

Bandwidth Optimization in 802.15.4 Networks through Evolutionary Slot Assignment

Vidya KRISHNAMURTHY, Edward SAZONOV

*Department of Electrical and Computer Engineering, Wallace H. Coulter School of Engineering
Clarkson University, NY, USA
Email: esazonov@clarkson.edu*

Received March 26, 2009; revised May 15, 2009; accepted July 29, 2009

ABSTRACT

Traditional Wireless Sensor Networks (WSNs) based on carrier sense methods for channel access suffer from reduced bandwidth utilization, increase energy consumptions and latency problems in networks with high traffic. In this work, a novel Evolutionary Slot Assignment (ESA) algorithm has been developed to increase the throughput of large wireless mesh networks with no centralized controller. In the presented scheme, the sensor nodes self-adapt to the traffic patterns of the network by selecting transmission slots using evolutionary learning methods. Each sensor node evolves an independent transmission schedule. Unlike traditional evolutionary methods, fitness evaluation of every node impacts fitness of every other sensor node in the network. The ESA algorithm has been simulated using Network Simulator-2 and compared with the IEEE 802.15.4 CSMA-CA, a Static Slot Assignment (SSA) and a Random Slot Assignment schemes (RSA). Results show a remarkable improvement in the network throughput using the proposed ESA method as opposed to other compared methods.

Keywords: CSMA-CA, 802.15.4, Sensor Networks, Evolutionary Algorithms, Bandwidth Optimization

1. Introduction

Wireless Sensor Networks (WSNs) consist of a group of sensors nodes that use wireless links to perform distributed sensing tasks. Sensor nodes combine simple wireless communication, minimal computational facilities, and sensing of the physical environment for an application-specific sensor network [1]. In monitoring applications of Wireless Sensor Networks (WSNs), multiple nodes may desire to transmit at the same time, for example in response to an event detected by multiple sensor nodes. Same time transmissions by multiple nodes lead to RF collisions that will cause loss of packets. There are multiple methods for collision detection and avoidance. The simplest method developed in the 1970s is ALOHA, where, a node with a packet to transmit waits for a random amount of time and then sends the packet. Another variation of this method was the slotted ALOHA, where time was divided into slots and nodes wait for the start of a slot for transmission. In case of a collision, the node simply retransmits the packet. These methods failed with increasing network size and data rate [2]. FDMA (Frequency Division Multiple Access) and TDMA (Time

Division Multiple Access) are methods developed and used in 1980s and early 90s that divide the available frequency spectrum or time into slots and assign them to different nodes. These methods divide the available resources amongst sensor nodes and therefore are not very cost effective and cannot deal with rapidly changing network topology, such as in mobile nodes [3]. Traditional WSNs are based on collision free carrier sense methods for channel access. The most commonly used scheme by the late 1990s is a Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) where the nodes sense the channel to be idle and wait for a random amount of time before transmission. This restricts the collision probability to a situation that two nodes may have a packet at the same time to transmit and the random wait time for them happens to be same.

IEEE 802.11 emerged as a popular choice for WSNs that use CSMA-CA with its four-way handshaking protocol [4] to increase the reliability of data transfer. But this protocol and related hardware are too energy consuming for low-data rate, low-power networks and energy scavenging applications. CSMA-CA mechanism also affects the network latency [5] thus making it unfit

for monitoring applications with large number of sensor nodes. The IEEE 802.15.4 protocol [6] has recently emerged as a new standard for low rate wireless personal area networks (LR-PANs) [7]. The popularity of the low-power, low-rate IEEE 802.15.4 protocol and availability of low-cost, low-power RF chips has led us to use it for developing low cost Wireless Sensor Networks targeted towards monitoring applications, such as applications of structural health monitoring. However, for synchronous sensor networks that sample data at the same time or respond to events, the packet collision probability is seen to tremendously increase with increase in network size [8]. Such collisions tremendously reduce bandwidth utilization, increase energy consumptions and latency of IEEE 802.15.4 devices. The poor performance of IEEE 802.15.4 based CSMA/CA have also been analyzed using discrete time Markov chain models [9].

We performed preliminary analysis of a multi-hop network to illustrate quality loss of performance in a high-traffic 802.15.4 CSMA-CA network. Network Simulator-2 (NS-2) [10] with IEEE 802.15.4 medium access protocol [11] has been used. The network topology consisted of a beacon-less mesh network with one PAN Coordinator and 30 sensor nodes placed at random in a 50m-by-50m area (Figure 1). The receiving threshold of every sensor node was limited to 15m. The input data rate per node was varied and the total network throughput was measured as the total number of packets received at the receiver (sink) per second. Figure 2 shows the result of the simulation that clearly suggests low bandwidth utilization. A possible solution that can alleviate the problems faced by IEEE 802.15.4 CSMA-CA based sensor network is a reservation based mechanism where the contention times of each wireless node should be separated from each other, thus minimizing collisions and retransmissions. A number of mechanisms in this direction have been proposed and developed [12–17].

Our earlier work [18] shows that a reservation scheduler could provide up to 5 times increase in data rate and 99% usage of effectively available bandwidth. The TDMA scheduler works very well in a single-hop star topology where the central node is aware of all the sensor nodes in the network. But in a typical wireless mesh network with large number of sources and less number of sinks, a centralized scheduling is undesirable. Also, in case of fluctuation in the topology, that is, if sensor nodes enter and leave the network continuously, a static scheduling would not work well. In this paper, we propose a new algorithm, the Evolutionary Slot Assignment (ESA) algorithm that continuously adapts transmission schedule of every sensor node without relying on a central coordinator. The proposed Medium Access Control

(MAC) layer method is inspired by the evolutionary algorithms [19–27]. However, the proposed ESA algorithm is based on a novel model of specimen interactions where fitness evaluation of each individual affects fitness of other nodes in the network. Unlike other de-synchronized algorithms, ESA does not pose assumptions on topology and connectivity of the network. ESA algorithm does not need a common time reference or beacons and is well suited for mesh networks and networks with high node mobility and substantially increases bandwidth utilization of IEEE 802.15.4 networks.

