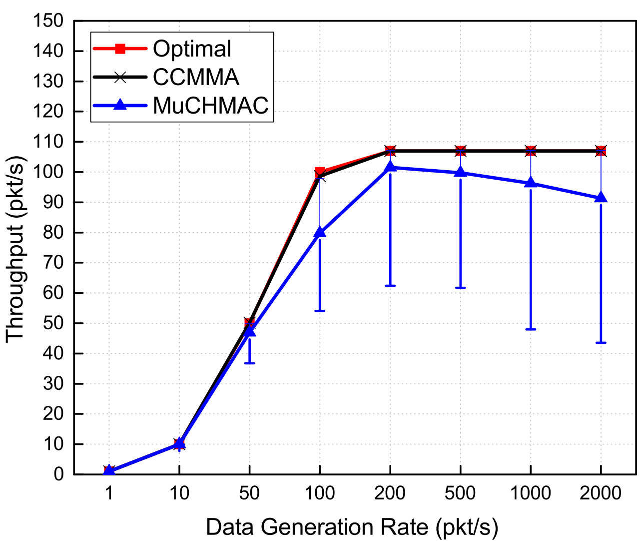


Fig. 7 Throughput of CCMMA and MuCHMAC and the optimal results in a bipartite network topology

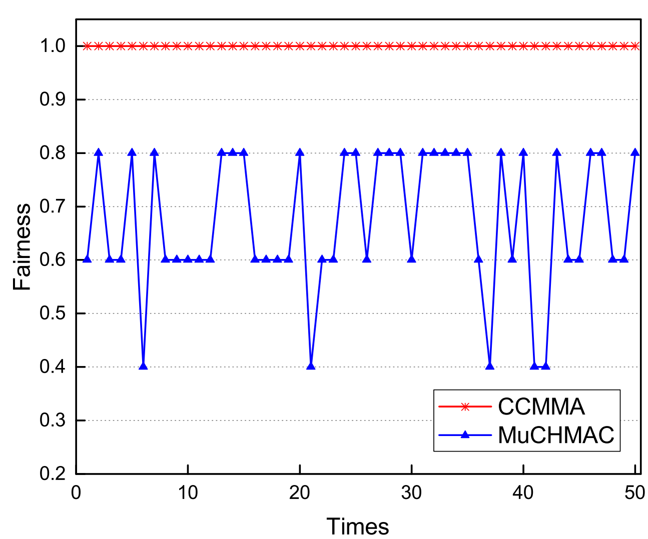


Fig. 8 The performance of fairness by CCMMA and MuCHMAC from 50 runs

Furthermore, the fairness index is only 0.4 among 10 % of the network's lifetime.

5.3 Multihop topology

In the last group of simulations, we evaluate the network performance of CCMMA and MuCHMAC in a multhop topology. In the simulations, we choose a chain topology where each node is 150 meters away from its upstream (and downstream) neighboring node and the last node is considered the sink node. We assume a Constant Bit Rate (CBR) traffic flows from the source to the sink nodes. The length of the topology varies from 5 to 20 hops.

5.3.1 End-to-end delay

Figure 9 shows the end-to-end delays achieved by MuCHMAC and CCMMA with various lengths of routing paths. Both the delays increase almost linearly with the number of hops. However, MuCHMAC has a much bigger rate of increase than CCMMA. When the path length is 20 hops, the average and maximum delay of MuCHMAC with 10 time slots are 11.64 and 17.41 seconds, respectively. The values for CCMMA are 2.42 and 2.94 seconds, respectively. We conclude that CCMMA reduces delay by 79 % compared to MuCHMAC. In MuCHMAC, a packet is delayed (on average) for a half of duty-cycle period in each hop, however, CCMMA is able to transmit a packet multiple hops away in a single duty-cycle period. If a duty-cycle period is divided into 5 time slots, the overall delay of CCMMA goes up a bit but it is still much smaller than MuCHMAC. The results also implies that more slots in a duty-cycle period can better help CCMMA to improve its delay performance, compared to MuCHMAC.

