Article:
Mickus, Tautvydas, Mitchell, Paul Daniel orcid.org/0000-0003-0714-2581 and Clarke, Timothy orcid.org/0000-0002-5238-4769 (2017) The emergence MAC (E-MAC) protocol for wireless sensor networks. Engineering Applications of Artificial Intelligence. pp. 17-25. ISSN: 0952-1976

https://doi.org/10.1016/j.engappai.2017.03.003

The Emergence MAC (E-MAC) Protocol for Wireless Sensor Networks

Tautvydas Mickus[a,*], Paul Mitchell[a], Tim Clarke[a]

[a] *Department of Electronics University of York, Heslington, York, YO10 5DD, UK*

Abstract

Large scale biological systems often exhibit emergent properties that are attractive in an engineering context. In this paper, the context is a class of wireless sensor networks for emergency environmental monitoring. The attractive properties are simplicity, self-organisation, adaptiveness to scenario change and a lack of scenario-specific parameter tunings. Emergence Medium Access Control (E-MAC) is a scheme inspired by biological social populations that individually react to environmental stimuli. Using a very simple protocol, it exhibits the desired emergent properties. When compared to a well established practical counterpart, the IEEE 802.11 CSMA/CA standard, it exhibits better throughput, end-to-end delay and fairness. This paper describes the motivation and design of E-MAC, and presents the above comparison.

Keywords: Medium Access Control, Distributed Artificial Intelligence, Wireless Sensor Networks

1. Introduction

Imagine the scenario where an emergency service, such as Fire and Rescue, is required to monitor a large area of moorland for spontaneous outbreaks of brush fire [1]. Any such monitoring would be required to report on temperature and humidity levels that indicate high risk conditions and, subsequently,

[*] Corresponding author

Email addresses: `t.mickus@york.ac.uk` (Tautvydas Mickus), `paul.mitchell@york.ac.uk` (Paul Mitchell), `tim.clarke@york.ac.uk` (Tim Clarke)

the movement of fire fronts. The movement of fire fronts can be highly unpredictable and poses a serious danger to personnel and equipment. This is an ideal opportunity to deploy a wireless sensor network (WSN) over a wide area from a suitable aerial platform. Based on the operational need it is also possible to deploy more of these low-cost nodes.

This scenario presents a set of significant challenges [2]. Long-term remote operation necessitates low power usage and a very simple MAC protocol in each inexpensive node. In contrast, nodes are required to minimise end-to-end delay with no sensor node being dominant (high fairness levels). In the case of these simple nodes, only one communications channel will be available, necessitating an efficient MAC protocol to control the transmissions, ensure correct operation and achieve high throughput. Nodes will be required, at different times, to act purely as relay nodes whilst at other times, they may be additionally required to generate and place data on the network. The protocol must facilitate adaptability.

Many protocols have been proposed for WSNs which offer different benefits [3][4][5]. Schemes that employ sophisticated synchronisation or significant information exchange to achieve organisation and performance are inappropriate in the context presented here. Yet, as the scale of networks increases, the need for some form of synchronisation and information exchange becomes overwhelming even if only at a local level.

Routing becomes a challenging task in large-scale networks as well. Dissemination of routing information and discovery of routes becomes difficult process. There are, however, many examples and proposals for good routing practices in the scientific community [6][7]. In this paper we focus on the MAC layer.

Here, we present our proposed solution, Emergence Medium Access Control (E-MAC), and compare its performance to that of a basic implementation of the IEEE 802.11 standard. We choose this latter protocol because it is well understood and well established. Even though it uses some hardware capabilities such as carrier sensing and additional RTS/CTS messages, the IEEE 802.11

protocol itself is very simple and clean. We focus on comparative performances over a multi-hop chain. The contributions of E-MAC are:

- A different approach to MAC. The nodes search for the throughput they are able to achieve and then use this information for data transmissions/generation, regulating traffic flow.
- A very simplistic MAC protocol that allows nodes to achieve high throughput through multi-hop networks under a variety of situations without the need to tune system parameters.
- The proposed protocol also shows several emergent behaviours:
  - self-organisation
  - flow control on both hop-by-hop and end-to-end basis
  - indirect synchronisation between the nodes as packets are relayed
  - minimal latency

The structure of this paper is as follows: Section 2 outlines the biological inspiration for the E- MAC protocol. We then describe the protocol itself in Section 3. Section 4 presents the scenario and simulation parameters for the comparative performance. The results are described in Section 5.

