Electronic Preprint for Journal of Information Processing Vol.27

Regular Paper

A Scheduling for Slotted-CSMA-based Wireless Mesh Networks to Reduce Delivery Delay

Takahiro Noi[1] Takuya Yoshihiro[2,a)]

Received: May 8, 2018, Accepted: November 7, 2018

Abstract: In this paper, we propose a new scheduling problem for WMNs based on slotted-CSMA. Slotted CSMA is a mechanism that divides a single frequency channel into several time slots where CSMA functions. With a schedule that matches links and slots, each node transmits frames in the assigned slot to avoid collision of frames. CATBS (CSMA-Aware Time-Boundable Scheduling) is a slotted-CSMA-based WMN architecture with a scheduling algorithm. However, it suffers from large end-to-end delay due to its long slot time that allows transmitting several frames within a single slot. This paper extends the scheduling problem of CATBS to consider inter-slot collision to reduce the overhead of collision at slot boundaries even if using short slot time. Evaluation results shows that the proposed scheduling problem reduces the overhead that arises at slot boundaries, and improves communication performance when using short time slots.

Keywords: wireless mesh networks, scheduling, slotted CSMA, routing

1. Introduction

Wireless Mesh Networks (WMNs) are a promising technology for widening the coverage of wireless networks [1], [2]. Large amount of studies have dedicated to develop the practical WMNs, but we have not yet succeeded in achieving high-performance WMNs that is applicable in practice. WMNs are often built on IEEE 802.11 devices for inexpensive and widely used system design, but all attempts so far have failed due to the hidden terminal problem. The hidden terminal effect is so harmful that the communication performance is severely degraded.

Several solutions have been considered so far. One of the most classical solutions is the RTS/CTS handshake [12] that is currently included in IEEE802.11 standard. RTS/CTS exchanges the RTS/CTS messages in advance to make the surrounding nodes suppress transmission, and then transmit data frame safely. However, the effect of RTS/CTS is known to be limited due to several reasons such as probabilistic collisions in RTS/CTS handshake, excessive suppression of transmissions known as the exposed terminal problem [13], and the inconsist effects caused by the difference between communication range and interference range [14]. Note that the difference between the communication and interference range grows larger when the communication speed increases, meaning that RTS/CTS does not work especially in high-speed environments. Even recently, IEEE802.11-based wireless mesh networks still suffer from heavy interference among nodes [3].

To reduce the influence of heavy interference, we proposed a novel joint MAC and Routing architecture for high-speed WMNs called CATBS (CSMA-Aware Time-Boundable Scheduling) [11]. CATBS is based on slotted CSMA, i.e., CSMA functions within time-divided slots, and avoid collisions due to hidden-terminals by applying a schedule that assigns a slot for each link. CATBS theoretically eliminates collisions and is robust to the time drift on synchronizing time among nodes. As a result, CATBS achieves high-speed communications with low frame loss over WMNs.

However, one of the problems in CATBS is its long end-to-end delay. Since CATBS uses slots in which several frames are transmitted, each packet inevitably experiences a delay while waiting for the active slot in which the packet is transmitted. This results in large end-to-end delay, which is not acceptable as a practical infrastructure.

In this paper, to reduce the end-to-end delay of packets in CATBS, we try to use far shorter slots in CATBS. In this case, an increasing number of collisions at the slot boundaries is known to cause a secere overhead problem. To reduce the overhead, we extend the scheduling problem of CATBS to consider inter-slot collision. We also propose using RTS/CTS handshake for all the frames, and show that this assumption can further improve the efficiency of the scheduling problem.

The reminder of this paper is organized as follows. In Section 2, we present related work on WMNs. In Section 3 we describe the base method CATBS concisely. In Section 4 we explain the proposed method in detail, and we evaluate the proposed method in Section 5. Finally we conclude the paper in Section 6.

2. Related Work

There are many studies on improving the performance of WMNs. In this section, we introduce a portion of those studies

1 Graduate School of Systems Engineering, Wakayama University, Wakayama 640–8510, Japan
2 Faculty of Systems Engineering, Wakayama University, Wakayama 640–8510, Japan
a) tac@sys.wakayama-u.ac.jp

that relate to this paper.

2.1 Utilization of Multiple Channels

One promising approach to eliminate the interference is to use multiple frequency channels. As a typical static channel assignment approach, Marina, et al. proposed a static channel assignment algorithm based on a greedy strategy that minimizes interference under the constraint that the network is connected [5]. Tang, et al. proposed INSTC [6] that formulates the similar channel assignment problem as an optimization problem based on linear programming (LP). However, those studies reveal that the severe interference is unavoidable in this basic approach within the few available channels 3-4, which is the number of orthogonal channels available in IEEE802.11 over 2.4 Ghz band. Ramachandran et al. proposed a method BFS-CA [7] that introduces an interference model called a multi-radio conflict graph that models the interference in a multi-radio environment. BFS-CA partially considers CSMA property in channel assignment that reduces the interference effects over the network, but it still suffer from considerable interference effects within the only 3 to 4 channels that are available. In the current state of the art, eliminating interference in the 3 to 4 channels is not possible.

2.2 Dynamic Channel Assignment

Dynamic channel assignment, in which interfaces dynamically switch their channels to avoid collisions, has been studied to further reduce the interference. Several such methods have been proposed so far. Raniwala, et al. proposed a IEEE802.11-based network system Hyacinth that has channel switching functions [8]. Draves et al. introduces a routing metric WCETT for optimal paths selection over multi-NIC WMNs [15]. So, et al. proposed a multi-channel MAC protocol that utilizes multiple channels to reduce collisions [9]. Although there are many studies following those approaches, they all have the common problem of a lack of stability that is unavoidable as long as they switch channels: Since synchronized channel transition between senders and receivers is mandatory, there exist a risk to fail synchronization, which may lead to instability of networks, and in the worst case the network is partitioned. We also note that the dynamic approach does not essentially solve the interference problem but is just a stopgap measure to reduce interference according to the traffic patterns.