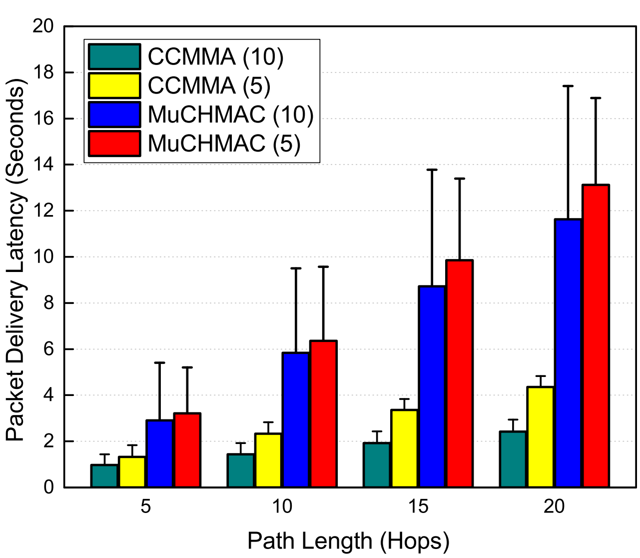


Fig. 9 End-to-end delays of CCMMA and MuCHMAC with various number of channels and routing path lengths

5.3.2 Jitter

Figure 10 depicts the jitter performance of CCMMA and MuCHMAC with different CBRs. When the interval of a CBR flow is set to be 10 seconds, the jitter of MuCHMAC ranges within [−5.3, 6.3] seconds. While the jitter range of CCMMA is only [−0.1, 0.9] seconds. When the interval decreases to 0.1 second, the jitter of MuCHMAC degrades to [−6.59, 18.1] seconds, while CCMMA's jitters are within [−0.08, 0.8] seconds. In MuCHMAC, because nodes wake up randomly and independently, large waiting time is expected on each hop, i.e., longer the routing path, larger the jitter. On the other hand, nodes in CCMMA wake up in a ripple fashion, so the packets do not suffer long waiting time at each hop.

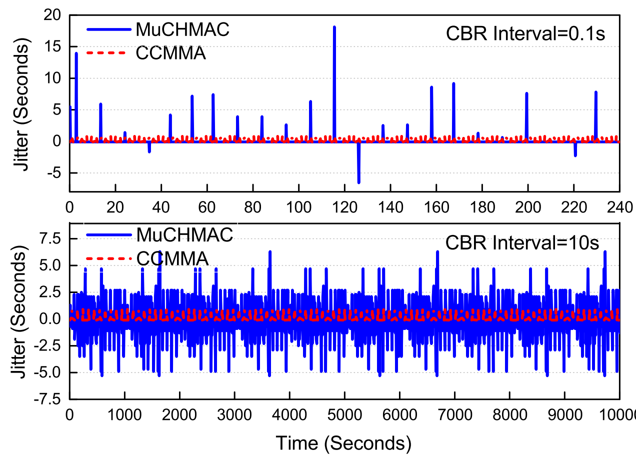


Fig. 10 Jitter performance of CCMMA and MuCHMAC in a 20-hop chain network topology with different CBRs

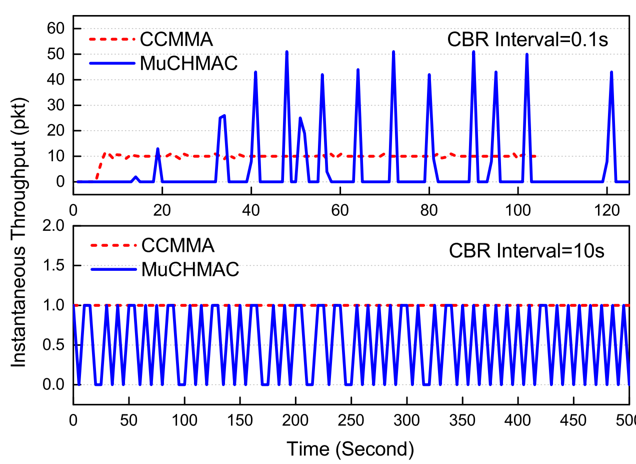


Fig. 11 Instantaneous throughput of CCMMA and MuCHMAC in a 20-hop chain network topology with different CBRs