The next section discusses the proposed method in detail. The simulation scheme has been outlined in Section 3 and Section 4 presents the results and discussion of the simulated model. Section 5 concludes the paper with remarks about the method.

2. Methods

To increase the bandwidth utilization and decrease collisions in ad-hoc beaconless wireless sensor networks, a probabilistic learning-based scheduling algorithm based on Evolutionary Slot Assignment (ESA) is proposed. We

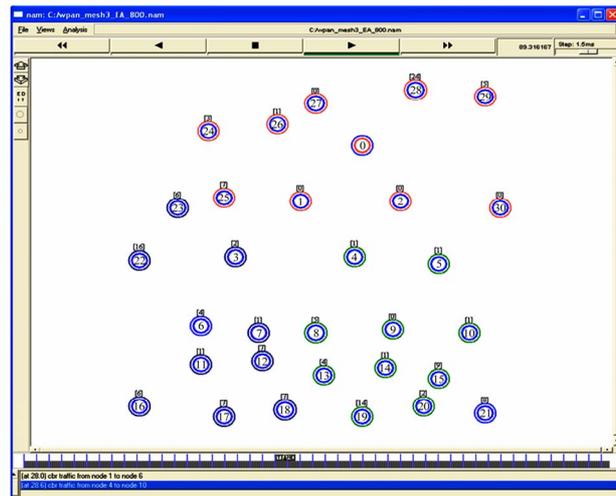


Figure 1. Tested mesh topology.

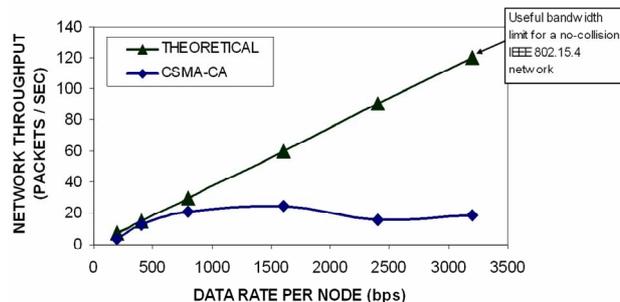


Figure 2. Throughput of a multi-hop mesh network with 31 nodes with increasing load.

consider a beaconless IEEE 802.15.4 network with high data rates characteristic for continuously monitoring applications as our main target. The nodes in the network are not synchronized in time. The ESA algorithm is implemented at the sensor node level and requires no central coordinator or cluster head; does not rely on beacons or common time reference and does not pose assumptions on topology and connectivity of the network.

Each node in the network schedules slots for periodic transmissions within a time period called the network period, or T . The network period T is divided into virtual slots where the sensor node attempts to transmit data. These transmission slots and network period T should not be confused with the Guaranteed Time Slots and beacon interval in a beacon-enabled IEEE 802.15.4 network. In ESA, duration of a transmission slot is determined by the maximum size of data packets with each slot consisting of 8 to 20 backoff periods corresponding

1) **initialize probability vectors:**

$$\begin{aligned} \text{a. } V_T &= \{V_T^1, V_T^2, \dots, V_T^N\} \\ V_T^i &\in \{0, 1\} \\ \text{b. } P_T &= \{P_T^1, P_T^2, \dots, P_T^N\} \\ P_T^i &\in [0, 1] \end{aligned} \quad ,$$

2) **if packet to transmit at time t :**

$$\begin{aligned} \text{a. } \textit{find } i: V_T^i &= 1 \text{ and } i > t \\ \text{b. } \textit{packet_wait_timer} &= t - i; \\ \text{c. } \textit{if packet_wait_timer} &== 0 \\ \text{i. } \textit{transmit packet; } \alpha &= \text{rand}(0, 0.2) \\ \text{ii. } \textit{if status} &== \textit{success} \\ P_T^i &= P_T^i + |\alpha|, \text{ if } P_T^i > 1, P_T^i = 1 \\ \text{iii. } \textit{elseif status} &== \textit{failure} \\ P_T^i &= P_T^i + \alpha, \text{ if } P_T^i < 0, P_T^i = 0, \text{ if } \end{aligned}$$

$$P_T^i > 1, P_T^i = 1$$

$$\text{iv. } \textit{elseif status} == \textit{medium_busy}$$

$$P_T^i = P_T^i - |\alpha|, \text{ if } P_T^i < 0, P_T^i = 0$$

3) **if $P_T^i < \beta$, then**

$$\begin{aligned} \text{a. } V_T^i &= 0 \\ \text{b. } j &= \text{random number } \in \{1, 2, \dots, N\}; \\ V_T^j &= 1 \end{aligned}$$

4) **adaptation:**

$$\text{a. } \delta = \left(\sum_{i=T-9}^T \delta_i \right) / 10$$

Figure 3. ESA Algorithm.