2.3 Slotted CSMA Based Hybrid Approach

Slotted CSMA which utilize time-division slots within which CSMA works, is a kind of hybrid approachs of CSMA and TDMA (Time Division Multiple Access). Recently, several hybrid approach for mainly MAC protocols of sensor networks are appearing. One of the best known is IEEE 802.15.4 (ZigBee) [16], which is standardized for low-power sensor networks to utilize multi-hop communications. In IEEE 802.15.4, CSMA is used for scheduling and TDMA is used for scheduled data transmission. Many hybrid protocols for WSNs including Flex-iTP [17] and TRAMA [18] use a similar approach. However, this approach cannot be used in WMNs because in the TDMA phase we require accurate synchronization. Only a few studies use the same type of slotted CSMA as ours that runs only CSMA in every macro slot. We introduce the work of Liu et al. [19] in which each node has an allocated slot to receive frames from child nodes. However, if this scheme is applied to WMNs where packets as large as 1,500 bytes are usually transmitted, it suffers from serious collisions due to hidden terminals between any pair of child nodes in two-hop distance. Z-MAC [20] also uses the same kind of slotted CSMA, but it does not use tree scheduling. Z-MAC behaves like CSMA in low-contention state but in contrast behaves like TDMA in high-contention state under distributed TDMA scheduling such as DRAND [21]. This is done by back-off contention; in low-contention every node competes whereas in high-contention only the owner of the slot and its 1-hop neighbors can compete. Because a slot is shared among neighbors of the owner node even in high-contention state, Z-MAC provides flexible response to changes in traffic patterns and also reduces end-to-end delivery delay caused by waiting for available slots. However, Z-MAC also induces the hidden terminal problem between direct neighbors of an owner node placed in 2-hop distance. This develops into a serious problem in WMNs.

3. The Base Scheduling Method: CATBS

3.1 Overview of CATBS and its Problem

CATBS (CSMA-Aware Time-Boundable Scheduling) is an architecture of WMNs that incorporates slotted CSMA to avoid the hidden-terminal problem [11]. CATBS divides a single channel into several time slots and within each slot CSMA functions to avoid collision within the range of carrier-sensing range. To prevent the hidden terminal problem, CATBS computes a schedule that assigns a slot for each link. By considering the mechanism of CSMA and detouring paths, the schedule of CATBS requires a far lower number of slots to achieve a collision-free schedule. Note that CATBS is based on CSMA so that it maintains compatibility with the existing MAC schemes such as IEEE 802.11, which would enable CATBS to work on the major 2.4 GHz or 5 GHz ISM band. **Figure 1** shows the slotted CSMA in which a channel is divided into several magazine slots in each of which CSMA functions. **Figure 2** shows an example of a CATBS schedule in which hidden-terminal is eliminated while neighboring nodes are assigned with the same slot considering carrier-sensing functionalities of CSMA.

In the slotted system such as CSMA, the time-slot is required to be as short as possible to reduce the end-to-end delay of communications. However, the overhead on the boundaries of time slots is known to grow larger as the time slots go shorter. The overhead on the time-slot boundaries comes from collisions. In **Fig. 3**, a collision occurs on the boundaries of slot 1 and 2 where

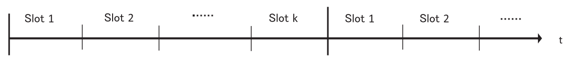


Fig. 1 Slotted CSMA.

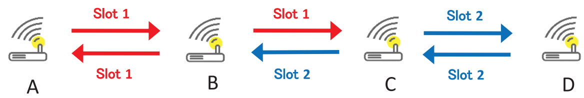


Fig. 2 Example of the CATBS schedule.

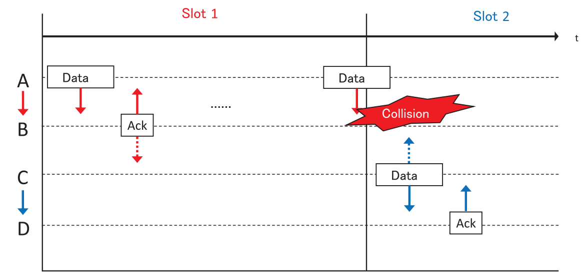


Fig. 3 Collisions on slot boundaries.

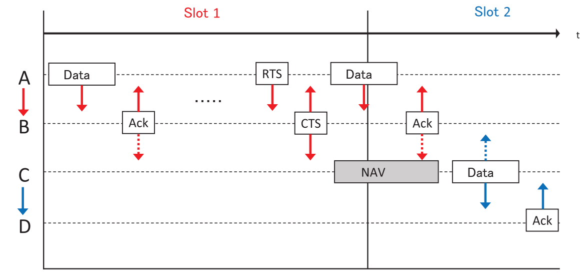


Fig. 4 Collision avoidance with RTS/CTS on slot boundaries.

transmission of a data frame spans slots 1 and 2, which collides with the data frame transmitted in slot 2, which is properly assigned to slot 2 within the applied schedule. In order to mitigate the problem, CATBS introduces RTS/CTS applied on the boundary of time slots. See **Fig. 4**. If the expected time to finish transmission of a data frame does not fall in the same slot, CATBS invokes RTS/CTS to prevent collision with a hidden terminal node assigned to the next slot. This mechanism enables CATBS to reduce the overhead due to slot boundaries, and simultaneously absorb the problem of the time synchronization errors, which allows deployment of a relatively loose synchronization mechanism [11]. However, CATBS still suffers from inter-slot collision.

3.2 Scheduling Problem of CATBS

CATBS computes a schedule that minimize collisions due to hidden terminal under the constraint of available channels. Generally in 2.4 GHz and 5 GHz ISM band, the available number of channel is not usually large. To reduce the required number of channels while aiming at achieving zero collision, CATBS introduces to consider the property of CSMA and to allow using paths longer than the shortest paths.

