Transient Coordinator: a Collision Resolution Algorithm for Asynchronous MAC Protocols in Wireless Sensor Networks

Sang Hoon Lee[1] Byung Joon Park[2] and Lynn Choi[1]
[1]School of Electrical Engineering, Korea University
Seoul, 136-701, Korea
[2]Department of Computer Science, Kwangwoon University
Seoul, 139-701, Korea
[e-mail: {smile97, lchoi}@korea.ac.kr, bjpark@kw.ac.kr]
*Corresponding author: Lynn Choi

Received June 20, 2012; revised October 16, 2012; revised October 30, 2012; accepted November 14, 2012; published December 27, 2012

Abstract

Wireless sensor networks (WSN) often employ asynchronous MAC scheduling, which allows each sensor node to wake up independently without synchronizing with its neighbor nodes. However, this asynchronous scheduling may not deal with collisions due to hidden terminals effectively. Although most of the existing asynchronous protocols exploit a random back-off technique to resolve collisions, the random back-off cannot secure a receiver from potentially repetitive collisions and may lead to a substantial increase in the packet latency. In this paper, we propose a new collision resolution algorithm called Transient Coordinator (TC) for asynchronous WSN MAC protocols. TC resolves a collision on demand by ordering senders' transmissions when a receiver detects a collision. To coordinate the transmission sequence both the receiver and the collided senders perform handshaking to collect the information and to derive a collision-free transmission sequence, which enables each sender to exclusively access the channel. According to the simulation results, our scheme can improve the average per-node throughput by up to 19.4% while it also reduces unnecessary energy consumption due to repetitive collisions by as much as 91.1% compared to the conventional asynchronous MAC protocols. This demonstrates that TC is more efficient in terms of performance, resource utilization, and energy compared to the random back-off scheme in dealing with collisions for asynchronous WSN MAC scheduling.

Keywords: Wireless sensor networks, collision resolution, MAC protocols, asynchronous scheduling, random back-off

---

A preliminary version of this paper appeared in KCC 2012, June 27-29, Jeju, Korea. This version includes a concrete analysis and additional simulations. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012- 0002515). This research was supported by a Korea University Grant (K1216991).

http://dx.doi.org/10.3837/tiis.2012.12.007

1. Introduction

Most of the existing WSN applications [1][2][3] usually assume converge-cast traffic that consists of concentrated simultaneous packets from multiple sources to a few sink nodes. Considering the burst and convergent traffic pattern of WSNs, both contentions and collisions among multiple senders would be the major factors for degrading the network performance. In particular, collisions may lead to accumulated data packets, which in turn will result in additional collisions. Asynchronous WSN MAC protocols such as sender-initiated preamble sampling [4][5][6] or receiver-initiated MAC protocols [7][8] may suffer from more collisions than synchronous protocols [9][10][11][12] under the same traffic scenario since they usually do not employ RTS/CTS handshaking for collision avoidance. Most of the existing asynchronous MAC protocols [4][5][6][7][8] utilize the random back-off technique to resolve collisions instead of avoiding them. In the random back-off technique each node retries the packet transmission by backing off a random value which is lower than the contention window size. Especially, in the exponential random back-off technique a node exponentially increases its contention window size if there is a consecutive collision.

However, the random back-off algorithm often increases buffering delay and accumulates contenders since all the contenders except a winner repetitively back off. This exacerbates the performance degradation especially within a hot spot around a sink node. Consequentially, the message latency as well as the energy consumption due to repetitive contentions and collisions may grow substantially especially under the converge-cast traffic. Therefore, an energy-efficient and low-latency collision resolution algorithm that can prevent repetitive contentions and collisions would be one of the key techniques for performance improvement.

To address the performance degradation due to repetitive contentions and collisions in asynchronous WSN MAC protocols, we propose a new collision resolution algorithm called Transient Coordinator (TC) that can resolve a collision on demand in a single resolution process. The receiver detecting a collision acts as a central coordinator and derives a collision-free transmission sequence by collecting the information from competitive senders. To accomplish this, we employ a time division multiple access (TDMA) scheme called *reservation based channel allocation* (RBCA) for information collection and channel allocation. A RBCA process consists of *collision announcement*, *contention-less reservation request*, and *sequential data transmission*. First, the receiver detecting a collision broadcasts a NACK packet with a collision flag set, announcing the collision and commencing a collision resolution procedure. Then, each sender transmits a dedicated control signal called *reservation signal* during a pre-allocated signal slot within a contention-less reservation request period. After receiving the reservation signals, the receiver orders the transmission sequence and the senders retransmit their packets during the following sequential data transmission period. By deriving a collision-free transmission sequence and by assigning exclusive transmission time to each sender, TC prevents repetitive collisions and thereby improves the network performance.

Without careful slot allocation, reservation signals may also collide. To prevent collisions among the reservation signals, we apply a vertex coloring algorithm to our slot allocation. However, the number of signal slots is generally proportional to the number of neighbor nodes. Thus, as the node density increases, both the time and the energy consumption for collision resolution may increase due to the long contention-less reservation request period. To reduce the time and the energy wastes we exploit a *double coloring algorithm* that allows two different nodes to share a single signal slot. Since each node transmits a reservation signal only during its dedicated slot, the receiver can figure out which nodes collide during a signal slot. In other words we exploit collisions as additional information source while the conventional MAC protocols regard collisions as what they have to avoid. Our double coloring algorithm can reduce the number of signal slots by half.