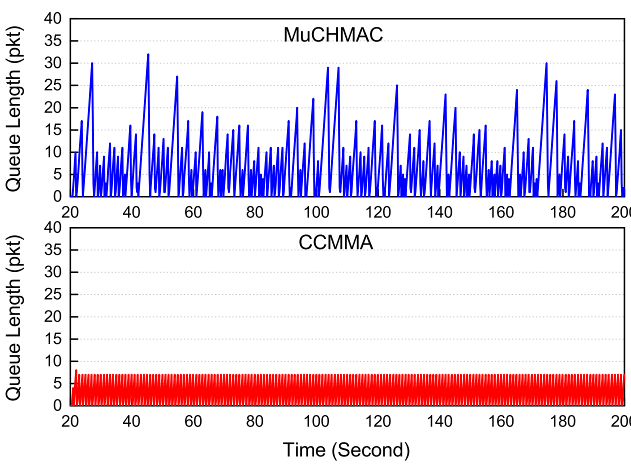
5.3.3 Instantaneous throughput

The instantaneous throughputs of CCMMA and MuCHMAC under low and high CBRs are shown in Fig. 11. We use the throughput achieved on the sink node in every 5-second period to represent the instantaneous throughput. No matter what CBR interval is, 10 or 0.1 seconds, CCMMA has steady instantaneous throughput. However, the instantaneous throughput of MuCHMAC fluctuates wildly. The smaller the CBR interval, the larger the fluctuation.

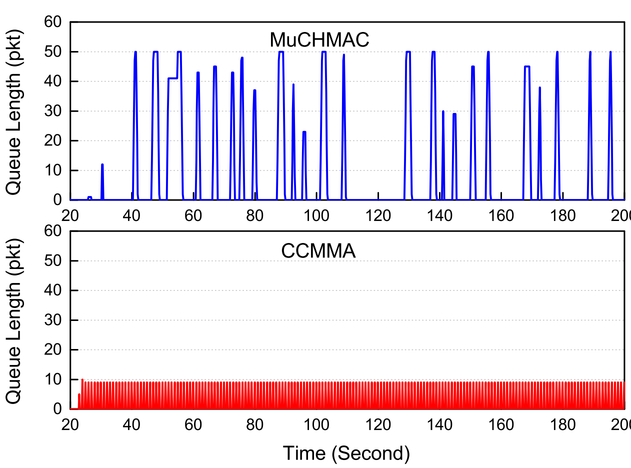
5.3.4 Queue length

At last, we discuss the average queue length on nodes in CCMMA and MuCHMAC. The queue size on each node is set to be 50 packets. We record the queue length on nodes when the CBR interval is 0.1 second. The queue lengths on the source node and the $10th$-hop node are shown in Fig. 12a and b, respectively. When the CBR interval is 0.1 second. 10 packets are generated in every duty-cycle period. The queue length on the source node with CCMMA is always less than 10. While with MuCHMAC, the queue length is up to 33. This is because the $2nd$-hop node wake up randomly and when it is in sleep mode, the source node has to put the packet into its queue. Furthermore, when the $2nd$-hop node sends packets to the $3rd$-hop node, it cannot receive the packets from the source node, due to wireless interference. As a result, the source node's queue length could be very large.

For the $10th$-hop node, if MuCHMAC is used, its queue length becomes 50 several times, leading to queue overflows. Although a total of 2000 packets are generated from the source node, only 1155 packets are received by the sink node, which means most packet are dropped due to queue overflow occurred in MuCHMAC. On the other hand, with the CCMMA protocol, the queue length of the $10th$-hop node is always less than 10, and all 2000 packets are successfully delivered to the sink node. This is because all forwarding nodes can send packets with a short waiting time.



(a) Queue length on the source node



(b) Queue length on the $10th$-hop forwarding node

Fig. 12 Queue lengths on the **a** source node and **b** $10th$-hop nodes in a 20-hop chain network topology when the CBR interval is 0.1 second

6 Conclusion

This paper presented the design and evaluation of an innovative cross-layer multichannel media access protocol for IoT. The protocol is able to fully utilize the scarce spectrum resource, leveraging cooperative multichannel access based on Latin rectangle. The routing-enhanced mechanism achieves low packet delivery delay, as well as small jitter and queue length, by enabling forwarding nodes to wake up sequentially.