to minimal and maximal possible packet lengths. Network period T defines the maximum number of transmission slots that can be scheduled and the length of the scheduling table. For the purpose of simplicity of analysis, the network period is kept constant for all the sensor nodes in the network and it is assumed that the network period starts on node power-up. Since sensor nodes start-up at random times, the beginning of a network period is not synchronized between sensors. However, since every sensor will evolve an independent schedule without knowing schedules of other nodes, the algorithm does not need a common time marker or a synchronizing beacon. All sensor nodes are assumed to have the same sampling rate and thus have the same average data rate. This represents heavy load conditions of a real-time multipoint continuous monitoring application. The sensor nodes may send data using constant bit rate (CBR) traffic or an exponential (Poisson) traffic pattern. The number of slots (N) each node would need to transmit packets of size (χ bytes) at an average data rate of γ kbps is calculated for the time T as:

$$N = \frac{T * \gamma}{\chi} \quad (1)$$

The ESA algorithm for evolution of transmission schedules (Figure 3) is continuously executed on every node in the network. The algorithm is divided into the following phases:

1) Initialization Phase: In this phase, the internal variables of the algorithm are initialized by each node. A binary slot vector V_T is created of size equal to the total number of transmission slots (N) possible in T .

$$V_T = \{V_T^1, V_T^2, \dots, V_T^N\}, V_T^i \in \{0, 1\} \dots \quad (2)$$

A $V_T^i = 1$ indicates a slot in which the node will attempt to transmit.

Another vector P_T contains the probability associated with every transmission slot (or fitness of a slot):

$$P_T = \{P_T^1, P_T^2, \dots, P_T^N\}, P_T^i \in [0, 1] \quad (3)$$

During initialization, all values of V_T are initialized to 0 and of P_T to 0.5. Due to periodic nature of the network period T both vector V_T and P_T are ring structures (Figure 4), i.e. the algorithm traverses from the last element into the first element of the vectors. Ring representation allows to eliminate the need for common time reference since slot m of a wireless node l will always correspond to slot n of wireless node i as long as relative time drift between the nodes is close to zero (a safe assumption for most applications). The initial set of ($N * r$) slots is selected from the list of slots using a tournament selection algorithm [28]. Specifically, a random number of slots is drawn from the list of available slots and a single slot j with the highest probability P_T^j is selected from the set and V_T is updated with $V_T^j = 1$. The tournament selection

is repeated till $N*r$ slots are assigned. During the initial selection procedure, however, the probability of all the slots is equal and is hence not critical. The redundancy factor r is needed for any retransmissions due to collisions. An example of the vectors V_T and P_T updated with the initial selected slots is:

$$V_T = \{0, 0, 1, 0, \dots, 0, 1, 1, 0, \dots, 0, 0\}, P_T = \{0.5, 0.5, 0.5, 0.5, \dots, 0.5, 0.5, 0.5, 0.5, \dots, 0.5, 0.5\}$$

2) Transmission Phase: Whenever a node has a packet to transmit, it traverses the vector V_T for the next available slot i ($V_T^i = 1$) and sets *packet_wait_timer* to wait until beginning of slot i to transmits the packet. After *packet_wait_timer* expires the node attempts an IEEE 802.15.4 compliant CSMA-CA transmission with acknowledgement and updates vector P_T based on the outcome of the transmission as:

a) Success: If the packet transmission in slot i is successful (acknowledgement received) the probability value P_T^i is increased by the absolute value of a random Gaussian integer α (mean 0, deviation 0.2): $P_T^i = P_T^i + |\alpha|$. If the results is greater than 1, then $P_T^i = 1$. The increase in slot's probability increases the fitness measure of the slot in reward for a successful transmission. Maintaining P_T^i above the threshold β is required for continuing use of slot i in transmission schedule.

b) Failure: If the packet transmission in slot x fails (no acknowledgement),

$$P_T^i = P_T^i + \alpha \dots \quad (4)$$

For a Gaussian α with zero mean, this operation equiprobably increases or decreases the fitness of slot i . However, if $P_T^i + \alpha < 0$, $P_T^i = 0$ or if $P_T^i + \alpha > 1$, $P_T^i = 1$. The goal of this fitness (probability) adjustment

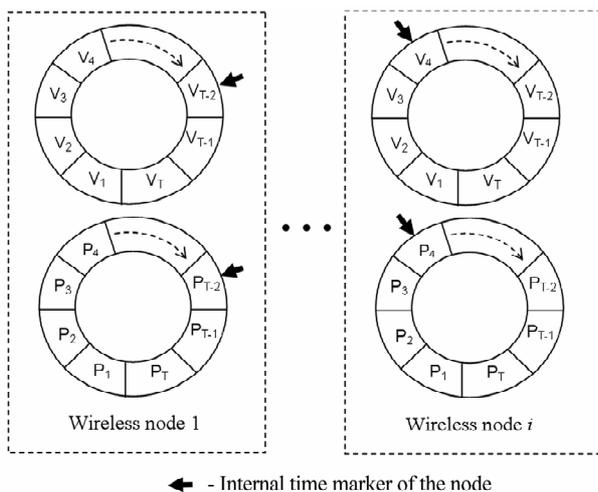


Figure 4. Ring structure of P_T and V_T vectors permits asynchronous operation of wireless.

is to move the transmission probability of nodes competing for access in time slot i in the opposite directions. The optimal outcome for two competing nodes is when the probability of the slot in one node is increased and probability of a matching slot in another node is decreased. Using random Gaussian variable as described will produce such an outcome in 50% of failures.

c) Medium Busy: If the medium is busy when the packet transmission is attempted in slot i , P_T^i is decremented by the absolute value of α : $P_T^i = P_T^i - |\alpha|$. However, if $P_T^i - |\alpha| < 0$, $P_T^i = 0$. The probability is decreased due to the fact that some other node is using the medium at this time. Repeated observation of busy medium will cause P_T^i falling below the threshold β and deselection of the slot i .