Consequently, each sensor node basically uses a contention based medium access scheme, but it utilizes a reservation based channel allocation on demand whenever there is a collision. TC can be integrated with all the existing asynchronous WSN MAC protocols [4][5][6][7][8] in an on-demand manner. To evaluate the performance of our algorithm we applied TC to two representative asynchronous MAC protocols: WiseMAC [5] and RI-MAC [7]. According to the detailed packet-level simulation results, we verify and demonstrate that TC can effectively reduce unnecessary time, bandwidth, and energy from repetitive collisions. TC improves the per-node throughput, the latency overhead, and the energy overhead by up to 19.4%, 71.2%, and 91.1% respectively compared to WiseMAC.

The rest of this paper is organized as follows. Section 2 reviews the existing asynchronous WSN MAC protocols and their collision resolution schemes. Section 3 presents the main idea and the detailed collision resolution procedure of TC. Section 4 analyzes both the latency and energy performance of TC compared to those of the existing random back-off scheme. Section 5 discusses simulation methodology and presents the detailed evaluation results of TC and two existing asynchronous WSN MAC protocols. Section 6 concludes the paper.

2. Related Work

The existing asynchronous WSN MAC protocols can be classified into two types: sender-initiated protocols and receiver-initiated protocols. In the *sender-initiated protocol* each sender starts its transmission without checking its receiver's wakeup. A preamble sampling scheme called LPL (low power listening) [4] is the most representative sender-initiated scheme. In the *receiver-initiated protocol* each sender starts its transmission after confirming its receiver's wakeup by receiving a beacon from the receiver.

Most of the conventional asynchronous MAC protocols [4][5][6] are sender-initiated protocols. B-MAC [4] is one of the earliest studies for asynchronous MAC protocols which exploits a preamble sampling scheme. In this protocol, a sender transmits a preamble long enough to cover the wakeup period of the receivers. This can not only minimize idle listening but it also allows each node to wake up asynchronously independent of other nodes. WiseMAC [5] extended this idea. In WiseMAC a sender can minimize the length of a preamble by piggybacking neighbors' sampling schedules during communication, reducing both the preamble transmission overhead and the preamble sampling time of the receivers. A recent scheme called X-MAC [6] is also based on preamble sampling but uses multiple short preambles containing receiver address information. With the address information, overhearing nodes can immediately go back to sleep while only the intended receiver participate in the communication by replying an ACK packet. By hearing the ACK packet, the sender can stop sending the preambles, reducing the preamble transmission overhead.

RI-MAC [7] is the representative receiver-initiated protocol. In RI-MAC each node transmits a beacon when it wakes up; notifying to its neighbors that it is ready to receive a packet. A sender keeps awake until it receives a beacon from its receiver. It starts to transmit its data packet after receiving the beacon.

Most of the existing asynchronous protocols [4][5][7][8] exploit a random back-off scheme to deal with collisions while X-MAC [6] can partially avoid collisions by exchanging preamble and ACK packets. The random back-off is simple and effective under the light traffic condition where collisions rarely occur. However, as the traffic volume increases it may increase buffering delay and accumulate contenders around a sink since all the contenders except a winner repetitively back off. This may also increase packet drops on heavy collisions. In addition, it may not avoid repetitive collisions, possibly leading to an indefinite postponement problem.

3. Transient Coordinator (TC)

TC is carried out in three steps, by: (1) announcing a transmission failure and the beginning of a collision resolution process; (2) collecting the information about collided senders; (3) arranging the transmission sequence of senders and retransmitting data packets. To embody these three steps TC employs a TDMA scheme called RBCA as a collision resolution process, which consists of collision announcement, contention-less reservation request, and sequential data transmission. **Fig. 1** shows an example of a collision resolution process of TC. Each sensor node basically uses a contention based medium access scheme, but it employs a reservation based channel allocation on demand whenever there is a collision.

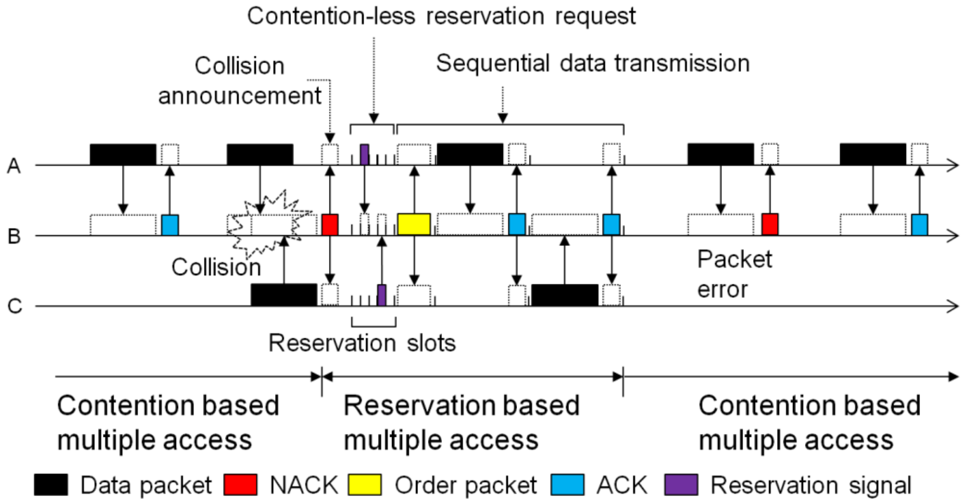

Fig. 1. The collision resolution process of TC

3.1 Collision Announcement

As shown in **Fig. 1**, on a collision, the receiver broadcasts a NACK packet with a collision flag set, announcing the collision and commencing a collision resolution procedure. On a packet error, the receiver broadcasts a NACK packet without a collision flag set, invoking a packet retransmission.