2. Biological Metaphors

The ability of natural systems to self-organise, reorganise and provide fault-tolerant operation has inspired a huge diversity of mathematical and engineering solutions [8][9][10]. For example, the evolutionary metaphor (e.g. genetic algorithms and genetic programming) has enabled otherwise intractable optimisations and facilitated the discovery of novel processes, algorithms and systems [11]. Similarly, the social metaphor (e.g. particle and robotic swarms and multi-agent systems) has done the same, and contributed to the understanding of the emergent properties of complex systems [12].

E-MAC was inspired by the social metaphor.

In this case, very simple entities, generally referred to as *agents*, can offer significant benefits and highly complex behaviours when operating in groups and interacting with each other using simple rules. This *swarming* offers emergent behaviour on a higher social level [13]. Examples from nature include:

- locust swarms which can fly in perfect synchrony in their billions, efficiently exploiting localised air streams [14]
- ant colonies which exhibit complex foraging and task allocation behaviour without central coordination [15]
- termite colonies that can build complex structures without a global blueprint [16]

All of these are achieved without central control, and only through very simple rules, interactions and reaction to the local agent environment, and without explicit encoding of the emergent behaviours. In each case there are up to millions of very simple entities that are continuously changing without affecting the overall performance. The complex behaviours arise from the interactions between individuals affecting their local environment. Self-organisation, adaptation and fault-tolerance are frequently the emergent properties of these systems. This simplicity and the same emergent properties correspond to what could be defined as ideal for WSNs.

When monitoring harsh environments over large areas of undulating terrain, we require cheap, simple nodes that can adapt to different communication scenarios without the need to tune specific system parameters. Also, network fault-tolerance is needed where nodes are likely to progressively fail at the onset of a fire front. Furthermore, adding nodes should not trigger wholesale network reconfiguration to accommodate them; only locally-affected regions should adapt without affecting global emergent behaviour.

All of this *can* be otherwise achieved with precise deployment planning and complex algorithms. Such approaches tend to introduce many tunable parameters which require more operational maintenance. Also, it is not usually possible to anticipate every scenario and its conditions. We assert that it is better exploit biological metaphors that offer appropriate emergent properties through simple rules of interaction.

The E-MAC protocol employs the notion of reaction to the intensity of stimulus from neighbouring agents. We use a stochastic approximation of the probability of successful message packet transmission as that stimulus.

2.1. Task Allocation and Division of Labour in Social Insects

Here we represent an example of stimulus-based self-organising emergent behaviour to illustrate our motivation for the development of the E-MAC scheme.

It has been observed that many species of social insects exhibit emergent task allocation and division of labour [17]. Without the need for a leader, colonies comprising huge number of individuals are able to organise their various tasks. The process usually arises through emergence from simple actions taken by individuals. In addition, such processes are highly robust and adapt to the different needs of the colony.

Bonabeau [18] proposed a model based on a response threshold that models the behaviour of ants and bees and shows emergence behaviour at the colony level for task allocation. The response threshold defines how individuals react to their environment (stimulus). It provides a way to define a probability of taking an action, given certain stimuli from environment and its relationship with the threshold of that stimulus. A threshold can be varied among different individuals - therefore creating specialised workers. For example, in an ant colony we can consider forager and fighter ants. Foragers will have a lower threshold for collecting food and a higher threshold for fighting. Therefore they will more likely take up foraging. Fighter ants with a reversed threshold would show a higher tendency towards fighting. Nevertheless given the lack of foragers, the stimuli for foraging increases, therefore fighter ants would start to get involved into foraging tasks as well. The process also involves a learning process. If an agent is performing a task, the threshold for that task will decrease (increasing the likelihood of performing that task again). This also provides a natural process for specialisation.