3) Selection Phase: The probability (fitness) of each transmission slot marked for transmission in V_T is adjusted in transmission phase. Based on those adjustments, corresponding values of P_T^i may fall below the threshold β , indicating that transmissions repeatedly fail in slot i . The re-selection of slots is performed after completion of every network period (T) for those slots whose probability falls below a certain threshold β . This is done using the tournament selection method. For a slot i with $P_T^i < \beta$, V_T^i set to 0. A random number of slots are drawn from the list of unused slots and the slot j with of the highest probability value P_T^j among these slots is selected. This procedure is repeated for all slots that need rescheduling (such slots are excluded from the pool from which new slots are drawn).

4) Adaptation Phase: Being a part of a multi-hop mesh network, a node may receive packets that need to be forwarded. Therefore, additional slots for forwarding traffic need to be added to the transmission schedule every interval T . Since a node makes no assumptions of topology of the network and ESA algorithm does not use information from upper layers (routing for example), the number of forwarding slots δ is estimated for by monitoring the total number of forwarding packets received by a node over each time period T . The value of δ is computed and updated every network period based on a floating average of δ for the last 10 network periods. Thus the total number of slots available for a node in the interval T is $(N*r + \delta)$. Selection of δ additional slots in the vector V_T is performed by tournament selection. After adaptation phase the node transitions into transmission phase.

This evolutionary process is performed on a continuous basis and adapts the node to the network traffic conditions. It should also be noted that the ESA algorithm is not a classical evolutionary algorithm. Traditional evolutionary algorithms (genetic algorithms, evolution strategies, etc.) evolve a population of individuals but fitness

of any given individual does not impact fitness of other individuals in the population. In ESA fitness is interpreted as transmission probability associated with a slot. A node evaluates fitness of a slot by attempting a transmission and thus actively impacting fitness of matching slots in schedules of all other nodes that may attempt a transmission at the same time. In this sense the ESA algorithm is better characterized as a competitive co-evolutionary algorithm.

To illustrate the gain in bandwidth utilization due to evolutionary component of the algorithm, we propose two simplified variants of the above Evolutionary Slot Assignment (ESA) and compare them with the regular CSMA-CA mechanism. First is a Static Slot Assignment (SSA) technique where both the evolutionary mechanism and randomness of slot reselection are eliminated. In SSA the slot vector V_T is initialed to $(N*r + \delta)$ slots. The data packets are only transmitted in this pre-selected random slot schedule and P_T vector is never updated. Second is Random Slot Assignment (RSA) algorithm in which the evolutionary component is eliminated but randomness of slot reselection is maintained through periodic re-initialization of the transmission schedule V_T every period for $(\eta*r + \delta)$ slots. Comparison with SSA and RSA allows evaluating the performance gain attributed to evolutionary learning. We also perform comparison to standard CSMA-CA technique to illustrate performance gain of ESA.

3. Simulation Scheme

Network Simulator-2 (NS-2) [10] with IEEE 802.15.4 medium access protocol [6] has been used to simulate ESA, SSA and RSA methods. The simulation scenario consists of a beaconless network with one PAN Coordinator and 30 nodes placed at random in a 50m-by-50m area (Figure1) with range of each node limited to 15m. The nodes utilize the IEEE 802.15.4 medium access and physical layers for transmissions and receptions. The nodes are not synchronized in time and start-up at random times. The ESA algorithm was implemented at the SSCS (Service Specific Convergence Sublayer) that acts as an interface between the MAC and the upper layers. The SSCS layer is an implementation specific module that provides access to the MAC primitives and allows for their modification for any specific application. A wireless channel with two-ray ground propagation model was used. Each node is chosen to have omni-directional antenna and the maximum number of packets allowed in the interface queue was set to 10. The packets were transmitted using AODV routing protocol [29].

One of the nodes (Node 0) is assigned as the PAN Coordinator. This node allows all other nodes to join the network through the regular association procedure of the IEEE 802.15.4 standard [6]. This node is also used as the

sink node. However, it is no different from any other node in functionalities and other nodes may be sinks too. The nodes formed a star network (single-hop, single receiver) or a mesh network (multi-hop, multiple receivers) topology with up to five hops. The average data rate per node is varied from 0.2 kbps to 3.2 kbps, placing the upper bound on the total network throughput at 100 kbps (max possible throughput of the IEEE 802.15.4 network [18]). No assumptions was made on packet inter-arrival time except that every node attempts to send all the data in the network period of $T=5$ seconds. The length of single transmission slot of the SSA, RSA and ESA algorithms was set to be 20 backoff slot (6.4 milliseconds) allowing the maximal possible packet size of 127 bytes.

The following medium access methods were used for each topology:

a) CSMA-CA: A regular CSMA-CA channel access mechanism to analyze the network throughput under varying loads. Here, whenever a sensor node has a packet to transmit, it waits for a random amount of time and then senses if the medium is free and if so, transmits the packet. In case the medium is busy or the packet fails delivery (no acknowledgement received), the node waits for another random amount of time and then retries. The limit on the maximum transmission attempts for a single packet was set to 3.

b) Static Slot Assignment scheme with a fixed redundancy factor $\delta=0.05*r$: The random schedule is selected at start-up with $(N*r + \delta)$ slots, as discussed in Section II. When a sensor node has a packet to transmit, it traverses the schedule to look for the next slot marked as available for transmission. The time until that slot is calculated and a wait timer is generated. The packet is transmitted in the slot using CSMA/CA. Any retransmissions also follow the same procedure. The mesh forwarding parameter δ remains constant and the slot schedule is not updated during operation.