Let a directed graph $G = (V, E, C)$ be the topology of a network where V is a set of nodes, E is a set of edges, and C is a set of available slots. A link $e \in E$ is represented by $e = (u, v, c)$ where $u, v \in V$ and $c \in C$, meaning that every link in G is associated with a time slot c. Note that, according to schedules, each link in the network can be assigned to any slot. So, in our problem formulation, we first define the network topology G in which every link is assigned to every slot. Specifically, for every pair of neighbor nodes (u, v), links (u, v, c) for all $c \in C$ is included in E. From G, schedule $G' = (V, E', C)$ is derived as a subgraph of G where $E' \subset E$ that satisfies the basic constraint and simultaneously minimizes the collision among E'. The definition of collision among links is described in the following section.

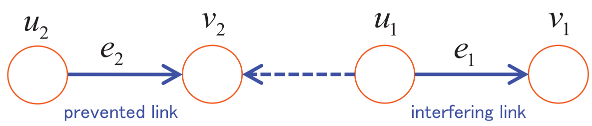


Fig. 5 Collision pattern 1 (CATBS).

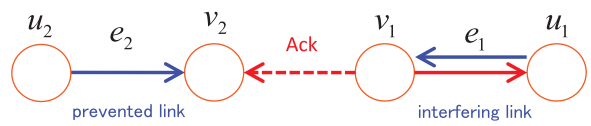


Fig. 6 Collision pattern 2 (CATBS).

3.3 Definition of Collision among Links

In CATBS, collision is defined as a pairwise property of links, i.e., each pair of links $e_1 = (u_1, v_1, c_1)$ and $e_2 = (u_2, v_2, c_2)$ are defined to collide or not. The scheduling problem in CATBS is designed based on the interference model so called the *single disk model* (or the protocol model) [22], in which the communication range and the interference range are represented by the same circle of a certain radius R. According to this definition, we let G be a graph in which two nodes u and v have links if and only if the distance between them is within R. Under the assumption, the link pairs (e_1, e_2) in collision are defined in the following two patterns.

Pattern 1 (Collision between Data frames)

A. $c_1 = c_2$

B. $(u_1, v_2, c_1) \in E$ (i.e., u_1 neighbors v_2).

C. $(u_1, u_2, c_1) \notin E$ (i.e., u_1 does not neighbor u_2).

Pattern 2 (Collision between Data and Ack frames)

A. $c_1 = c_2$

B. $(v_1, v_2, c_1) \in E$ (i.e., v_1 neighbors v_2).

C. $(u_1, u_2, c_1) \notin E$ (i.e., u_1 does not neighbors u_2).

Pattern 1 defines the case in which two data frames collide. See **Fig. 5**, in which a transmission on link e_2 is prevented by the simultaneous transmission on e_1 due to the hidden terminal problem. To make this happen, the slots of those two transmissions must be the same (Cond. A), and the radio of u_1 must reach v_2 (Cond. B). Additionally, if carrier sensing function of CSMA operates then this will not happen. Thus, two transmitters u_1 and u_2 must not be neighbors with each other (Cond. C). If all of those three conditions suffice, e_1 interferes e_2. On the other hand, the condition of Pattern 2, in which a Data and an Ack frame collide as shown in **Fig. 6**, is the same except that the link direction of e_1 is reverse.

Applying the definition, we can identify a set of collision link pairs S_G from a graph G. Naturally, from a schedule, we can identify a set of collision link pairs $S_{G'}$, and the number of which, denoted by $|S_{G'}|$, is the value that we wish to minimize in the problem formulation.

3.4 Scheduling Problem Formulation

CATBS formulates a scheduling problem as an optimization problem that minimizes the number of collision link pairs given the available number of slots. The scheduling in wireless multi-hop networks generally requires a large number of slots to eliminate collision. To reduce the number of required slots, CATBS incorporates two techniques: one that takes CSMA into account, and another that allows detour paths in routing. The former is included in the definition of collisions as described in the previous Section 3.3. The latter is considered in the definition of the scheduling problem with the path stretch factor k, by which, for each pair of source and destination nodes $(s, d) \in V$, we allow using the routing paths if their length is less than the shortest path plus k. By relaxing the condition of routing paths, the schedule can be sparser with less links, which reduces collisions significantly.

The input of the optimization problem is the network topology $G = (V, E, C)$ in which all combinations of links (u, v, c) for neighboring nodes (u, v) and available slots $c \in C$ are included in E. The output schedule is represented as a subgraph of G, which is $G' = (V, E', C)$ where $E' \subset E$ that minimizes the number of collision pairs defined as $|S_{G'}|$. We let $\delta^{G}_{(u,v)}$ be the shortest path length from u to v on G, and give the constraint of the paths length as $\delta^{G'}_{(u,v)} \leq \delta^{G}_{(u,v)} + k$ for all pairs of nodes (u, v). Also, since we assume that every node has the traditional architecture of routers so that it has only one output queue, the constraint that each node has just one sending slot is required. In summary, the problem formulation is given as follows.

Figure 7 shows an example of CATBS scheduling. From the input topology in which links with all available slots for each neighboring nodes exist, CATBS computes a very sparse collision-free schedule with only 3 slots by applying the paths stretch factor k. Note that, for every pair of source node s and destination node d, G' has a path shorter than $\delta^{G}_{(s,d)} + k$. In Fig. 7, the distance from s and d is 5 (hops) in G' while that in G is 1, which satisfies the path length constraint under $k = 4$. You can see that the path length constraint is satisfied for every source and destination pair in G.