A wireless node generally can detect a collision by checking whether the length of an incoming packet is longer than the original length recorded in the MAC header field [13]. It also can detect a transmission error due to a background noise by using an error detecting code such as a cyclic redundancy check (CRC) [14]. However, if a node fails to decode the header field or the error detecting code, then it cannot clarify the cause of its reception failure. To address this problem we exploit the electrical signal characteristics of a collision signal to distinguish it from a background noise [15]. In general, the received signal strength of a collision signal is higher than a normal signal and its variation is consistent and small. In contrast, the signal strength of a background noise is generally lower than a normal signal and its strength variation is inconsistent and irregular. Therefore, if a node receives an indecipherable signal which has regular and consistent strength above a normal signal threshold value, we can conclude that there is a collision [15].

Since a receiver can transmit a NACK packet after receiving all simultaneous transmissions, the senders may not know when they can retransmit data. To avoid a collision between a NACK packet and a data packet, each node uses an *ACK timer* to hold its communication. If a sender receives neither an ACK packet nor a NACK packet before the ACK timer expires, it believes that there was a packet error and the sender retransmits the data packet.

3.2 Contention-less Reservation Request

After broadcasting a NACK packet the receiver carries out RBCA to collect the information about collided senders. The receiver divides its contention-less reservation request period into multiple *reservation slots* and allocates a dedicated reservation slot to each neighbor node. Each collided sender receiving the NACK packet requests the receiver to allocate an exclusive transmission time in the following sequential data transmission period by transmitting a *reservation signal* during its reservation slot. Since the receiver maintains its slot allocation information, each sender can notify a receiver of its presence by simply transmitting a very short signal, thereby minimizing both the energy and the time for transmitting the signal.

However, the number of neighbor nodes may be variable because of node failure or additional node deployment. Therefore, TC dynamically adjusts the number of reservation slots to avoid the shortcomings of the reservation slots. Excessive number of slots may lead to unnecessary time for increased reservation request period while the shortage in the number of slots may cause additional collisions among the nodes that are assigned to the identical slot. When a node detects a new neighbor, it allocates an empty or supplementary slot to the neighbor. To prevent collisions among the reservation signals, no more than one pair of nodes can share the same reservation slot within a two-hop communication range. This slot allocation problem can be regarded as an edge coloring problem, which can be converted to a vertex coloring problem. In the slot allocation problem each node carries out an allocation only for a new node while it maintains the information about pre-allocated signal slots. Therefore, we apply the C++ Boost Graph Library [16] that contains an implementation of greedy incremental vertex coloring heuristics to the slot allocation problem.

Note that the number of reservation slots is generally proportional to the number of neighbor nodes. Therefore, although we minimize the length of a reservation signal, both the time and the energy consumption for collision resolution may increase due to the long contention-less reservation request period as the node density increases. To reduce both the time and energy consumption due to exclusive slot allocation based on a vertex coloring algorithm we propose a *double coloring algorithm* that allows two different neighbor nodes to share a single reservation slot. Since each node maintains its slot allocation information, a receiver node can figure out which nodes simultaneously transmit reservation signals when it detects a collision during a reservation slot. Therefore, each node can extract the senders' information from collisions during a contention-less reservation request period. In other words we exploit collisions as additional information source while the conventional MAC protocols regard collisions as what they have to avoid. Our double coloring algorithm can reduce the number of reservation slots by half.

3.3 Sequential Data Transmission

After the reservation request period, a receiver arranges the transmission sequence of senders in the order of the received reservation signals; the receiver broadcasts an *order packet* that contains the arranged sequence. Assuming a fixed size data packet, each sender can estimate the time for exchanging a data packet and an ACK packet. Therefore, each sender suspends its transmission until its data transmission slot. However, WSN applications may have packets of variable sizes. In this case, each sender transmits its data packet after overhearing an ACK packet from the previous data transmission on the transmission sequence.

Note that TC provides exclusive data transmission time to each sender. Therefore, to avoid a collision during the consecutive data transmissions all the collided senders should sleep after their transmission until the end of the sequential data transmission period. Since all the senders can check the number of collided senders, they can estimate the time when the sequential data transmission period ends. However, an extra sender which doesn't participate in the on-going collision resolution may attempt to transmit a packet to another receiver. Fortunately, we can simply reduce this collision by inserting a flag bit called *on-going* bit into an ACK packet. The on-going bit indicates that there is an on-going resolution process and requests any sender to overhear for a single data packet transmission time before starting its transmission. If an additional collision due to hidden terminals nevertheless occurs, a sender cannot receive an ACK packet. In this case, the sender retransmits its packet after the time long enough to finish a single data packet transmission. Since both the hidden terminal and its receiver would reside in the outside of the sender's communication range, the hidden terminal's transmission will complete normally. Therefore, the sender can retransmit its packet without an additional collision by holding its transmission for a single packet transmission time.