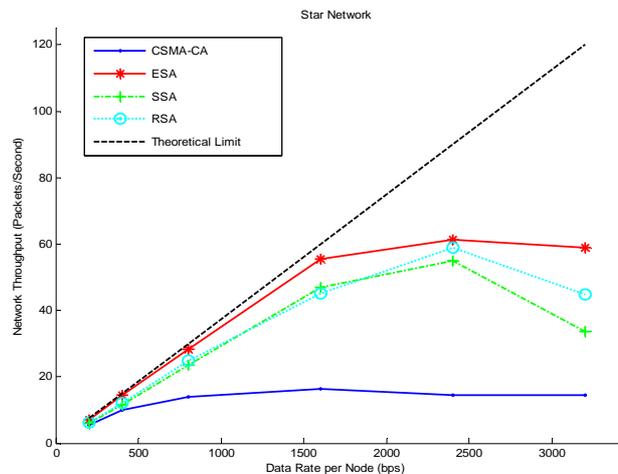


Figure 5. Comparison of throughput of a Star Network with CBR traffic.

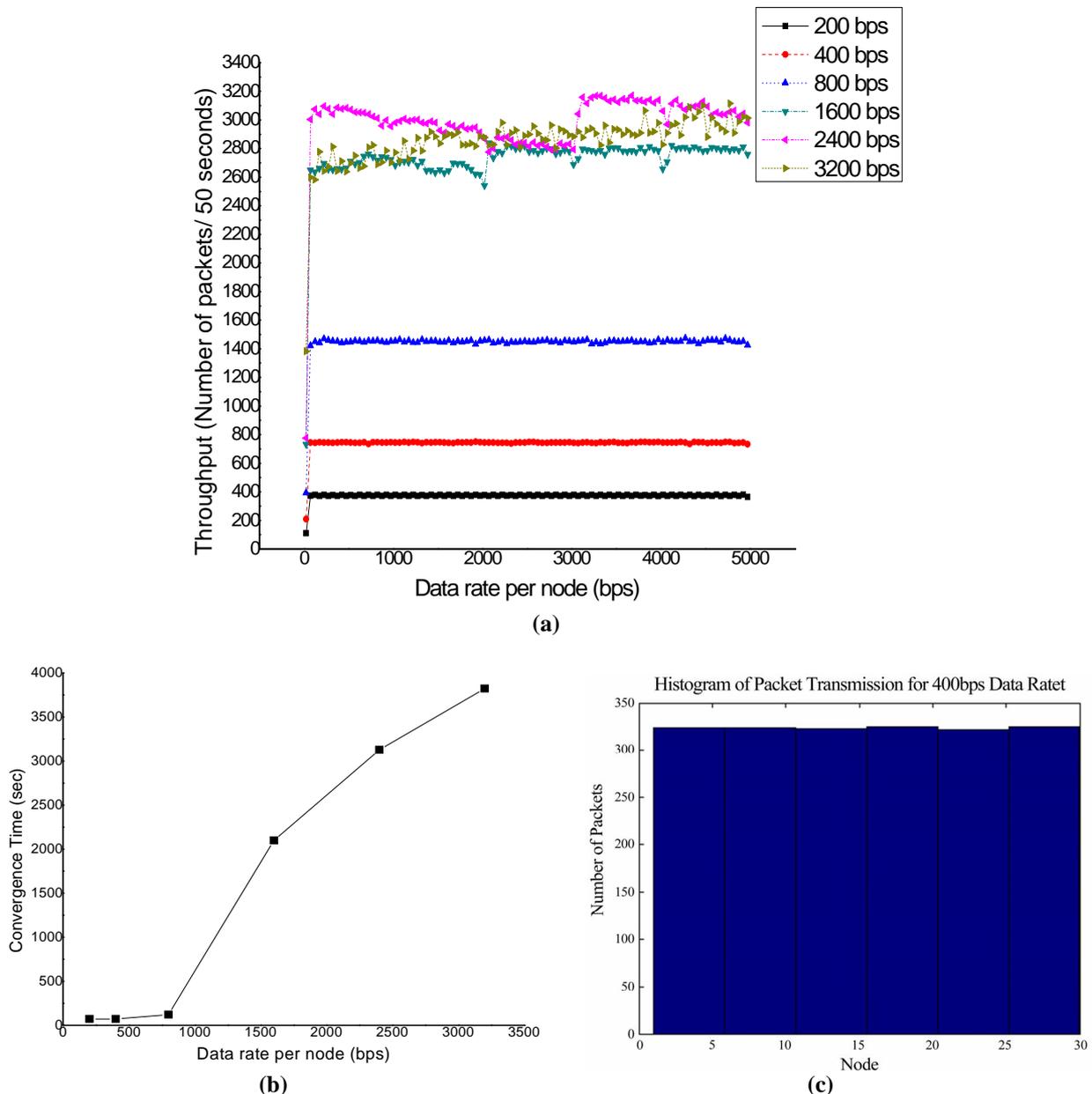


Figure 6. (a) Comparison of throughput varying with simulation time of an ESA Star network with CBR traffic with different data rates (b) Convergence times of an ESA Star network with CBR traffic at different data rates (c) Histogram of packet transmission for 400bps data rate for a Star Network with CBR traffic.

c) Random Slot Assignment: A random schedule is generated by each sensor node every network period by random re-initialization of V_T to mark $(\eta * r + \delta)$ slots as available for transmission. Packet transmissions follow the same procedure as in part (b) above. The vector P_T is not updated with the status of the transmissions and parameter $\delta = 0.05 * r$ remains constant.

d) Evolutionary Slot Assignment: The network is simulated with every node using ESA algorithm as out-

lined in Section II. Vector P_T is updated after every transmission attempt and vector V_T is updated by tournament selection every network period. Redundancy parameter δ is updated every network period.