The Scheduling Problem of CATBS

- Input: The network topology $G = (V, E, C)$
- Output: The schedule $G' = (V, E', C)$
- Constraint:
  (1) $\forall(u, v) \in V,\ \delta^{G'}_{(u,v)} - \delta^{G}_{(u,v)} \leq k$
  (2) $\forall u$, only one slot $c \in C$ such that $(u, v, c) \in E'$ exists.
- Subject to: Minimize the number of collision pairs $|S_{G'}|$

3.5 Solving the Scheduling Problem

Unfortunately, the scheduling problem belongs to the class NP-Hard so that computational time is far larger than practical. To solve the problem within a practical time bound, CATBS reduces the problem into a well-known NP-Hard problem called PMAX-SAT (Partial Max-SAT). For PMAX-SAT, many excellent solvers are available with which we can obtain approximated solution within a practical time limit. For specific methodologies, see the original paper [11].

4. The Proposed Method

4.1 Overview

As we mentioned before, CATBS suffers from large end-to-end delay if we deploy a relatively large slot duration. However, CATBS only cares the inter-slot interference by using RTS/CTS invoked at the slot boundaries. Since we have a collision pattern that cannot be avoided by the RTS/CTS-based method, considerable overhead arises due to loss at the slot boundaries. Naturally, this overhead grows larger when we use shorter slot duration in CATBS.

To reduce the loss due to collision at the slot boundaries, we propose a new scheduling problem that aware inter-slot collision occurring at the slot boundaries. We modify the collision model of CATBS shown in Section 3.3 to mitigate the inter-slot collision. With our inter-slot collision aware scheduling method, the communication performance of CATBS when deploying short time slots is significantly improved.

4.2 Defining Collision that Aware Inter-slot Interference

We extend the scheduling problem of CATBS to consider inter-slot collisions to reduce the overhead at slot boundaries. Only intra-slot collision is considered in the conventional collision defined in CATBS. So, (1) we first introduce a new collision definition that comprehensively includes both intra-slot and inter-slot collision. After that, (2) we further point out a pattern that will not collide in the short-slot case, and remove the pattern from the collision definition.

As for (1), the definition for the inter-slot collision is quite similar to the conventional one given in the previous Section 3.3: Since the interfering link e_1 is always assigned to the next slot of the prevented link e_2, we only assume that e_1 is allocated in the next slot of e_2. By considering both Patterns 1 and 2 for two cases where e_1 lies in the next slot or the same slot of e_2, the scheduling problem defined in CATBS is applicable as is.

As for (2), there is a case in which collision does not occur when the slot length is sufficiently short. To reduce end-to-end delays, we shorten the slot length so that it only contains one communication frame. In this case, every frame is transmitted with RTS/CTS handshake. As we assume that RTS/CTS is applied for every frame, we can find cases where collision does not exist.

See **Fig. 8** for this case. In this case, data-frame transmissions on two links $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ exist and both pairs of nodes (u_1, v_2) and (u_2, v_1) are neighboring. Note that v_1 and v_2

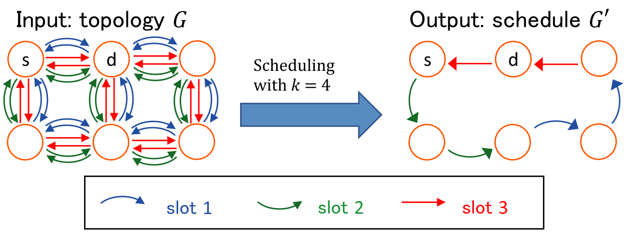


Fig. 7 Example of CATBS scheduling.

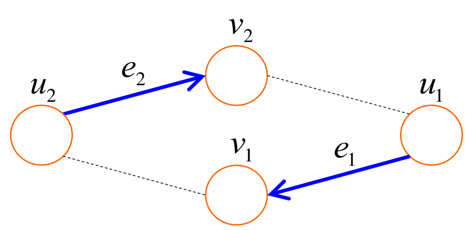


Fig. 8 The collision pattern avoidable by RTS/CTS.

could be the same node. If RTS/CTS is issued on e_1, both nodes u_2 and v_2 are aware of the transmission so that e_1 and e_2 does not collide.

Note that the collision definition of CATBS Pattern 1 includes the case; if the constraint that (u_2 and v_1 are neighboring is added to the CATBS collision Pattern 1, this case happens. By removing this pattern from the defined collision link pairs, we can reduce the collision count in the scheduling problem. Also note that, when we suppose the case where u_2 is neighboring v_1, even if RTS/CTS is exchanged on e_1, u_2 do not receive them and u_2 sends RTS on e_2. When the slot size is short, this RTS transmission significantly degrades the communication performance. Thus, the case where (u_2 is neighboring v_1 should remain in the collision definition.

As a result, the proposed method is expressed as follows. If a directed link pair $(e_1, e_2) \in E'$ satisfies either of the following Patterns 1 and 2, it is included in the collision set. Specifically in the proposed definition, we separately define the collisions in the same slot and in the different slots; for each link pair (e_1, e_2), if it matches either Pattern 1 or 2 and it belongs to the same slot, then it is included in $S_{G'}$, and if e_1 belongs to the next slot of e_2, it is included in $T_{G'}$.

Pattern 1 (Collision between Data frames)

A. $c_1 = c_2$, or, c_1 is the next slot of c_2

B. $(u_1, v_2, c_1) \in E$ (i.e., u_1 neighbors v_2).

C. $(u_1, u_2, c_1) \notin E$ (i.e., u_1 does not neighbor u_2).

D. $(u_2, v_1, c_1) \notin E$ (i.e., u_2 does not neighbor v_1).

Pattern 2 (Collision between Data and Ack frames)

A. $c_1 = c_2$, or c_1 is the next slot of c_2

B. $(v_1, v_2, c_1) \in E$ (i.e., v_1 neighbors v_2).

C. $(u_1, u_2, c_1) \notin E$ (i.e., u_1 does not neighbors u_2).

With the definition above, we formulate our problem as follows. Under the constraint that the collision in the same slot is kept as zero, we minimize the inter-slot collision in the schedule.