We conducted intensive evaluations of the proposed protocol in *ns-2*. CCMMA is demonstrated to reduce communication congestion up to 41 % in dense network with high data rate, and significantly improves throughput in high traffic loads. It is also demonstrated to reduce average delivery delay by 79 % in a 20-hop multi-hop topology. In addition, CCMMA achieves better jitter performance that is only 17 % and 5 % of the jitters generated in MuCHMAC in low data rate and high data rate situations, respectively. CCMMA also achieves much smaller queue length at both the source node and forwarding nodes.

Acknowledgments The authors would like to thank anonymous reviewers for their valuable comments and constructive suggestions that significantly helped us to improve the quality of this paper. This work was partly supported by the Natural Science Foundation of China (Grant No.61550110244) and Natural Science Foundation of Jiangsu (Grant No.BK20161147).

References

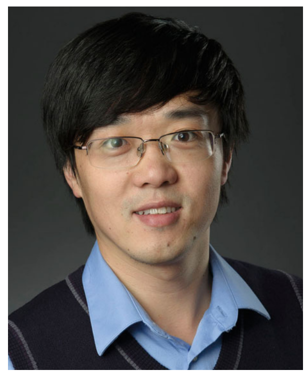
1. Stankovic J (2014) Research directions for the internet of things. IEEE Internet of Things Journal 1(1):3–9
2. Chen S, Xu H, Liu D, Hu B, Wang H (2014) A vision of iot: applications, challenges, and opportunities with China perspective. IEEE Internet of Things Journal 1(4):349–359
3. Joh H, Yang I, Ryoo I (2016) The internet of everything based on energy efficient p2p transmission technology with bluetooth low energy. Peer-to-Peer Networking and Applications 9(3):520–528
4. Deng X, He L, Li X, Liu Q, Cai L, Chen Z (2016) A reliable qos-aware routing scheme for neighbor area network in smart grid. Peer-to-Peer Networking and Applications 9(4):616–627
5. Xia Z, Wang X, Sun X, Wang Q (2016) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE Trans Parallel Distrib Syst 27(2):340–352
6. Ren Y-J, Shen J, Wang J, Han J, Lee S-Y (2015) Mutual verifiable provable data auditing in public cloud storage. J Intell Technol 16(2):317–323
7. Zhangjie F, Xingming S, Qi L, Lu Z, Jiangang S (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. IEICE Trans Commun 98(1):190–200
8. Fu Z, Wu X, Guan C, Sun X, Ren K (2016) Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Trans Inf Forensics Secur 11(12):2706–2716
9. Guo P, Wang J, Geng XH, Kim CS, Kim J-U (2014) A variable threshold-value authentication architecture for wireless mesh networks. J Intell Technol 15(6):929–935
10. Hnat TW, Srinivasan V, Lu J, Sookoor TI, Dawson R, Stankovic J, Whitehouse K (2011) The hitchhiker's guide to successful residential sensing deployments. In: Proceedings of the 9th ACM conference on embedded networked sensor systems, ser. Sensys '11, pp 232–245
11. Jin M, Jia R, Kang Z, Konstantakopoulos IC, Spanos CJ (2014) Presencesense: Zero-training algorithm for individual presence detection based on power monitoring. In: Proceedings of the 1st ACM conference on embedded systems for energy-efficient buildings, ser. Buildsys '14, pp 1–10
12. Zhang T, Chowdhery A, Bahl PV, Jamieson K, Banerjee S (2015) The design and implementation of a wireless video surveillance system. In: Proceedings of the 21st annual international conference on mobile computing and networking, ser. Mobicom '15, pp 426–438
13. Abari O, Vasisht D, Katabi D, Chandrakasan A (2015) Caraoke: an e-toll transponder network for smart cities. In: Proceedings of the 2015 ACM conference on special interest group on data communication, ser. SIGCOMM '15, pp 297–310
14. Zhu T, Cao Z, Gong W, He Y, Liu Y (2013) Illuminations and the revelations: lessons learned from greenorbs project development. SIGMOBILE Mob Comput Commun Rev 17(4):42–46
15. Shen J, Tan H, Wang J, Wang J, Lee S (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. J Intell Technol 16(1):170