For example the probability to take up a task given a certain threshold and stimuli can be expressed as:

$$T_\theta(s) = \frac{s^n}{s^n + \theta} \tag{1}$$

where s is the environmental stimuli, θ - the response threshold and n defines the steepness of the curve (see Figure 1).

θ essentially defines the tendency to take up action given the environmental stimuli, so differently-specialised insects would have different threshold towards certain tasks. For example, when θ is 1 in Figure 1 the stimuli has to be very high to increase the probability of performing the task defined by this threshold. However, when θ is 50, even a small stimuli will have high probability of eliciting a response.

Another example (Figure 2) of a response curve function is given by Plowright [19] [18]:

$$T_\theta(s) = 1 - e^{-s/\theta} \tag{2}$$

Similar trends arise in both functions where the probability of engaging is small for $s << \theta$ and is close to 1 for $s >> \theta$ [18].

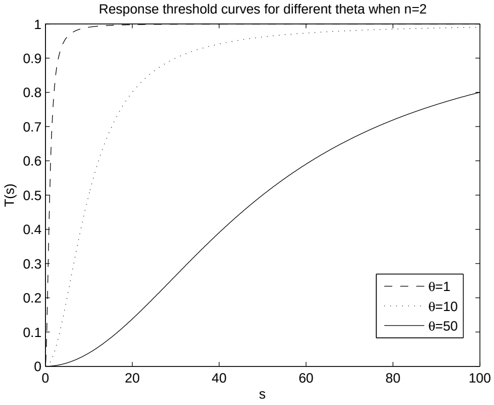

Figure 1: Response threshold curves based on Equation 1

The very simple model presented here can provide very powerful and complex behaviour. Without explicitly specifying a behaviour, it emerges due to so-

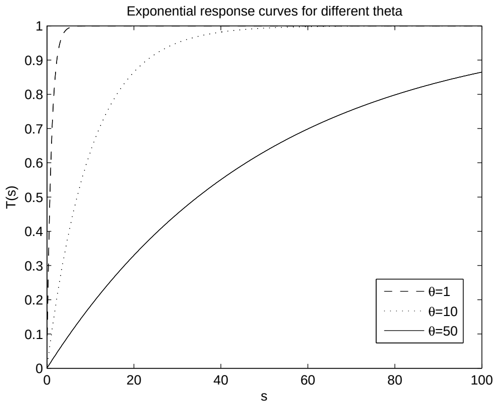

Figure 2: Exponential response curves based on Equation 2

cial interactions between insects and the stigmergy (the phenomena of indirect communication through altering the environment). In addition very robust, self-organising, scalable and adaptive behaviour is achieved.

This type of behaviour can be applied to a variety of optimisation problems as well as resource allocation algorithms. In addition, the same process of response thresholds and stimuli can be found in other emergent swarm behaviours such as clustering or sorting [20][18].

3. Protocol Design

The goal of E-MAC is to provide good performance with very low complexity. The protocol is based on a simple implementation inspired by the biological social metaphor of swarm reactions to an environment. The bare minimum amount of data is shared during each data packet transmission. No additional transmissions are made and there is no need for carrier sensing.

This section will start with a basic overview of components in E-MAC and present an overall view of what E-MAC does. Then it will continue with detailed information on the algorithm.

3.1. Transmission Delay

Many MAC schemes or protocols employ the concept of back-off to reduce congestion and offered traffic, allowing other transmissions to compete for channel access. Once a packet is either received or dropped, back-off is usually reset. Any information on previous actions and outcomes in the environment is then lost. E-MAC employs a back-off strategy that does not subsequently reset, but either increases or decreases incrementally. We more appropriately use the term *transmission delay* which is changed after each (un)successful packet transmission. Increasing or decreasing transmission delay controls the overall transmission rate and, in the manner of conventional back-off, allows other nodes to transmit on the channel. However, unlike traditional back-off schemes, it maintains a transmission rate that becomes periodic and predictable. In effect, the transmission delay retains historical information about the environment which helps to prevent nodes from experiencing repeated congestion.