The Proposed Scheduling Problem

- Input: The network topology $G = (V, E, C)$
- Output: The schedule $G' = (V, E', C)$
- Constraint:
  (1) $\forall (u, v) \in V$, $\delta^{G'}_{(u,v)} - \delta^{G}_{(u,v)} \leq k$
  (2) $\forall u$, only one slot $c \in C$ such that $(u, v, c) \in E'$ exists.
  (3) The number of collision pairs on the same slot is zero, i.e., $|S_G| = 0$.
- Subject to: Minimize the number of inter-slot collision pairs $|T_{G'}|$

5. Evaluation

5.1 Evaluation of Scheduling Algorithm

5.1.1 Method

We first evaluate the scheduling time and optimality of our scheduling problem. As a typical network topology of WMNs, we assume 10×10 grid topology in which each node can communicate with four nodes located in vertical and horizontal direction. We varied the number of slots $|C| = 3$ to 10, and the stretch factor $k = 0$ to 10.

When the input G, $|C|$ and k are given, they are transformed to the corresponding PMAX-SAT formula, and solved by a PMAX-SAT solver. As a PMAX-SAT solver, we used QMAX-SAT [4] developed by Koshimura. We limit the execution time of the solver to 3,600 seconds because empirically longer execution does not improve the solution largely. As a execution environment, we use a consumer PC with Intel(R) Xeon(R) Processor E3-1280 (3.70 GHz, 8 MB Cache), 64 GB memory.

Table 1 Residual collision in CATBS schedule.

	3slot	4slot	5slot	6slot	7slot	8slot	9slot	10slot
$k = 0$	2,295	1,623	1,012	651	281	0	0	0
$k = 2$	1,087	703	273	0	0	0	0	0
$k = 4$	824	344	99	0	0	0	0	0
$k = 6$	491	232	0	0	0	0	0	0
$k = 8$	463	182	0	0	0	0	0	0
$k = 10$	385	119	0	0	0	0	0	0

Table 2 Residual collision with technique (1).

	3slot	4slot	5slot	6slot	7slot	8slot	9slot	10slot
$k = 0$	-	-	-	-	-	944	1,077	885
$k = 2$	-	-	-	773	605	682	423	345
$k = 4$	-	-	-	546	444	355	296	173
$k = 6$	-	-	510	396	352	286	179	69
$k = 8$	-	-	410	371	302	212	130	22
$k = 10$	-	-	366	367	231	170	77	17

Table 3 Residual collision with techniques (1), (2).

	3slot	4slot	5slot	6slot	7slot	8slot	9slot	10slot
$k = 0$	-	-	-	-	-	690	843	704
$k = 2$	-	-	-	576	532	410	327	215
$k = 4$	-	-	525	405	319	216	165	20
$k = 6$	-	674	402	372	229	103	42	0
$k = 8$	-	450	344	267	172	83	23	0
$k = 10$	-	379	367	191	121	61	0	0

We execute CATBS, and two proposed methods, i.e., the case only technique (1) is applied, and the case both techniques (1), (2) are applied. For each of the techniques (1), (2), see Section 4.2.

5.1.2 Results

We first show the results of CATBS scheduling in **Table 1**. Note that the scheduling problem of CATBS minimizes the number of collision on the same slot included in the output schedule G'. So, the value zero means that the optimal solution G' is found in which no collision link pair as defined in Section 3.3 remains. The results in Table 1 shows that, if we have more than 4 slots, the optimal solution, i.e., zero-collision solution, is found for some values of k.

Next, **Table 2** shows the proposed method with technique (1). Note that the scheduling problem is different from CATBS; we constrain the collision on the same slot to be zero, i.e., $|S_{G'}| = 0$ while minimizing the inter-slot collision $|T_{G'}|$. In the result, we see that every place where zero-collision solution was not found in CATBS, the proposed method also did not find the solution that satisfies the constraint $|S_{G'}| = 0$. This is natural because the collision pair set S_G is the same in both cases. In the remaining cases, although inter-slot collision still remains, they reduce the number of inter-slot collision link pairs. Note that, in CATBS, more than thousands of inter-slot collision remains in each case so that the proposed method clearly reduces the inter-slot collisions.

Finally, **Table 3** shows the proposed method with both techniques (1), (2). We see that in every case the inter-slot collision link pairs reduces. Moreover, the computable cases, i.e., the case where $|S_{G'}| = 0$, increase because the collision link pairs in S_G reduces due to technique (2).

From above, we confirmed that the proposed scheduling problem not only reduces the inter-slot collision, but also increases the feasible cases in which intra-slot collision is zero. Specifically, the proposed method expands the feasible cases in which possible intra-slot collision patterns under the single disk model is zero. We have feasible solutions with only 4 slots and $k \geq 6$.

5.2 Traffic Simulation

5.2.1 Scenario

We apply the schedule of CATBS and the proposed methods using a network simulator to evaluate the traffic performance of the schedule. As a network simulator, we use Scenargie [10], which runs up-to-date communication models.

As a network topology, we use 10×10 grid topology again in which node interval is 500 [m]. Since the transmission power is set as 20 [dBm], every node neighbors at most four nodes in the vertical and horizontal directions. Each node runs IEEE802.11g, but it is modified to behave based on slots according to the mechanism of CATBS. We fixed the slot length to 2 [ms] through preliminary tests to find the adequate value with which the proposed method effectively works.

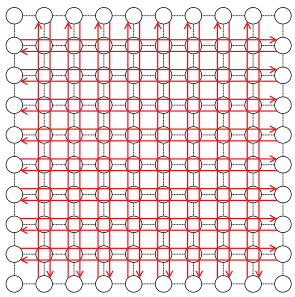


Fig. 9 Topology and flows for evaluation.

