Efficient and Adaptive Resource Allocation Scheme for Self-Organizing DESYNC TDMA Network

Bongsoo Roh, Dongwon Kum, Taehyun Kim, Jeungwon Choi
The 2nd R&D Institute
Agency for Defense Development, Daejeon, Republic of Korea
{saintroh,kdw80,navycos96,jwchoi}@add.re.kr

Kwangsoo Kim, Byeong-hee Roh
Department of Computer Engineering
Ajou Univ.,Suwon, Republic of Korea
{zublian,bhroh}@ajou.ac.kr

ABSTRACT

DESYNC TDMA, based on biologically inspired primitive[1], provides excellent total throughput with a low implementation complexity in the under high traffic loads without using the global clock synchronization and centralized coordination. However, DESYNC TDMA has the inefficient time resource scheduling because all participating nodes always fairly share the time resources for accessing the transmission medium in a self-organized and decentralized manner without considering whether a node has a transmission data or not. Therefore, this paper addresses an Efficient and Adaptive Resource Allocation Scheme for Self-Organizing DESYNC TDMA Network, called EA-DESYNC, which improve the inefficient time resource scheduling of DESYNC TDMA. EA-DESYNC uses the mini-time slot to distinguish between the actual use time slots and the unused time slot of DESYNC's transmission time slot. Also, this scheme proposes a method that can be shared with the transmission medium according to the residual amount of its mini-time slot and 1-hop neighbors. Simulation results show that EA-DESYNC could outperform the DESYNC in terms of throughput.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Networks Architecture and Design – *Distributed Networks, Network Communication, Wireless Communication*

Keywords

DESYNC, MAC, TDMA

1. INTRODUCTION

Medium Access Control (MAC) protocol plays a key role in optimally utilizing the shared transmission medium which directly impacts on overall network performance. Recently, in the IEEE 802.11 and 802.15.4/ZigBee applied to various fields of application, MAC reliability and tight QoS requirement for the timely delivered of data has emerged increasingly as an important technology issue. Especially, in the fully-decentralized network, Research about MAC protocols performing efficient resource allocation based on self-organizing has been actively studied [2][3][4][5][6].

CSMA/CA (Carrier Sense Multiple Access/Collision Avoid), a well-known MAC protocol in the distributed network such as Ad-Hoc Network, is typically robust to topology changes. However, its performance can be significantly degraded under high contentions due to high overhead accrued for resolving collisions. In contrast, Schedule-based protocols like Time Division Multiple Access (TDMA) solves the above mentioned disadvantage of CSMA/CA by utilizing synchrony among neighboring nodes to achieve collision-free transmission by assigning transmission time slots to individual nodes.

DESYNC TDMA[2], one of most promising TDMA approaches for the distributed network, is based on the biologically inspired primitive, which has a low implementation complexity. In addition, it provides scheduling for accessing the shared transmission medium between nodes autonomously through the distributed manner.

In DESYNC TDMA, without the need of global clock synchronization and centralized coordination, all participating nodes can always fairly share the time resource for accessing the shared transmission medium through firing message.

However, DESYNC TDMA exhibits inefficient throughput as all nodes have equally the fixed-transmission time slots that cannot adapt to dynamic variations in traffic load.

In this paper, we present Efficient and Adaptive Resource Allocation Scheme for Self-Organizing DESYNC TDMA Network, called EA-DESYNC, which improve the weakness of DESYNC TDMA that cannot adapt to dynamic variations in traffic load. EA-DESYNC uses the mini-time slot to distinguish between the actual use time slots and the unused time slot of DESYNC′s transmission time slot.

Also, this scheme proposes a method that can be shared with the transmission medium according to the residual amount of its mini-time slot and 1-hop neighbors. Simulation results show that EA-DESYNC could outperform the DESYNC in terms of throughput.

2. DESYNC TDMA Algorithm

Desynchronization is a biologically inspired primitive for the equidistant distribution of oscillators, e.g periodically transmitting sensor nodes. So, The network is composed of a set of nodes N.

All communication links are symmetric and fully-connected network. Each node has a unique identifier $i (i \in N)$ and oscillates at an identical frequency.

The phase φ_i of node i denotes the elapsed time since its last transmission relative to its current period. When a node finishes its period, it broadcasts a so called firing packet and immediately resets its phase. An update of the system state is only required each time a node broadcasts its firing packet.