4. Analytic Evaluation

In this section we analyze the time and the energy consumption of TC compared to the random back-off scheme [14]. For simplicity, we assume that the length of a single back-off slot is long enough to cover a single data transaction that consists of a DATA packet followed by an ACK packet exchange. We further assume that the communication range of a node has a hexagon shape whose side is r and the average number of neighbor nodes for each node is d. Then, the average number of collided senders can be given by $2+\varepsilon(d-2)$ where ε denotes the event rate of a node and $\varepsilon(d-2)$ represents the number of extra senders according to the event rate.

4.1 The Probability of Repetitive Collisions

In the random back-off scheme, the probability for a receiver to suffer from repetitive collisions can be derived from the probability for a collision to be resolved in the first collision resolution process, which can be expressed as:

$$1-\frac{BS}{BS}\frac{BS-1}{BS}\cdots\frac{BS-(2+\varepsilon(d-2)-1)}{BS}=1-\frac{BS!}{BS^{2+\varepsilon(d-2)}(BS-(2+\varepsilon(d-2)))!} \quad (1)$$

where BS denotes the number of back-off slots. This equation represents that the probability of repetitive collisions is proportional to both the number of neighbor nodes and the event rate.

As we have discussed in Section 3, TC can resolve a collision on demand in a single resolution process. However, TC cannot prevent additional packet generations during RBCA. Therefore, another hidden terminal of an on-going packet transmission may cause an additional collision. If the distance between a sender and a receiver is *l*, the potential area of hidden terminals becomes $\sqrt{3}lr$ as shown in **Fig. 2**. Then, the size of average potential area can be given by:

$$\int_0^r \sqrt{3}lrdl = \frac{\sqrt{3}r^2}{2} \tag{2}$$

Therefore, the average number of hidden terminals becomes *d/3* since the area of the communication range is $3\sqrt{3}r^2/2$.

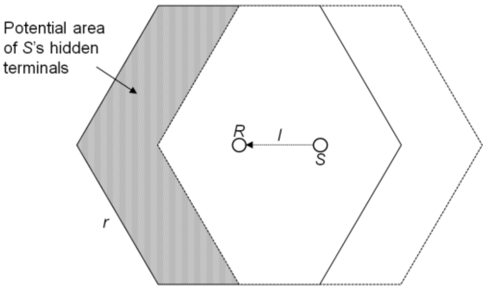

Fig. 2. Potential area of node S's hidden terminals

A hidden terminal will cause an additional collision if it transmits its packet before overhearing ACK. The probability for the hidden terminal to collide with an on-going packet transmission can be expressed by:

$$\frac{T_{DATA}}{(T_{DATA}+T_{ACK})*(2+\varepsilon(d-2))+dT_{slot}/2+T_{Order}} \tag{3}$$

T_{DATA}, T_{ACK}, T_{slot}, and T_{Order} denote the times for transmitting a DATA packet, an ACK packet, a reservation signal, and an order packet respectively. The denominator represents the time for a collision resolution process. Therefore, the probability of repetitive collisions in TC can be given by:

$$1-\left(1-\frac{T_{DATA}}{(T_{DATA}+T_{ACK})*(2+\varepsilon(d-2))+dT_{slot}/2+T_{Order}}\right)^{\frac{\varepsilon d}{3}} \tag{4}$$

Fig. 3 shows the probability of repetitive collisions as we vary the number of neighbor nodes. For this comparison we assume that each node generates 0.1 data packet per second and the initial number of slots for exponential back-off is 30. According to the Equation (1) and (4) we can expect that the probability of repetitive collisions is proportional to the number of neighbor nodes as shown in this figure. Note that the number of neighbor nodes is used as the exponent within the denominator of Equation (1). Therefore, the random back-off scheme is more sensitive to the number of neighbor nodes than TC.

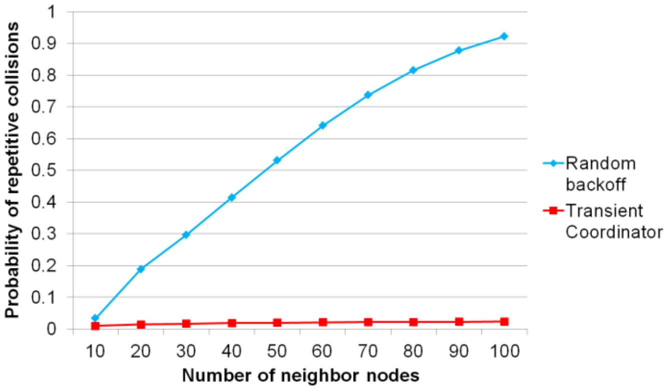

Fig. 3. The probability of repetitive collisions according to the number of neighbor nodes