16. Wang Z, Song H, Watkins DW, Ong KG, Xue P, Yang Q, Shi X (2015) Cyber-physical systems for water sustainability: challenges and opportunities. IEEE Commun Mag 53(5):216–222
17. Given DD, Cochran ES, Heaton T, Hauksson E, Allen R, Hellweg P, Vidale J, Bodin P (2014) Technical implementation plan for the shakealert production system: an earthquake early warning system for the west coast of the United States. US Geological Survey, Tech. Rep.
18. Yang Q, Lim A, Casey K, Neelisetti R-K (2009) An enhanced cpa algorithm for real-time target tracking in wireless sensor networks. Int J Distrib Sens Netw 5(5):619–643
19. Yang Q, Lim A, Casey K, Neelisetri R-K (2009) An empirical study on real-time target tracking with enhanced cpa algorithm in wireless sensor networks. Adhoc & Sensor Wireless Networks 7
20. Zhang Y, Sun X, Wang B (2016) Efficient algorithm for k-barrier coverage based on integer linear programming. China Communications 13(7):16–23
21. Jamieson K, Balakrishnan H, Tay Y (2006) Sift: a mac protocol for event-driven wireless sensor networks. In: Wireless sensor networks. Springer, pp 260–275
22. Miskowicz M (2009) Average channel utilization of csma with geometric distribution under varying workload. IEEE Trans Ind Inf 5(2):123–131
23. Tang C, Song L, Balasubramani J, Wu S, Biaz S, Yang Q, Wang H (2014) Comparative investigation on csma/ca-based opportunistic random access for internet of things. IEEE Internet of Things Journal 1(2):171–179
24. Lu G, Sadagopan N, Krishnamachari B, Goel A (2005) Delay efficient sleep scheduling in wireless sensor networks. In: INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies. Proceedings IEEE, vol 4, pp 2470–2481
25. Lou C, Zhuang W (2016) Energy-efficient routing over coordinated sleep scheduling in wireless ad hoc networks. Peer-to-Peer Networking and Applications 9(2):384–396
26. Huang P, Xiao L, Soltani S, Mutka M, Xi N (2013) The evolution of mac protocols in wireless sensor networks: a survey. IEEE Commun Surv Tutorials 15(1):101–120
27. Almotairi KH, Shen XS (2015) A distributed multi-channel mac protocol for ad hoc wireless networks. IEEE Trans Mob Comput 14(1):1–13
28. Zhou G, Huang C, Yan T, He T, Stankovic J, Abdelzaher T (2006) Mmsn: Multi-frequency media access control for wireless sensor networks. In: INFOCOM 2006. 25th IEEE international conference on computer communications. Proceedings, pp 1–13
29. Kim Y, Shin H, Cha H (2008) Y-mac: an energy-efficient multi-channel mac protocol for dense wireless sensor networks. In: International conference on information processing in sensor networks, 2008. IPSN '08, pp 53–63
30. Borms J, Steenhaut K, Lemmens B (2010) Low-overhead dynamic multi-channel mac for wireless sensor networks. In: Wireless sensor networks. Springer, pp 81–96
31. Tang L, Sun Y, Gurewitz O, Johnson DB (2011) 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, pp 23:1–23:11
32. Al Nahas B, Duquennoy S, Iyer V, Voigt T (2014) Low-power listening goes multi-channel. In: IEEE international conference on distributed computing in sensor systems (DCOSS), 2014, pp 2–9
33. Dunkels A (2011) The ContikiMAC Radio Duty Cycling Protocol, SICS Technical Report T2011: 13
34. Watteyne T, Molinaro A, Richichi M, Dohler M (2011) From manet to ietf roll standardization: a paradigm shift in wsn routing protocols. IEEE Commun Surv Tutorials 13(4):688–707
35. Pantazis NA, Nikolidakis SA, Vergados DD (2013) Energy-efficient routing protocols in wireless sensor networks: a survey. IEEE Commun Surv Tutorials 15(2):551–591
36. Yu S, Zhang B, Li C, Mouftah HT (2014) Routing protocols for wireless sensor networks with mobile sinks: a survey. IEEE Commun Mag 52(7):150–157