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257934

• node p(i) broadcasts its firing packet just before node i and thus is called previous phase neighbor($\varphi_{pre(i)}$)

• node s(i) broadcasts its firing packet just after node i and thus is called successive phase neighbor($\varphi_{succ(i)}$)

To determine a more appropriate firing phase $\varphi_i{}'$ according to an equidistant distribution, node i must first calculate the midpoint of its phase neighbors using the delta phase

$$\Delta_{i,j} = \varphi_j - \varphi_i$$

The firing times of the previous and next neighbors are recorded relative to node i's firing as Δ_{i+1} and Δ_i, respectively. In this way, node i can approximate the phases of its previous and next phase neighbors as $\varphi_{i+1}(t) = \varphi_i(t) + \Delta_{i+1}(\text{mod } 1)$ and $\varphi_{i-1}(t) = \varphi_i(t) + \Delta_i$ (mod 1). Using this information, node i can calculate the midpoint of its neighbors:

$$\text{mid}(\varphi_{pre(i)}, \varphi_{succ(i)}) = \varphi_i + \frac{(\Delta_{i,pre(i)} - \Delta_{succ(i),i})}{2} = \frac{(\varphi_{pre(i)} - \varphi_{succ(i)})}{2}$$

Finally, the new firing phase $\varphi_i{}'$ of node i can be estimated by itself as

$$\varphi_i{}' = (1-\alpha)\cdot\varphi_i + \alpha\cdot\text{mid}(\varphi_{pre(i)}, \varphi_{succ(i)})$$

The jump size parameter $\alpha \in (0.0, 1.0)$ regulates, how fast a node moves toward the assumed midpoint of its phase neighbors.

Convergence to the stable state of desynchrony is achieved, if each node has the same distance to its phase neighbors(fig 1.) and thus the transmission times do not change anymore.

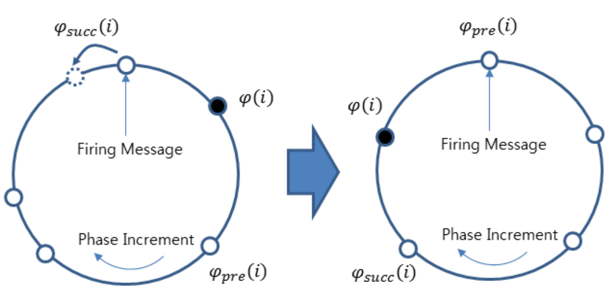

Figure 1. Snapshots of desynchronization process

3. Proposed EA-DESYNC TDMA Algorithm

EA-DESYNC minimizes the waste of unused timeslots and proposes distributed scheduling strategy that enables maximum 2 times flexible control of the resource depending on actual traffic usage of available nodes.(fig 2.)

EA-DESYNC not only allows flexible resource sharing among adjacent nodes but also considers maintaining the features of distributive scheduling and desynchronicity which corresponds to primitives of original DESYNC.

Another way to increase the resource occupancy, we can consider the method which defines a node has no data to send as idle state. However, this way must re-enter the network each time the message occurs, and has the problem that occurs substantial inefficient bandwidth usage due to the DESYNC TDMA has the cost by the temporary desynchronization.

In this paper, regardless of the presence or absence of the transmission data, we maintain the participation state at the network and continue the notification to adjacent nodes using the modified firing message. Also, we increase the efficiency of resource-sharing by sharing unused bandwidth.

3.1 Mini Time Slot

Time slot of original DESYNC is defined as time interval that determines proposed unit for the time slot of each node evenly with φ_{mid} value. In distributed environment, in order to share remaining time interval allocated to each node with adjacent nodes, the concept of quantitative criteria for remaining time resources is required.

EA-DESYNC establishes the quantitative criteria for remaining time resource sharing, and it applies the concept of mini time slot for efficient time resource management. The mini time slot specifies T/N logically, time slot assigned to the existing node, to distinguish remainder except for actual using amount of data as the remaining time resource.

The number of mini time slots (MTS) is divided logically by node itself to share its time resources. Thus, it has only fewer influences than DESYNC algorithm in terms of the time synchronization.

The smaller the size of the mini time slot is, the more precise time resource allocation is possible. However, it involves further consideration of hardware requirements for ensuring accurate time synchronization.