Star and multi-hop mesh network topologies with constant-bit-rate (CBR) traffic from the sensor nodes and a mesh topology with Poisson traffic were analyzed. All simulations were run for 5000 seconds. The throughput was measured as the net received data in the last 1000

seconds of the simulation. The effect of increasing traffic on the throughput network was studied. The convergence time was measured as the time taken to settle to 90% of the maximum stable value. In order to avoid any localized inferences, the results are averaged from three random seeds of each simulation setup.

4. Results

Three different simulation scenarios were created as discussed in the previous section: a star topology with CBR traffic, mesh topology with CBR traffic and a mesh topology with Poisson traffic.

Figure 5 shows the throughput of the network at varying input data rates for the different methods of medium access for a 31 node star network. It can be clearly seen that as the data rate per node increases, the throughput of CSMA-CA method falls below 20 kbps or $\approx 20\%$ of useful bandwidth. The peak of SSA and RSA methods bring the throughput up to 50 kbps but for a large data rate of 3.2 kbps per node, the network throughput is seen to fall. On the other hand, the maximum network throughput achieved by ESA is about 60 kbps which is a 300% improvement over a CSMA-CA network. From Figure 6(a) and 6(b), it can be seen that the convergence time of ESA increases exponentially with increase in the data rate per node.

Mesh networks do not follow the same behavior as that of the star networks since they have to adjust their schedules to cater to intermediate traffic, even when these nodes are absolutely unaware of the routes established. Figure 7(a) shows that for mesh networks, CSMA-CA outperforms SSA and RSA for lower data rates, however for higher data rates, these algorithms begin to show better performance than CSMA-CA. The ESA method out-performs all other methods by providing up to a 200% improvement in throughput compared to any other tested method. ESA provides fair channel access to all nodes of the network as illustrated in Figure 6(c) and Figure 9(c).

5. Discussions

The very first observation is that in both star and mesh network configurations ESA provides a consistent advantage over IEEE 802.15.4 CSMA-CA protocol and simpler SSA and RSA methods. Results demonstrate that ESA creates a substantial (200%–300%) improvement in throughput over CSMA-CA networks while providing fair access to the medium. This increase in performance is created without any awareness of packet routing, without centralized scheduling and extra scheduling traffic, and without using a common time reference, beacons or knowing schedules of other nodes. Performance gain