4.2 Time and Energy Wastes due to Repetitive Collisions

To analyze the time and energy consumption due to repetitive collisions we need to figure out the average number of collided packets. From Equation (1) we can derive the probability for *k* packets to be collided as follows:

$$\frac{BS!}{BS^{2+\varepsilon(d-2)}(BS-(2+\varepsilon(d-2)-k))!} \tag{5}$$

Therefore, in the random back-off scheme the average number of collided packets, $N_{Random}(d)$, is given by:

$$N_{Random}(d)=\sum_{k=2}^{2+\varepsilon(d-2)}\frac{BS!*k}{BS^{2+\varepsilon(d-2)}(BS-(2+\varepsilon(d-2)-k))!} \tag{6}$$

For simplicity, we assume that there is no extra traffic after a collision. Since the random back-off scheme usually exploits an exponential back-off window whose length is $BS*2^{(s-1)}$ where s denotes the s^{th} continuous collision resolution process. Therefore, the average time consumed for the s^{th} resolution process, $Over^{time}_{Random}(s)$, can be given by:

$$Over^{time}_{Random}(s)=T_{DATA}+T_{ACK}+BS*2^{s-1} \tag{7}$$

By using Equation (1) we can calculate the probability of a collision during s^{th} process, $Col_{Random}(s)$, as follows

$$Col_{Random}(s)=1-\frac{(2^{(s-1)}BS)!}{(2^{(s-1)}BS)^{2+\varepsilon(d-2)}((2^{(s-1)}BS)-(2+\varepsilon(d-2)))!} \tag{8}$$

Therefore, the time overhead due to repetitive collisions can be expressed by:

$$\begin{aligned}Over^{time}_{Random} &= (1-Col_{Random}(1))*Over^{time}_{Random}(1)\\ &+Col_{Random}(1)*\left\{\begin{matrix}(1-Col_{Random}(2))*Over^{time}_{Random}(2)+\ldots.\\ +Col_{Random}(N_{Random}(d)-1)*\\ \left[(1-Col_{Random}(N_{Random}(d)))*Over^{time}_{Random}(N_{Random}(d))\right]\end{matrix}\right\}\end{aligned} \tag{9}$$

Note that all the collided senders and a receiver need to keep awake during a collision resolution process. Therefore, the energy consumption, $Over^{energy}_{Random}$, can be calculated by:

$$\begin{aligned} Over_{Random}^{energy} &= P_{Idle} * Over_{Random}^{time} \\ &+ P_{Tx} * \left(\begin{aligned} & N_{Random}(d) \\ & + \sum_{k=2}^{N_{Random}(d)} Col_{Random}(1) * \ldots * Col_{Random}(k) * (N_{Random}(d) - k) \end{aligned} \right) \end{aligned} \tag{10}$$

where P_{Tx} and P_{Idle} denote the power consumed during a transmission state and an idle state respectively.

The probability of a *k*-packet collision due to hidden terminals in TC can be derived from Equation (4) as follows:

$$1-\left(1-\frac{T_{ACK}}{T_{CR}}\right)^{\frac{\varepsilon d}{3}-k}\left(\frac{T_{ACK}}{T_{CR}}\right)^{k} \tag{11}$$

where $T_{CR} = (T_{DATA} + T_{ACK}) * (2 + \varepsilon(d-2)) + dT_{slot}/2 + T_{Order}$.

Then, the average number of collided packets, $N_{TC}(d)$, is given by:

$$N_{TC}(d) = \sum_{k=2}^{\varepsilon d/3} k \left(1-\left(1-\frac{T_{ACK}}{T_{CR}}\right)^{\frac{\varepsilon d}{3}-k}\left(\frac{T_{ACK}}{T_{CR}}\right)^{k} \right) \tag{12}$$

As explained in Section 3, hidden terminals may transmit packets regardless of an on-going transaction and may collide with the original packet that is retransmitted after the time for a single packet transmission time. Therefore, the average time waste due to repetitive collisions, $Over_{TC}^{time}$, is given by:

$$Over_{TC}^{time} = 2T_{DATA} * \left(1-\left(1-\frac{T_{ACK}}{T_{CR}}\right)^{\frac{\varepsilon d}{3}-k}\left(\frac{T_{ACK}}{T_{CR}}\right)^{k} \right) \tag{13}$$

The average additional energy consumption due to repetitive collisions, $Over_{TC}^{energy}$, is the energy required for retransmitting the original data packet.

$$Over_{TC}^{energy} = T_{DATA}(P_{Tx} + P_{Idle}) * \left(1-\left(1-\frac{T_{ACK}}{T_{CR}}\right)^{\frac{\varepsilon d}{3}-k}\left(\frac{T_{ACK}}{T_{CR}}\right)^{k} \right) \tag{14}$$

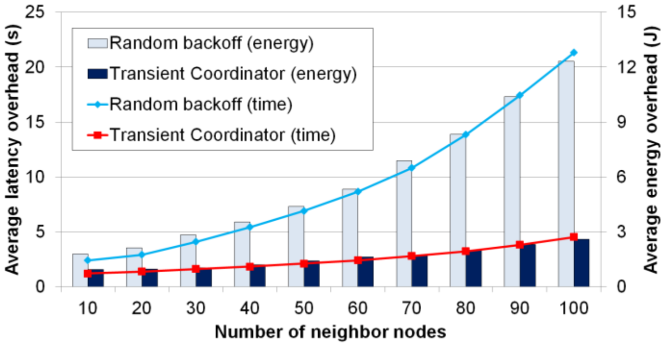

Fig. 4. The average latency and energy overhead according to the number of neighbor nodes

Fig. 4 shows the average unnecessary time and the energy required by the random back-off scheme and TC. We assume that a reception state and an idle state consume the same power.