37. Xie S, Wang Y (2014) Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. Wirel Pers Commun 78(1):231–246
38. Schurgers C, Srivastava MB (2001) Energy efficient routing in wireless sensor networks. In: Military communications conference, 2001. MILCOM 2001. Communications for network-centric operations: creating the information force. IEEE, vol 1. IEEE, pp 357–361
39. Gnawali O, Fonseca R, Jamieson K, Moss D, Levis P (2009) Collection tree protocol. In: Proceedings of the 7th ACM conference on embedded networked sensor systems. ACM, pp 1–14
40. Ietf roll, https://datatracker.ietf.org/wg/roll/charter/
41. Winter ET, Thubert EP, Brandt A, Hui J, Kelsey R, Levis P, Pister K, Struik R, Vasseur J, Alexander R (2012) Rpl: Ipv6 routing protocol for low-power and lossy networks. Internet Requests for Comment 6550(5):853–861
42. Buettner M, Yee GV, Anderson E, Han R (2006) X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks
43. Doyle PG (2007) The number of latin rectangles. arXiv:math/0703896
44. Texas instrument inc. cc2420 data sheet. http://www.ti.com
45. Sun Y, Gurewitz O, Johnson DB (2008) Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In: Proceedings of the 6th ACM conference on embedded network sensor systems. ACM, pp 1–14

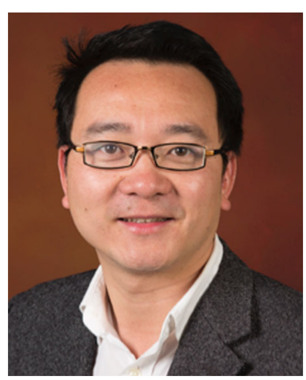
Ye Liu received the M.S. degree from Southeast University, Nanjing, China in 2013. He was a visiting scholar at Montana State University, Bozeman, USA from October 2014 to October 2015. He is currently working towards the Ph.D. degree in the School of Electronic Science and Engineering, Southeast University, China. His current research interests include wireless sensor networks, radio frequency identification (RFID) and energy harvesting systems.

Chenglin Fan is a current Ph.D student of computer science department in Montana State University. He received his M.S. degree in computer science from Central South University, Changsha, in 2011. His research interests are in algorithm design and analysis, wireless sensor networks.

Hao Liu received the B.S. degree from Xi'an Jiaotong University, China in 1996 and the Ph.D. degree from Southeast University, China in 2011. Now he is an associate professor in Southeast University. His research interests include wireless sensor networks, signal processing, communication circuit, DPM, SoC system.



Qing Yang is a RightNow Technologies Assistant Professor in the Department of Computer Science at Montana State University. He received B.S. and M.S. degrees in Computer Science from Nankai University and Harbin Institute of Technology, China, in 2003 and 2005, respectively. He received his Ph.D degree in Computer Science from Auburn University in 2011. His research interests include Online Social Network, Trust Management, Vehicular Ad-Hoc Networks (VANETs), Wireless Sensor Networks (WSNs), Network Privacy and Security. He has published papers in prestigious journals such as IEEE Transactions on Dependable and Secure Computing, ACM/Springer MONET, IEEE Network Magazine and IEEE Communications Magazine, in prestigious conferences such as INFOCOM, CNS and SECON.



Shaoen Wu received a Ph.D. in computer science in 2008 from Auburn University. He is presently an assistant professor with the Department of Computer Science at Ball State University. He has been an assistant professor in the School of Computing at the University of Southern Mississippi, a researcher scientist at ADTRAN Inc., and a senior software engineer at Bell Laboratories. His current research is in the areas of cyber security, wireless networking, cloud computing, and mobile computing.