3.2. Basic Operation

E-MAC performs a simple update action which is called when an acknowledgement is received or a time-out occurs. During an update E-MAC simply changes the transmission delay duration depending on the acknowledgement outcomes. The adjustable transmission delay is engaged every time the MAC layer passes a packet to the physical layer for transmission. Therefore once the transmission delay is engaged, the node is not allowed to transmit packets, in the manner of conventional back-off. When the transmission delay expires, the node is again allowed to send a packet. Controlling the transmission delay can effectively allow control of the time period between packet transmissions. The way transmission delay is varied is based on a biological social metaphor. Both the averaged and most recent acknowledgement outcomes are used to define an apppropriate stimulus to modify the transmission delay.

3.3. Robbins Monro and Probability of Success

The stimulus used to increase/decrease transmission delay is the average observed probability of successful packet transmission. Here we employ a stochastic approximation, the Robbins Monro algorithm [21]. It offers approximate averaging without the need for significant storage of past values. Additionally, it approaches an average value in a non-linear way, which provides a more realistic stimulus representation and offers the possibility of continuous reaction. The Robbins Monro algorithm is given by Equation (3):

$$X_i = (1-\alpha)X_{i-1} + \alpha X_{new} \tag{3}$$

where X_i is the approximated mean after iteration i and X_{new} is a new sample. In E-MAC, X_{new} represents the outcome of the i^{th} transmission (0 or 1 for failure or success respectively). Updating X_i at each transmission outcome gives an approximate average (probability of success). It provides a way to track the current probability of success at each node. This can be then used as the intensity of stimulus for appropriate agent action. α weights current experience against the prior approximation of the mean.

This forms the response threshold which was discussed in Section 2.1. If we draw a random number between 0 and 1 and take action if the drawn number is larger than X_i then the probability of responding is $1 - X_i$. For a lower X_i value the algorithm will be more likely to respond. Figures 3 and 4 show the response probabilities given the starting value and number of consecutive events (success or failure). The curves also show very similar trends to the exponential response functions shown in Figures 1 and 2 and Equations 1 and 2.

3.4. The Basic E-MAC Algorithm

Using the stimulus proposed in Section 3.3 we implement Algorithm 1 that determines the changes to the transmission delay.

When a node experiences contention on the channel there is a greater likelihood of corrective action otherwise there is none (lines 9 - 10 and 17 - 18). In addition to responding to acknowledgement failures (lines 11 - 12) we want a node to react to the historical performance of the adjacent downstream node (lines 19 - 20) which prevents congestion. The whole algorithm mimics the

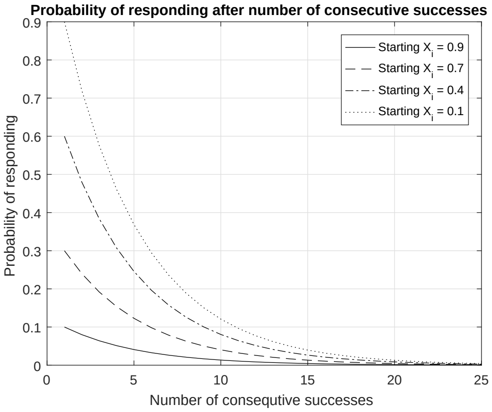

Figure 3: Probability of responding (take action) after consecutive successes starting at different X_i values

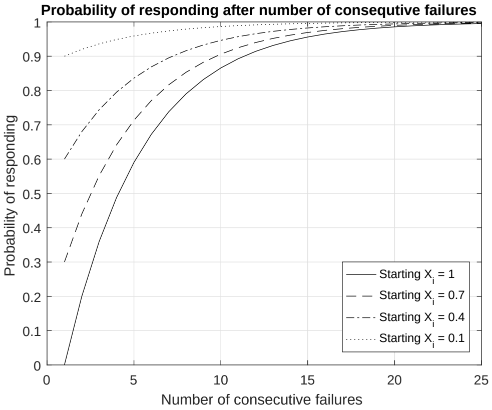

Figure 4: Probability of responding (take action) after consecutive failures starting at different X_i values