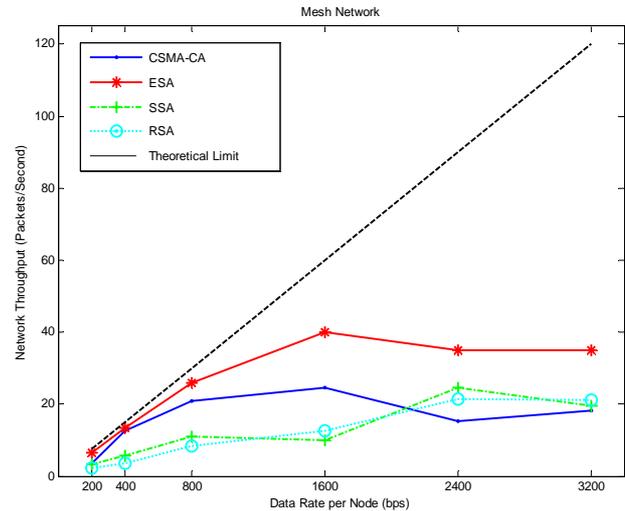


Figure 7. Comparison of throughput of a Mesh Network with CBR traffic

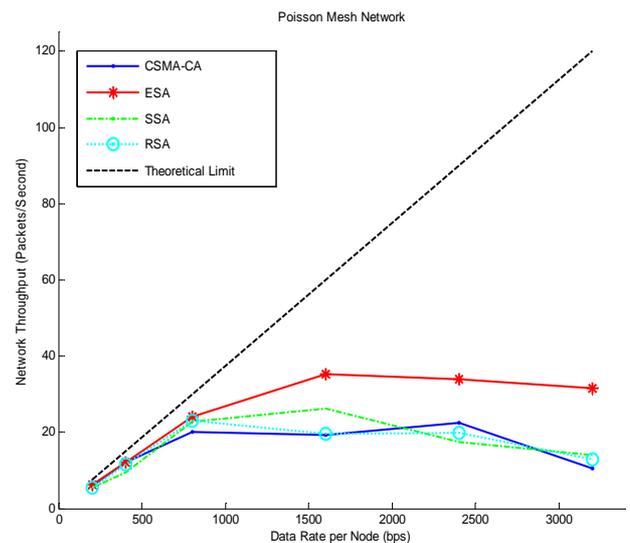


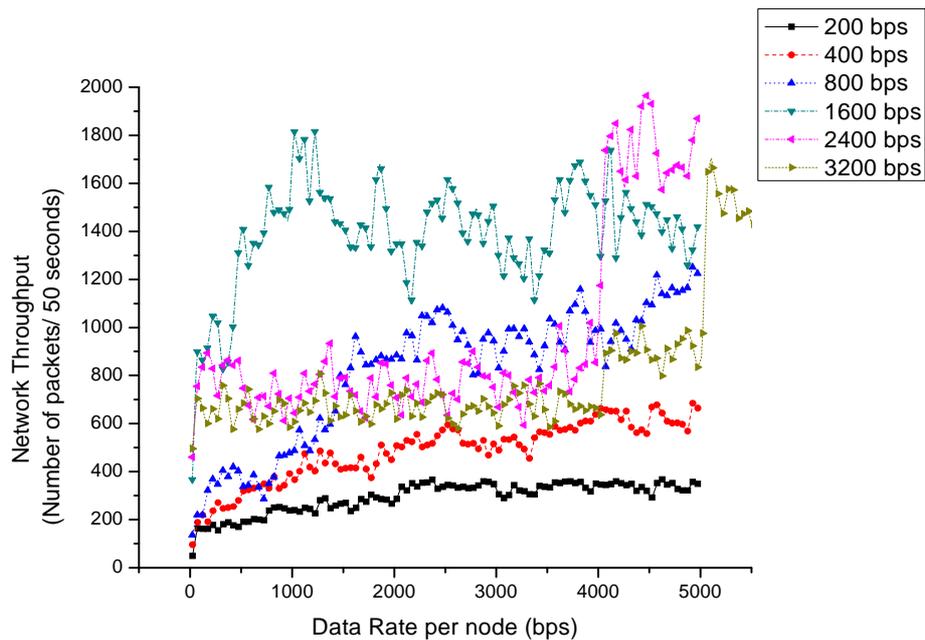
Figure 8. Comparison of throughput of a Mesh Network with poisson traffic.

of the ESA method becomes very clear for networks with high traffic, where probability of collisions grows significantly. As results in Figure 5, Figure 7 and Figure 8 show, performance of the simpler SSA and RSA methods degrade significantly for higher bit rates since neither transmission slots nor CSMA-CA procedure are not capable of effective collision resolution under heavy loads. ESA gains this improvement by adapting individual transmission schedule of each node so that the total number of collisions is minimized. This adaptation is purely based on outcome of packet transmission and independent of neighboring node visibility, thus eliminating any need for knowledge of network topology (ex. hidden terminal problem).

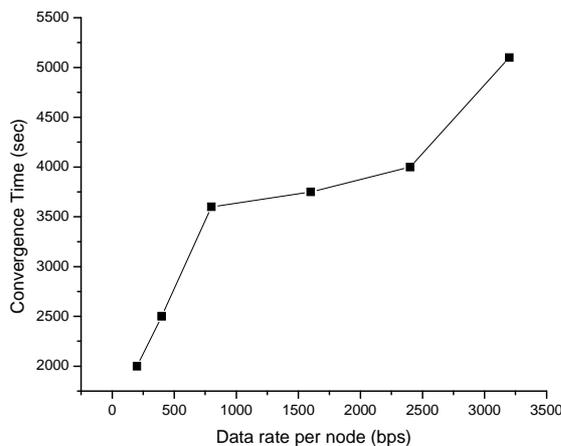
The second observation is that the consistent gain in bandwidth utilization of ESA method is achieved through the adaptive, evolutionary nature of the algorithm. Random selection nature of SSA and RSA methods itself shows improvement over CSMA-CA for higher data rates by increasing the overall randomness of the channel access. However, under heavy load condition ESA provides almost twice the bandwidth in comparison to SSA and RSA (Figure 5 and Figure 7). Another important clue showing the advantage of evolutionary adaptation is presented in Figure 7, where throughput of SSA and RSA for data rates below 2.4kbps is worse than

pure CSMA-CA. We attribute this effect to the fact that a node in a pure IEEE 802.15.4 CSMA-CA network is not bound by a schedule and has more attempts at transmission than a node using SSA or RSA. ESA will attempt virtually the same number of transmission attempts as SSA and RSA but adapting the schedule will allow ESA to avoid most collisions and thus provide reliable performance over a wide range of data rates.

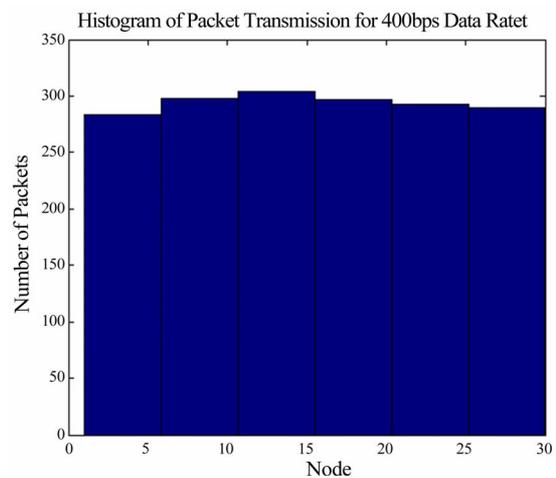
Our third observation is that the speed of convergence for ESA (Figure 6(b) and Figure 9(b)) depends on a specific network configuration and traffic. The convergence is fastest for networks with low traffic since there exists



(a)



(b)



(c)

Figure 9. (a) Throughput varying with simulation time of an ESA mesh network with poisson traffic at different data rates (b) Convergence times of an ESA Star network with CBR traffic at different data rates (c) Histogram of packet transmission for 400bps data rate for a Mesh Network with CBR traffic.

a multitude of possible non-conflicting transmission schedules. As the network size grows, the number of possible solutions is decreased and the algorithm spends a longer time seeking a suitable set of schedules.

Finally, we would like to note that the ESA algorithm is computationally lightweight and can be easily implemented even on most energy-constrained platforms. For example, the number of C code lines in the ESA scheduler is 652. On the other hand, the proposed algorithm can actually reduce power consumption of a node by reducing the number of retransmission attempts. Overall, one of the strong points of ESA is that the schedule will remain stable if all packets are delivered as needed and will self-adapt if the traffic pattern is changed.