We have two choices on CATBS deployment strategy, i.e., distributed and central deployment. The former one combines CATBS with a link-state routing protocol such that CATBS computes a schedule from the collected network topology, and the protocol computes the shortest paths over the schedule. However, this requires the farther challenge of designing a new routing protocol for distributed operation of CATBS. Thus, we this time chose the central deployment to measure the basic performance of the proposed method. Specifically, only at the beginning of the scenario, a central controller computes a CATBS schedule from the network topology and installs the schedule to each node. To mimic this in our simulation, we statically installed the routing table into each node so that packets are routed along the shortest paths computed over a CATBS schedule.

As schedules of both CATBS and the proposed method, we applied those with 8 slots and $k = 0$ computed in the previous section. We fixed $k = 0$ to measure the basic performance of the proposed techniques by excluding the effect of path detouring, and 8 is the smallest number of slots that achieve zero intra-slot collision.

CATBS supposes that routing paths are computed based on some shortest path based routing protocol over the schedule. , but this time we statically installs the routing tables that are computed from the schedule used in the scenario. We generate 32 CBR (Constant Bit Rate) flows as shown in **Fig. 9** with 1,500 [Bytes] packets from time 0 [s] to 540 [s]. After waiting until the traffic stabilizes, we measure the traffic performance from 60 [s] to 540 [s]. The transmission rate of each CBR flows are varied be-

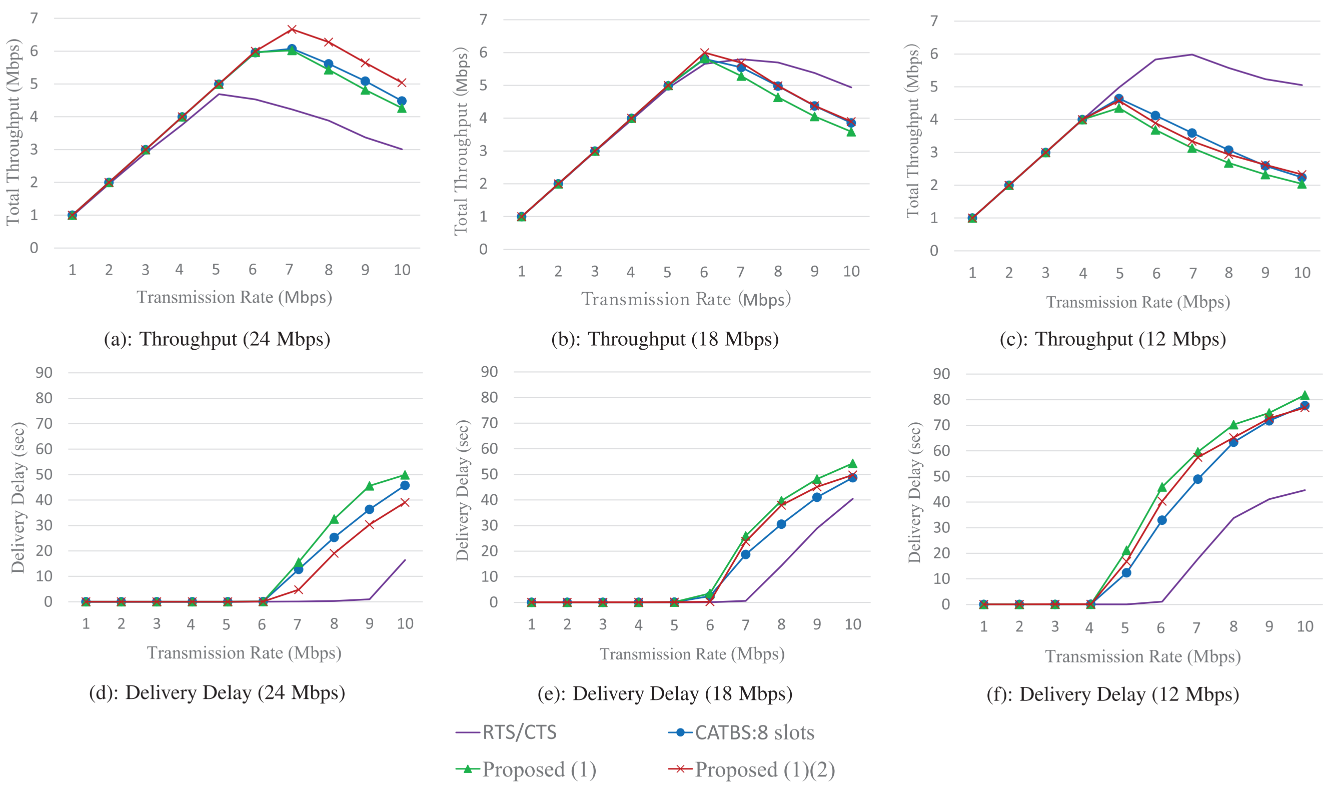


Fig. 10 Evaluation results 1.