Algorithm 1: The E-MAC Algorithm

```
// Initialisation
set $\alpha$ , $change\_scale$ , $pSuccess$ , $tx\_delay$
while network running do
    // Update
    if ack failed then
        $recent\_outcome = 0$
    else
        $recent\_outcome = 1$
    end
    $pSuccess = (1 - \alpha) * pSuccess + \alpha * recent\_outcome$
    // Ack Effect
    $R$ = generate random number between 0 and 1
    if $R > pSuccess$ then
        if $recent\_outcome = 0$ then
            $tx\_delay = tx\_delay + change\_scale$
        else
            $tx\_delay = tx\_delay - change\_scale$
        end
    end
    // Queue Effect
    $R2$ = generate random number between 0 and 1
    if R2>pSuccess then
        if queue at next hop > my queue then
            $tx\_delay = tx\_delay + change\_scale$
        end
    end
end
```

way in which swarm colonies react to their immediate environment which is usually represented as the stimuli intensity.

Experimentation shows that direct response to acknowledgement performance (`Ack Effect` at lines 9 - 16) effectively controls transmission delay. The network settles at reasonable delay values throughout and avoids collisions along a multi-hop chain. Nevertheless, congestion can build up at nodes. To alleviate queue build-up we have added another action (`Queue Effect` at lines 17 - 22). It requires each node to share its queue size with its adjacent upstream node by adding this small amount of information to every transmission and acknowledgement.

3.5. Multiple source operation

The protocol, in the form depicted in Algorithm 1, does not search for a transmission delay that gives fair operation when more than one source node exists in a multi-hop chain. For this, some extra functionality is necessary. In the spirit of the biological social metaphor, the chain continues to use established information and forces nodes that become active to join the flow rather than disrupt it through dissonant transmission delays.

Using a simple extension, if a relay node also starts to function as a source node (or source node also starts to function as a relay), its packets joins the flow by limiting its own transmissions to the incoming receptions. It is only allowed to send a packet forward once the relay packet is received. This prevents collisions between flows from different sources along the chain.

3.6. Fair Queuing

We have adopted a fair queuing strategy in E-MAC. This implies that packets in the queue from different sources are treated fairly to avoid the formation of dominant nodes. The queue prioritises packets from different sources in a round-robin fashion. In addition if there is more than one packet in the queue from a specific source, the most recent one is transmitted and older ones associated with that source are discarded. The use of such queueing is justified by the emergency environmental monitoring scenario which requires up-to-date data. Nevertheless, the queuing itself does not guarantee fair operation as the relayed packets can be lost based on MAC behaviour and collisions further down the chain.

This strategy may seem wasteful as not all packets coming from upstream sources are passed on, but all are acknowledged - despite some later being dropped. However, the pay-off is that the chain can quickly adapt to new sources arising along a chain using one simple protocol. Given the scenario described at the start, the availability of information from all active source nodes at a high and sustainable data rate is important.

We could have taken a more parsimonious approach where new-source nodes inform those upstream to send only every n^{th} packet. However, our experimentation shows that, if a particular node then stops sourcing packets, it takes a lot longer for upstream source nodes to re-adapt and begin appropriately to send data more frequently.

3.7. Overall protocol process

Several different events take place during wireless node operation at the MAC layer. These are packet reception from the Physical Layer, packet reception from the Network Layer, Acknowledgement Timeout and Back-off Timeout or, in E-MAC, Transmission Delay expiry.

Initialisation of Algorithm 1 occurs during node startup. When packet reception from the Physical layer occurs, the MAC layer passes the packet to the Network Layer if appropriate, and an acknowledgement is sent back. When packet reception from the Network Layer occurs, if the node is currently not receiving a packet at the Physical Layer and/or a Transmission Delay is not in progress, the node passes the packet for transmission to the Physical Layer immediately and the Transmission Delay is then engaged. Otherwise it waits until the current Transmission Delay expires. Once the acknowledgement is received or a time-out occurs Algorithm 1 lines 2-23 execute to update the Transmission Delay value.

4. Simulation Parameters and Assumptions

4.1. The Basic Scenario

We evaluate E-MAC as a 12-node multi-hop chain, indexed from 0 to 11 (0 is the sink). All nodes use the same channel for transmission and reception. All nodes are identical and can act either as relays, sources, or both. There is no direct synchronisation between the nodes and the inter-hop distance is 200 meters. This scenario is shown for clarity in Figure 5.