The average latency overhead represents the extra delay caused by collided packets. The average energy overhead represents the additional energy consumed by collided nodes and their receivers. As shown in this figure, both the average latency and energy overhead of the random back-off scheme have an exponential shape. In contrast, the overheads of TC are mostly linear in shape and the gap between TC and random back-off and TC is more pronounced as we increase the number of neighbor nodes. Considering the probability of repetitive collisions and these overheads, TC can efficiently improve both the network and energy performance by eliminating repetitive collisions.

5. Simulation Results

To evaluate our collision resolution algorithm, we have implemented TC on NS-2 simulator [17]. WiseMAC [5] and RI-MAC [7] are used as underlying asynchronous WSN MAC protocols since they are one of the most representative sender-initiated and the receiver initiated protocols respectively. We compare the performance of the TC-applied versions of both WiseMAC and RI-MAC to those of the baseline WiseMAC and RI-MAC. We assume a random topology where each node generates a 100 byte data packet every 30 seconds. We further assume that the initial number of slots for exponential back-off is 30.

We use four metrics: average throughput per sender, average repetitive collision ratio, average latency overhead, and average energy overhead. The *average throughput per sender* is the number of bits successfully transmitted by each sender, which represents the efficiency of channel usage. The *average repetitive collision ratio* is the ratio of repetitive collisions to all the collisions. This metric inversely represents the degree of collision resolution ability of each protocol. The *average latency overhead* and the *average energy overhead* measure the additional packet latency and the additional energy consumption due to collisions respectively.

5.1 Average Throughput per Sender

Fig. 5 shows the average throughput per sender as we increase the number of neighbor nodes. As shown in this figure, all the protocols suffer from performance degradation as the number of neighbor nodes increases. This is due to the fact that contentions and collisions between multiple senders increase as the number of neighbor nodes increases. Note that, however, the total throughput within a network field may increase since the number of simultaneous transmissions is increased. In addition, the additional performance gain achieved by TC slightly increases as the number of neighbor nodes increases since TC effectively handles collisions due to multiple simultaneous communications and minimizes the idle state of channel around a receiver. When the number of neighbor nodes is 10, TC improves the average throughput by 14.1% and 12.2% compared to WiseMAC and RI-MAC respectively. As we increase the number of neighbor nodes up to 50, the performance gains of TC against WiseMAC and RI-MAC reach 16.3% and 16.1% respectively. Especially, TC marks the highest performance gain, 19.4% compared to WiseMAC, when the number of neighbor nodes is 45. These results confirm that TC can efficiently resolve collisions and thereby mitigates the performance penalty due to repetitive collisions by deriving a collision-free transmission sequence and by assigning exclusive transmission time to each sender.

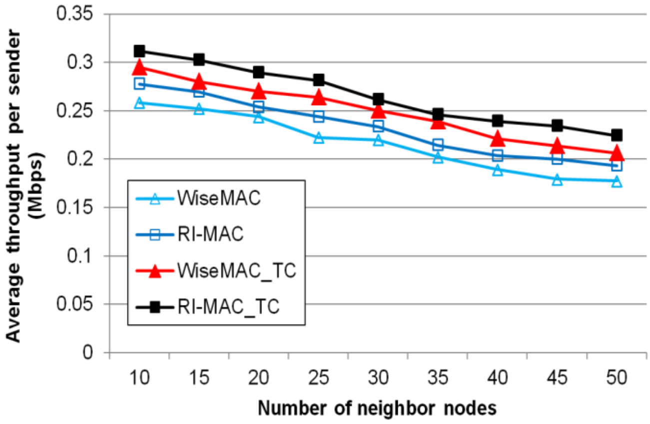

Fig. 5. Average throughputs per sender node

5.2 Average Repetitive Collision Ratio

Fig. 6 shows the average repetitive collision ratio for 4 protocols simulated. As shown in the figure, the repetitive collision ratios of the baseline protocols exponentially increase as the number of neighbor nodes increases. Since the random back-off scheme used in WiseMAC and RI-MAC regards each collision as an independent event, it cannot avoid repetitive collisions. Since only one winner can access the communication medium, the random back-off scheme tends to accumulate packets on a receiver side. These accumulated packets may lead to more serious contentions and collisions as the number of neighbor nodes increases. However, since a receiver coordinates all the collided senders and derives a collision-free transmission sequence, TC can eliminate the possibility of a re-collision among multiple simultaneous packets that are already collided with each other. Nevertheless, the results show that TC versions of the protocols still suffer from continuous collisions. This is due to the fact that a new sender may interfere with an on-going transaction. As shown in the result, RI-MAC is more robust to this problem since a sender always starts to transmit a packet after it receives a beacon from its receiver while in WiseMAC a node can transmit a preamble whenever the node detects an idle channel regardless of the state of its receiver.