6. Conclusions

In this work, a novel Evolutionary Slot Assignment was developed and tested. In the ESA method, the sensor nodes, independent of each other, adapt internal schedules to a traffic pattern minimizing collisions and improving bandwidth utilization. ESA method makes no assumption of network topology, packet routing, traffic from neighboring nodes, does not require a centralized scheduler and does not create scheduling traffic. ESA also does not need to know schedules of neighboring nodes and does not require a common time marker or synchronizing beacons. Simulations were performed for two topologies, a star and a mesh network, and two different traffic scenarios, constant-bit rate traffic and Poisson traffic. Networks using CSMA-CA medium access were compared with the static slot assignment, random slot assignment and the evolutionary slot assignment algorithms. Simulation results show 200%–300% improvement in throughput of proposed ESA algorithm in comparison to pure CSMA-CA.

7. References

- [1] H. Karl and A. Willig, "A short survey of wireless sensor networks," Telecommunication Networks Group, Technische Universitat Berlin, Hasso-Plattner Institute, Potsdam.
- [2] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in Proceedings of IEEE International Conference Communications, pp. 3418–3423, 2002.
- [3] I. Demirkol, C. Ersoy, and F. Alagöz, "MAC protocols for wireless sensor networks: A survey," IEEE Communications Magazine, Vol. 44, No. 4, pp. 115–121, 2006.
- [4] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, pp. 535–547, March 2000.
- [5] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15.4," IEEE Press Book, 2004.
- [6] IEEE P802.15.4/D18, Draft Standard: Low Rate Wireless Personal Area Networks, February 2003.
- [7] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "Performance evaluation of the IEEE 802.15.4 MAC for low-rate low power wireless networks," in Proceedings of IEEE International Performance Computing and Communication Conference (IPCCC'04), pp. 701–706, Phoenix, AZ, April 2004.
- [8] J. Misić, S. Shafi, and V. B. Misić, "Maintaining reliability through activity management in 802.15.4 sensor clusters," IEEE Transactions on Vehicular Technology, Vol. 55, No. 3, pp. 779–788, 2006.
- [9] J. Misić, S. Shafi, and V. B. Misić, "Performance of beacon enabled IEEE 802.15.4 cluster with downlink and uplink traffic," IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 4, pp. 361–376, 2006.
- [10] K. Varadhan, "The Ns manual," The VINT Project, August 2000.
- [11] E. Sazonov, R. Jha, K. Janoyan, V. Krishnamurthy, M. Fuchs, and K. Cross, "Wireless intelligent sensor and actuator network (WISAN): A scalable ultra-low-power platform for structural health monitoring," SPIE Proceedings 6177, March 2006.
- [12] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," IEEE Infocom'02, Vol. 3, pp. 1567–1576, June 2002.
- [13] T. van Dam, and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," ACM SenSys 2003, pp. 171–180, November 2003.
- [14] C. S. Raghavendra and S. Singh, "PAMAS—power aware multi-access protocol with signaling for ad hoc networks," ACM Computer Communications Review, Vol. 28, No. 3, pp. 5–26, 1998.
- [15] M. J. Miller and N. H. Vaidya, "On-demand TDMA scheduling for energy conservation in sensor networks," Technical Report, University of Illinois at Urbana Champaign, 2004.
- [16] M. L. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," IEEE Infocom'04, March 2004.
- [17] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," ACM SenSys'03, pp. 181–192, November 2003.
- [18] V. Krishnamurthy and E. Sazonov, "A reservation-based protocol for monitoring applications using IEEE 802.15.4 sensor networks," in International Journal of Sensor Networks, Vol. 4, No. 3, pp. 155–171, 2008.
- [19] T. Bäck, "Evolutionary algorithms in theory and practice," Oxford University Press, New York, 1996.
- [20] T. Bäck, D. Fogel, and Z. Michalewicz, "Handbook of evolutionary computation," Oxford University Press, 1997.
- [21] S. Baluja, "Population-based incremental learning: A

- method for interacting genetic search based function optimization and coemptive learning,” Technology Republic, No. CMU-CS-94-163, Carnegie Mellon University, 1994.
- [22] G. Harik, F. G. Lobo, and D. E. Goldberg, “The compact genetic algorithm,” in Proceedings of the International Conference on Evolutionary Computation (ICEC 98), pp. 523–528, 1998.
- [23] H. Pang, K. Hu, and Z. Hong, “Adaptive PBIL algorithm and its application to solve scheduling problems,” IEEE International Symposium on Computer-Aided Control Systems Design, pp. 784–789, October 2006.
- [24] Z. He, C. Wei, B. Jin, W. Pei, and L. Yang, “A new population-based incremental learning method for the traveling salesman problem,” in Proceedings of the 1999 Congress on Evolutionary Computation, Vol. 2, pp. –1156, 1999.
- [25] S. Yang and X. Yao, “Dual population-based incremental learning for problem optimization in dynamic environments,” in M. Gen *et al.* (editors), Proceedings of the 7th Asia Pacific Symposium on Intelligent and Evolutionary Systems, pp. 49–56, 2003.
- [26] S. Baluja and D. Simon, “Evolution-based methods for selecting point data for object localization: Applications to computer-assisted surgery,” International Journal of Applied Intelligence, Vol. 8, pp. 1–13, 1997.
- [27] L. Chen and A. Petroianu, “Application of PBIL to the optimization of PSS tuning, power system technology,” in Proceedings of 1998 International Conference on Power System Technology, Vol. 2, pp. 834–838, 1998.
- [28] B. L. Miller and D. E. Goldberg, “Genetic algorithms, tournament selection, and the effects of noise,” Complex Systems, pp. 193–212, June 1995.
- [29] C. E. Perkins and E. M. Royer, “The ad hoc on-demand distance vector protocol,” Ad hoc Networking, Addison-Wesley, pp. 173–219, 2000.