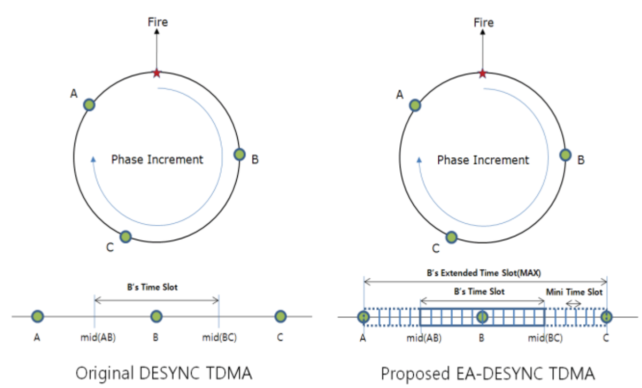

Figure 2. Proposed EA-DESYNC TDMA

3.2 Modified Firing Message

In addition to MAC time-stamp message for TDMA frame synchronization, EA-DESYNC includes information of 'Available number of Mini Time Slot($AMTS_{prev}$, $AMTS_{succ}$)', 'Steady State Parameter Threshold(SSP_{thresh})'by employing modified firing message. (Table 1.)

Table 1. Modified Firing Message Information

Node_ID	Slot_Order	Msg_Time	SSPthresh	AMTSprev	AMTSsucc

- Node_ID : Node Identifier

- Slot_Order : TDMA slot sequence information of each node.
- Msg_Time : MAC Time Stamp for TDMA frame synchronization.
- SSP_{thresh} : Preset convergence threshold value for the convergence state determination.
- $AMTS_{prev}$: Usable slot of previous phase neighbor direction on the basis of own φ_i
- $AMTS_{next}$: Usable slot of next phase neighbor direction on the basis of own φ_i

3.3 Steady State Parameter

When new node join or leave, it will happen the fluctuation interval for adjusting the time phase difference because of the characteristic of DESYNC. This interval adjusts the speed of the phase difference according to α that is phase-coupling constant about the scale of movement of node i from current phase to desired midpoint. (α∈(1,0). α is generally set 0.95)

Because of unstable fluctuation period on DESYNC TDMA, this period cannot be shared to inter-node time slots. In this paper, we use the steady state parameter threshold(SSP_{thresh}), previously proposed a similar equation in [5].

However, it is possible for SSP_{thresh} to obtain a more accurate steady-state by applying the phase difference between the mini time slots.

$$\left|\varphi_{\mathrm{curr}}^{(\mathrm{MTS}_p)} - \varphi_{\mathrm{before}}^{(\mathrm{MTS}_p)}\right| \le \mathrm{SSP}_{\mathrm{thresh}}$$

The SSPthresh value of EA-DESYNC is a criterion for the difference between phases of p-th mini time slots of current and previous cycle. That is, a node determines whether its adjacent nodes enter into the steady-state or not, which is a prior condition for sharing time slot.

3.4 Pseudo Algorithm for EA-DESYNC

Pseudo code of EA-DESYNC TDMA

```
On_Receive_Firing_Message(Node ID, Slot_Order, Msg_time,
AMTS_prev, AMTS_succ, SSPthresh)
{    last_fire=msg_time
       if(SSPthresh==TRUE and just_fired ==TRUE)
       if(BurstyTraffic==FALSE)
          slot_start=T+(prev_fire+my_fire)/2
          slot_end=T+(next_fire+my_fire)/2
       if (BurstyTraffic==TRUE)
          slot_start=T+(prev_fire+my_fire)/2+AMTS_prev
          slot_end=T+(next_fire+my_fire)/2+AMTS_next
      endif
      my_adjust_fire_time=1+(1-alpha)*my_fire+
alpha*(prev_fire+next_fire)/2
}
```

4. Simulation Results

In this section we experimentally measure the performance of EA-DESYNC relative to original DESYNC-TDMA[2].We use OPNET modeler as the simulation tool.

The simulation environment is given as following parameters. There are 10 nodes in the network and randomly start their firing. We assume all nodes are randomly deployed in 1km ×1km with 1-hop transmission range. (i.e. Its propagation delay is negligible.)

The period of a cycle (superframe length) is set by 1 second. We plot the firing times and their packet transmissions during simulation in Fig. 3. Each line represents the firing time of each node relative to an arbitrarily chosen node. The dots are packet transmission of the node to show node's slot time.