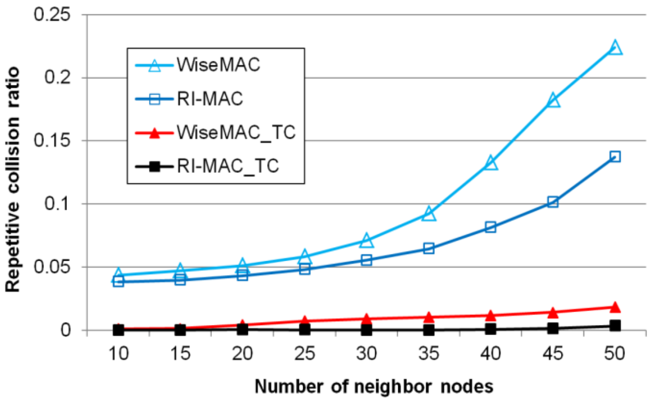

Fig. 6. Average repetitive collision ratios according to the number of neighbor nodes

5.3 Average Latency Overhead

Fig. 7 shows the average latency overhead of 4 protocols as we vary the number of neighbor nodes. As discussed in Section 4, the random back-off scheme exponentially increases buffering delay due to repetitive collisions. According to our simulation results RI-MAC suffers 4.3% more collisions than WiseMAC. Since all the senders to the same receiver start their transmissions at the same time in RI-MAC, the probability of collisions in RI-MAC is generally higher than that in WiseMAC. As shown in the figure, TC can effectively reduce the latency overhead by up to 64.1% and 71.2% compared to WiseMAC and RI-MAC respectively. By removing repetitive collisions as we have seen in **Fig. 6**, TC can reduce the time required for additional collision resolution processes.

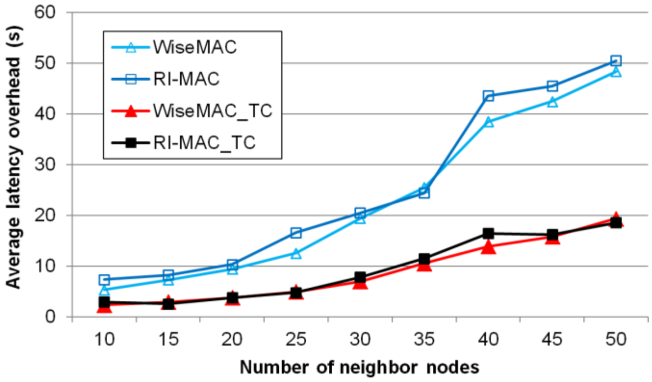

Fig. 7. Average latency overhead according to the number of neighbor nodes

5.4 Average Energy Overhead

As we analyze in Section 4, the energy overhead is proportional to the number of collided packets which have to be retransmitted. **Fig. 8** shows the average energy overhead per sender as we vary the number of neighbor nodes. Since the numbers of collided packets of WiseMAC and RI-MAC exponentially increase according to the number of neighbor nodes, the average energy overhead per sender also exponentially increases. In contrast, TC can reduce the energy overhead by up to 90.3% and 91.1% compared to the baseline WiseMAC and RI-MAC respectively. Since TC does not rely on random back-off, it can minimize the additional energy consumption due to longer idle listening and repetitive collisions caused by the back-off resolution process.

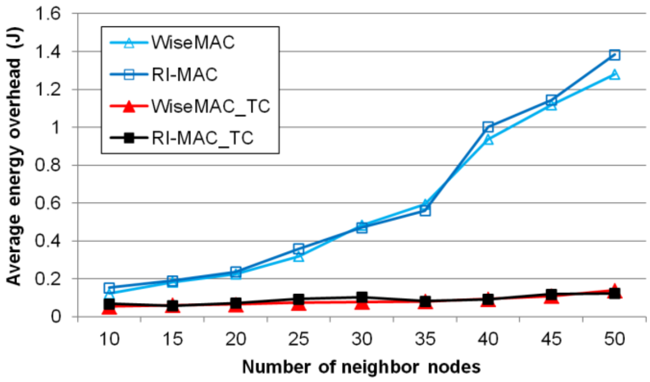

Fig. 8. Average energy overhead according to number of neighbor nodes

6. Conclusion

Conventional asynchronous WSN collision resolution schemes that select a single winner increase buffering delay and accumulate contenders around a sink under the burst and convergent traffic. With a simple tuning of the existing WSN MAC protocols, TC can effectively and efficiently reduce the unnecessary time, bandwidth and energy from repetitive collisions and contentions by coordinating multiple simultaneous transmissions into a collision-free transmission sequence on demand. By allowing each node to convert its operations from a contention based medium access scheme to a reservation based channel allocation on demand TC can efficiently resolve collisions and minimize the time and the energy consumption. According to our simulation results, TC improves the average per-node throughput, the average latency overhead, and the average energy overhead by up to 19.4%, 71.2%, and 91.1% compared to the existing asynchronous WSN MAC protocols respectively. This demonstrates that TC is a more robust and efficient collision resolution solution for asynchronous WSN MAC protocols in terms of network performance, bandwidth utilization, and energy consumption compared to the existing random back-off algorithms.