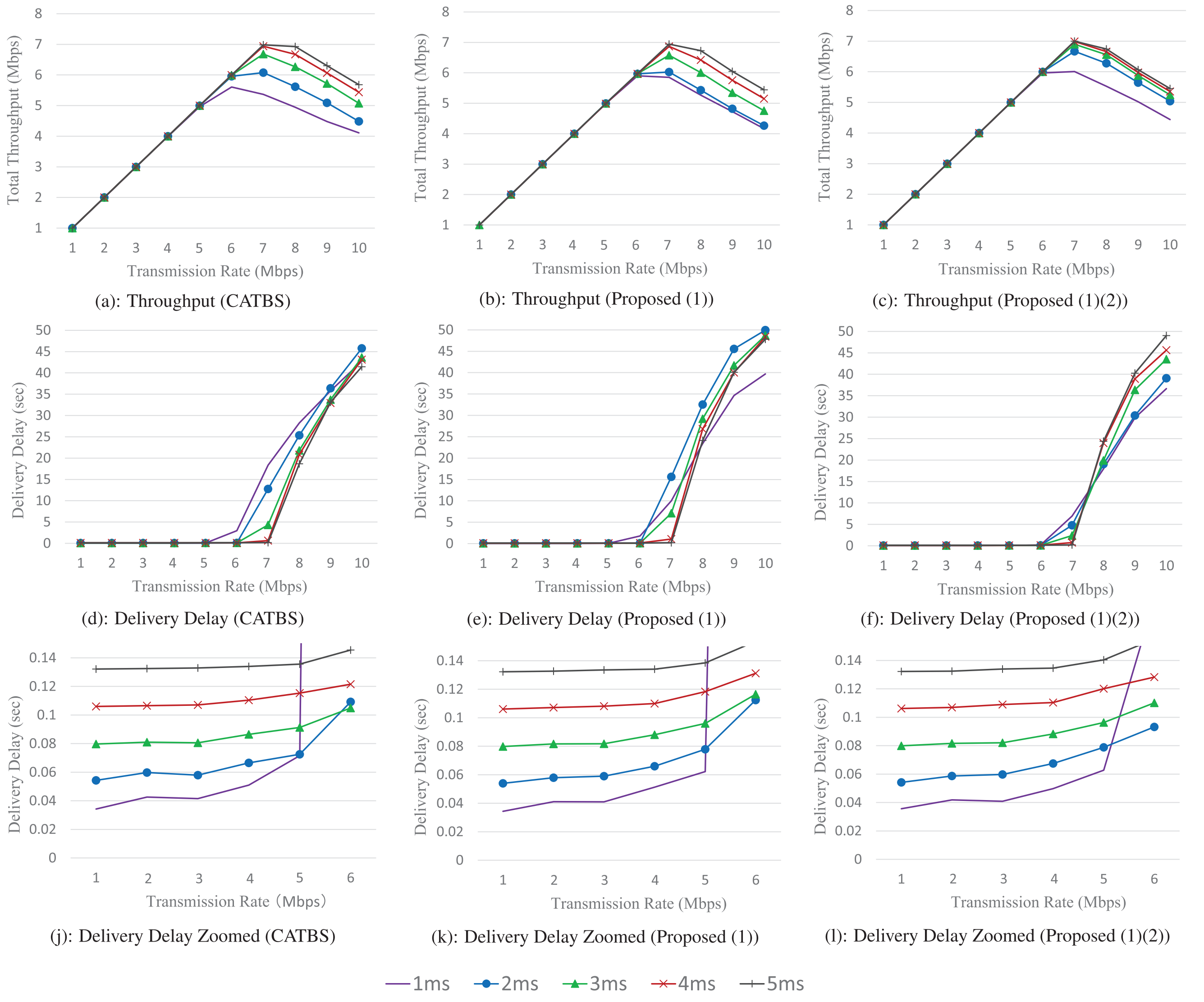


Fig. 11 Evaluation results 2.

tween 1–10 [Mbps].

We compared CSMA/CA (i.e., CSMA with RTS/CTS), CATBS, and two proposed methods. Note that CSMA/CA does not use time-division slots, it is well-known normal CSMA/CA.

5.2.2 Results

First, we see the comparison results shown in **Fig. 10**. Among the four compared methods, CATBS and the two proposed methods marks a similar performance. The throughput initially improves in proportion to the offered load, but it reduces after the offered load exceeds the link capacity. In contrast, CSMA/CA (i.e., CSMA with RTS/CTS) shows different tendency; it offers a good performance under low-speed links whereas bad performance under high-speed links. Note that the tendency exactly matches the results in the literature in which RTS/CTS does not work especially in high-speed environment due to the radio property of physical layer. We see that the performance degradation due to collision in CATBS and proposed methods are far lower than the conventional CSMA/CA schemes. Note that in both CATBS and the proposed methods, although throughput improves as link speed raises, the improvement is not in proportion to link speed. This is due to higher interference with higher link speed. Namely, although throughput actually improves as link speed raises, the improvement is limited by interference among nodes that grows larger under higher link speed.

Next, we see the effect of changing slot length in **Fig. 11**, in which we executed the simulation with 24 Mbps links. In Figs. 11 (a)–(c), we see that the throughput performance is better when the slot length is long. This is due to the collision at the slot boundaries. Also we see that the proposed method (1) outperforms CATBS, and the proposed method (1), (2) outperforms method (1). Specifically, the proposed methods prevent the degradation of performance when slot size shortens. In other words, the proposed methods are robust against the overhead at the slot boundaries. In Figs. 11 (d)–(f), delay performance is shown, which also indicates the robustness of the proposed methods against the overhead at the slot boundaries. In Figs. 11 (g)–(i), delay performance focused on the unsaturated state is shown. We see that the end-to-end delay is reduced according to the slot size applied. As above, we can conclude that both proposed methods (1) and (2) are effective in improving delay performance with less throughput degradation due to the collision at the slot boundaries.

5.3 Discussion on Time Synchronization

Note that our simulation does not consider errors in time synchronization among nodes. Generally time synchronization with millisecond order accuracy is a challenging task for commodity systems in practice. However, several research papers reported that they achieved synchronization on the order of tens to hundreds of microseconds [23], [24]. They showed that the accurate time synchronization may be achievable in the future.

Note that the effect synchronization error in the proposed method is limited because it considers collision with neighboring slots. So, we can basically allow the error smaller than the slot length. Of course, if the error is introduced in simulation, the performance may be degraded. However, we can use the schedule with smaller inter-slot collisions to prevent degradation in performance. In contrast, although RTS/CTS at slot boundaries avoids collision to some extent, CATBS still suffer from inter-slot collisions and the performance degradation is not small enough. As above, it is concluded that the proposed method is more robust against synchronization error than CATBS.