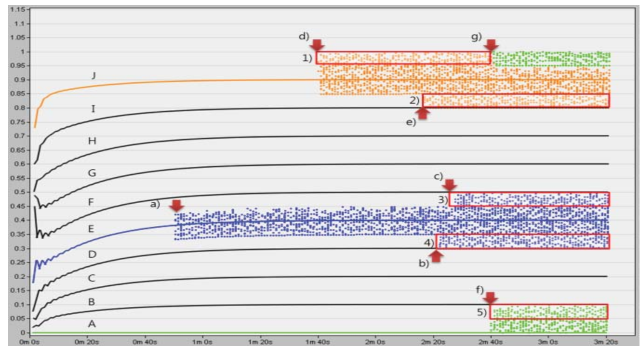

Figure 3. The Firing times and packet transmissions under EA-DESYNC

We generated 3 traffic flows (node A, E, and J) to show our algorithm intuitively, and the traffic is starts 50 second, 100 second, and 160 second, respectively. Each packet arrives at exponential distribution with $\mu = 0.01$ second and their packet size is set by exponential distribution with $\mu = 8192$ bit. The data rate of the channel is given by 1,000,000 bps.

Firstly, a) node E starts to transmit packets within its slot time range since D and F are not stable, thus E cannot use time slots of D and F additionally. Then, b) E can additionally transmit using slots of D (denoted by box 4) since D becomes stable and has no data to send. Similarly, c) F becomes stable and has no data to send, and then E can send its data using slots of F (denoted by box 3).

Next traffic is generated at 100 second; d) J starts to transmit its packets and it can use more slots of A (denoted by d) because A is stable and has no data to send. Also J can use more slots at e) because node I becomes stable and has no data to send. However, f) A starts to transmit packets at 160 second, A can use slots of B with same reason and J should return slots of A at g). Compare to the normal DESYNC case; the red boxes 1) to 5) are additionally achieved in the EA-DESYNC.

The result shows the firings are globally desynchronized at approximately 3m20s (200 rounds).

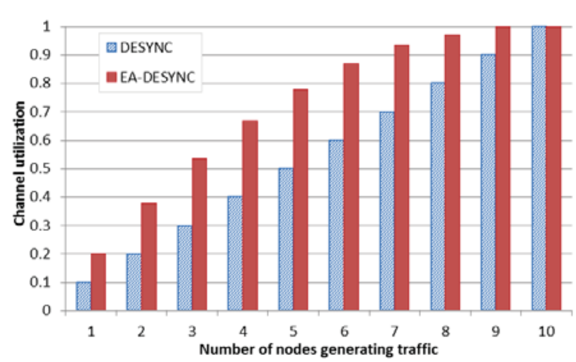

Figure 4. Channel utilization: EA-DESYNC vs DESYNC

The channel utilization is shown in Fig. 4. to compare the efficiency of time resource allocations between proposing EA-DESYNC and original DESYNC directly. There are 10 nodes in the network with 1-hop range. The figure shows channel utilization according to the number of nodes generating traffic is given to 1 to 10 nodes.

We observed that the EA-DESYNC improve the channel utilization performance noticeably than original DESYNC because the original one uses slots evenly but our algorithm uses slots of adjacent node additionally when the adjacent node has less packet to send.

5. ACKNOWLEDGMENTS

This work was done at Agency for Defense Development.(ADD-IBR-245)

6. REFERENCES

[1] Renato E. Mirollo and Steven H. Strogatz. “Synchronization of Pulse-Coupled Biological Oscillators,” *SIAM Journal on Applied Mathmatics*, 50(60):1645-1662, 1990

[2] J. Degesys, I. Rose, A. Patel, and R. Nagpal, “DESYNC: Self-Organizing Desynchronization and TDMA on wireless sensor networks,” In IPSN , pp. 11-20. April 2007.

[3] A. Tyrrell, G. Auer, and C. Bettstetter, "Fireflies as role models for synchronization in ad hoc networks," in Proc. of 1st Bio-Inspired Models of Network, Information, and Computing Systems 2006, (BIONETICS), Madonna di Campiglio, Dec 2006, pp. 1-7.

[4] H. Kang and J. Wong, “M-desync: A multi-hop desynchronization for wireless sensor networks,” Stony Brook University, Embedded Systems, Optimization Lab, Tech. Rep., 2008.

[5] Dujdow Buranapanichkit and Yiannis Andreopoulos, “Distributed Time-Frequency Division Multiple Access Protocol for Wireless Sensor Networks,” IEEE wireless communication letters, 2012

[6] A. Mutazono, M. Sugano, and M. Murata, “Evaluation of robustness in time synchronization for sensor networks,” in Proceedings of the 2nd International Conferenece on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS 2007), pp. 89–92, Dec. 2007