References

[1] Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen, “Compressive data gathering for large-scale wireless sensor networks,” in *Proc. of the 15th Annual International Conference on Mobile Computing and Networking*, pp. 145-156, 2009. Article (CrossRef Link)

[2] Jang-Ping Sheua, Prasan Kumar Sahoob, Chang-Hsin Suc, and Wei-Kai Huc, “Efficient path planning and data gathering protocols for the wireless sensor network,” *ELSEVIER, Computer Communications*, Vol. 33, Issue 3, pp. 398-408, 2010. Article (CrossRef Link)

[3] K. Ramanan and E. Baburaj, “Data Gathering Algorithms for Wireless Sensor Networks: A Survey,” *AIRCC, International Journal of Ad hoc, Sensor & Ubiquitous Computing*, pp. 102-114, 2010. Article (CrossRef Link)

[4] Joseph Polastre, Jason Hill, and David Culler, “Versatile low power media access for wireless sensor networks,” in *Proc. of the International Conference on Embedded Networked Sensor Systems*, pp. 95-107, 2004. Article (CrossRef Link)

[5] Christian C. Enz, Amre El-Hoiydi, Jean-Dominique Decotignie, and Vincent Peiris, “WiseNET: An Ultra Low-Power Wireless Sensor Network,” *IEEE computer Magazine*, Vol. 37, No. 8, pp. 62-70, 2004. Article (CrossRef Link)

[6] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han, “X-MAC: a Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks,” in *Proc. of the International Conference on Embedded Networked Sensor Systems*, pp. 307-320, 2006. Article (CrossRef Link)

[7] Yanjum Sun, Omer Gurewitz, and David B. Johnson, “RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks,” in *Proc. of the 6th ACM Conference on Embedded Network Sensor Systems*, pp. 1-14, 2008. Article (CrossRef Link)

[8] Dongyu Yang, “RW-MAC: An asynchronous receiver-initiated ultra low power MAC protocol for Wireless Sensor Networks,” in *Proc. of IET International Conference on Wireless Sensor Network*, pp. 393-398, 2010. Article (CrossRef Link)

[9] Wei Ye, Jhon Heidemann, and Deborah Estrin, “Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks,” *IEEE Transactions on Networking*, Vol. 12, Issue. 3, pp. 493-506, 2004. Article (CrossRef Link)

[10] Sang Hoon Lee, Joon Ho Park and Lynn Choi, “AMAC: Traffic-Adaptive Sensor Network MAC Protocol through Variable Duty-Cycle Operations”, in *Proc. of the IEEE International Conference on Communications*, pp. 3259-3264, 2007. Article (CrossRef Link)

[11] Sang Hoon Lee and Lynn Choi, "A+MAC: A Streamlined Variable Duty-Cycle MAC Protocol for Wireless Sensor Networks," in *Proc. of International Conference on Computer and Applications*, 2012
[12] Amin Irandoost, Sara Taheri, and Ali Movaghar, "PL-MAC: ProLonging Network Lifetime with a MAC Layer Approach in Wireless Sensor Networks," in *Proc. of the International Conference on Sensor Technologies and Applications*, pp. 109-114, 2008. Article (CrossRef Link)
[13] IEEE 802.11 WLAN, *IEEE 802.11 working group*, http://www.ieee802.org/11/
[14] C. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks – Architectures and Protocols," *Prentice Hall PTR*, 2004
[15] Lynn Choi, Sang Hoon Lee, and Jong-Arm Jun, "SPEED-MAC: Speedy and Energy Efficient Data Delivery MAC Protocol for Real-Time Sensor Network Applications," in *Proc. of IEEE International Conference on Communications*, pp. 1-6, 2010. Article (CrossRef Link)
[16] C++ Boost Graph Library, http://www.boost.org/libs/graph/doc, accessed 10th May 2012
[17] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns/, accessed 10th May 2012

Sang Hoon Lee received the B.S. and M.S. degrees in computer engineering from Korea University, Seoul, Korea, in 2004 and 2006 respectively. He is preparing Ph.D. degree at Korea University.

Byung Joon Park is currently a professor at the Computer Science Department, Kwangwoon University in Seoul, Korea. His research interests include artificial intelligence and web applications, especially machine Learning and data/web Mining. He received his Ph.D in Computer Science from University of Illinois at Urbana-Champaign, Master's degree in Computer Science from University of Minnesota, Minneapolis, and Bachelor's degree in Computer Engineering from Seoul National University, Korea.

Lynn Choi received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, Korea, in 1986 and 1988 respectively, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, in 1996.

From March 1998 to July 1990, he worked at Korea Telecom Research Center as a Member of Technical Staff. After he received his Ph.D. degree, he joined Intel Corporation as a Senior Design Engineer, participated in the Itanium processor design team, where he worked on the microarchitecture design and validation of the Itanium processor. From September 1998 to August 2000, he was with the Department of Electrical and Computer Engineering at the University of California, Irvine as a tenure track faculty member. Since September 2000, he joined the School of Electrical Engineering, where he is currently a Professor.

Prof. Choi has served on several program committees and editorial boards, and published more than 40 international conference papers and journals in the areas of computer architecture, compilers, internet security, and computer networks. His current research interests include algorithm and architecture issues in application-specific processors, sensor networks and applications, and processor architectures for secure computing. He is a member of both IEEE and ACM.