6. Conclusion

In this paper, we extend the scheduling problem of CATBS and propose a new scheduling problem that reduces the collision at the slot boundaries in CATBS. The proposed scheduling problem is aware of inter-slot collision that occurs at the slot boundaries and reduces collision in cases using a short slot length. By using short slot length in CATBS, we can significantly reduce the end-to-end delay, which is an important performance measurement index in WMNs. The proposed method enables us to use short-slot schemes without introducing heavy overhead due to inter-slot collision which improves the end-to-end delay without degrading the throughput performance of WMNs.

References

[1] Akyildiz, I.F. and Wang, X.: *Wireless Mesh Networks*, John Wiley & Sons, Ltd., Publication (2009).

[2] Pathak, P.H. and Dutta, R.: A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks, *IEEE Communications Surveys & Tutorials*, Vol.13, No.3 (2011).

[3] Sheshadri, R.K. and Koutsonikolas, D.: Comparison of Routing Metrics in 802.11n Wireless Mesh Networks, *The 32nd IEEE International Conference on Computer Communications* (*INFOCOM '13*) (2013).

[4] Koshimura, M., Zhang, T., Fujita, H. and Hasegawa, R.: QMaxSAT: A Partial Max-SAT Solver, *Journal on Satisability, Boolean Modeling and Computation*, Vol.8, pp.95–100 (2012).

[5] Marina, M.K., Das, S.R. and Subramanian, A.P.: A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks, *Computer Networks*, Vol.54, pp.241–256 (2010).

[6] Tang, J., Xue, G. and Zhang, W.: Interference-aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks, *Proc. MobiHoc '05* (2005).

[7] Ramachandran, K.N., Belding, E.M., Almeroth, K.C. and Buddhikot, M.M.: Interference-aware Channel Assignment in Multi-Radio Wireless Mesh Networks, *Proc. Infocom '06* (2006).

[8] Raniwala, A. and Chiueh, T.: Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network, *Proc. IEEE INFOCOM '05*, pp.2223–2234 (2005).

[9] So, J. and Vaidya, N.: Multi-channel MAC for Ad Hoc Networks: Handling Multi-channel Hidden Terminals Using A Single Transceiver, *Proc. ACM MobiHoc '04*, pp.222–233 (2004).

[10] Network Simulator Scenargie, available from ⟨https://www.spacetime-eng.com/en/products⟩.

[11] Yoshihiro, T. and Nishimae, T.: Practical Fast Scheduling and Routing over Slotted CSMA for Wireless MeshNetworks, *Proc. IEEE/ACM International Symposium on Quality of Service* (*IWQoS2016*) (2016).

[12] Bharghavan, V. et al.: MACAW: A Media Access Protocol for Wireless LANs, *Proc. ACM SIGCOMM* (1994).

[13] Sobrinho, J.L., de Haan, R. and Brázio, J.M.: Why RTS-CTS Is Not Your Ideal Wireless LAN Multiple Access Protocol, *Proc. WCNS '05* (2005).

[14] Xu, K., Gerla, M. and Bae, S.: Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks, *Ad Hoc Networks*, Vol.1 No.1, pp.107–123 (2003).

[15] Draves, R., Padhye, J. and Zill, B.: Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks, *Proc. 10th Annual International Conference on Mobile Computing and Networking* (*MOBICOM2004*), pp.114–128 (2004).

[16] IEEE 802.15.4b standard, Wireless Medium Access Control and Physical Layer Specification for Low Rate Wireless Personal Area Networks (2006).

[17] Lee, W.L., Datta, A. and Cardell-Oliver, R.: FlexiTP: A Flexible-Schedule-Based TDMA Protocol for Fault-Tolerant and Energy-Efficient Wireless Sensor Networks, *IEEE Trans. Parallel and Distributed Systems*, Vol.19, No.6 (2008).

[18] Rajendran, V., Obraczka, K. and Garcia-Luna-Aceves, J.J.: Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks, *Proc. SenSys '03* (2003).

[19] Liu, Q., Chang, Y. and Jia, X.: Real-time Data Aggregation for Contention-based Sensor Networks in Cyber-Physical Systems, *Proc. 7th International Conference, WASA2012* (2012).

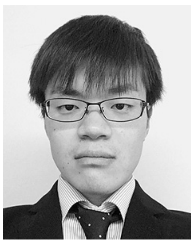
[20] Rhee, I., Warrier, A., Aia, M. and Min, J.: Z-MAC: A Hybrid MAC for Wireless Sensor Networks, *Proc. SenSys '05* (2005).

[21] Rhee, I., Warrier, A., Min, J. and Xu, L.: DRAND: Distributed randomized TDMA scheduling for wireless ad-hoc networks, *Proc. 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (*MobiHoc '06*) (2006).

[22] Gupta, P. and Kumar, P.R.: The capacity of wireless networks, *IEEE Trans. Information Theory*, Vol.46 No.2, pp.388–404 (2000).

[23] Chiang, J.H. and Chiueh, T.: Accurate clock synchronization for IEEE 802.11-based multi-hop wireless networks, *17th IEEE International Conference on Network Protocols* (*ICNP2009*), pp.11–20 (2009).

[24] Koutsonikolas, D., Salonidis, T., Lundgren, H., LeGuyadec, P., Hu, Y.C. and Sheriff, I.: TDM MAC protocol design and implementation for wireless mesh networks, *Proc. CoNEXT '08* (2008).



Takahiro Noi received his B.E. and M.E. degrees from Wakayama University in 2015 and 2017, respectively. He is currently working with Fujitsu FSAS Inc. He is interested in wireless networking and communications as well as systems on them.

Takuya Yoshihiro received his B.E., M.I. and Ph.D. degrees from Kyoto University in 1998, 2000 and 2003, respectively. He was an assistant professor at Wakayama University from 2003 to 2009. He has been an associate professor in Wakayama University from 2009. He is currently interested in graph theory, distributed algorithms, computer networks, wireless networks, medical applications, bioinformatics, etc. He is a member of IEEE, IEICE, and IPSJ.