4.2. Propagation and Radio

A traditional hop based model is used for the communication and interference where nodes are able to transmit their data over 1 hop (nearest neighbours) but interference is experienced over 2 hops (as shown in Figure 5). Later, we increase the interference range to observe the adaptability and performance of the protocols in different conditions. Packets are only received correctly if no interference and collisions are present. We define propagation delay based on the distance between the nodes. Given that real device hardware can only perform one action, transmit or receive, in simulation nodes are not permitted to transmit if they are in successful reception state.

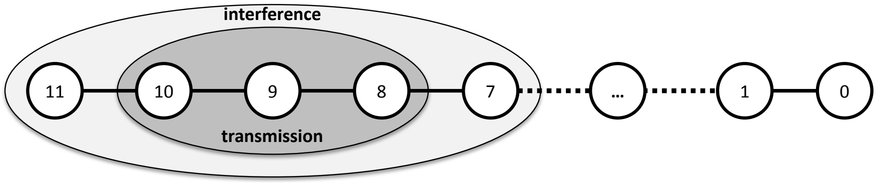

Figure 5: The chain scenario

4.3. Traffic

We use saturated traffic to simulate packet generation. This is to test the stability and maximum performance of the protocol. Also, we want to mimic the behaviour that would be required during critical monitoring situations where as much data as possible needs to be generated and conveyed along the chain.

A new packet is generated as soon as one is successfully transmitted and that node is available to transmit again. The initial packet transmissions start within the first second of simulation according to the uniform random distribution. The purpose of this is to avoid starting multiple sources at the same time.

4.4. The Comparison Scheme

This is the IEEE 802.11 standard - CSMA/CA with RTS/CTS and Binary Exponential Backoff (BEB). It is a widely used scheme that provides simplicity and great performance without requirements for synchronisation. It uses Collision Avoidance by means of carrier sensing and uses RTS/CTS messages to inform surrounding nodes of transmissions to deal with the hidden node problem on multi-hops. BEB aims to avoid further collisions or interference. Compared to many other much newer WSN protocols, CSMA is very low in complexity but offers good performance without synchronisation even for substantial networks. State-of-art MAC protocols that address particular aspects of WSN, in fact, use raw CSMA or 802.11 standard as a fall-back mechanism to maintain good performance when synchronisation is not available [22]. Due to its popularity and clearly defined implementation many researchers also use this scheme for comparison [23][4]. While IEEE 802.11 is not an energy efficient protocol, it still provides comparable or even better performance under varying conditions when compared to state-of-art protocols [24]. There are many alternative protocols for WSN. Some of more well known and established ones are S-MAC, Z-MAC, LEACH. S-MAC achieves an energy efficient operation via periodic sleeping, auto-synchronisation of sleep schedules and formation of virtual clusters [25]. Z-MAC is a hybrid protocol that combines TDMA and CSMA to improve performance and robustness to synchronisation errors [22]. LEACH,on the other hand,is a very different approach. It focuses on distributing the load evenly throughout the network, therefore prolonging the useful system lifetime [26]. There is a considerable range of approaches, many of which are complex. We now make some general comments on their suitability. Contention schemes are appropriate for distributed networks but suffer from energy waste through collisions. Distributed scheduling is potentially energy efficient but requires a lot of signalling and therefore scalability suffers. With the increased complexity of the state-of-art schemes, their appropriateness for comparison becomes questionable, whereas the classical IEEE 802.11 scheme is well known and established which aids in the understanding of performance.

This is why we chose it for comparison. E-MAC is of even lower complexity as it does not employ RTS/CTS messages or any hardware sensing to avoid collisions. E-MAC exploits collisions as part of the notion of reaction to the stimulus intensity of neighbouring agents. For clarity and comparison, we also show and discuss maximum theoretical bounds when evaluating the performance. Note, the same fair queuing policy is adopted in the CSMA scheme.

4.5. Simulation Parameters

Table 1 shows the simulation parameters.

Table 1: Simulation Parameters

Parameters	**Values**
Channel bit rate	250 Kbits/s
Data packet length	1000 bits
ACK packet length	20 bits
RTS/CTS packet length	20 bits
Transmit range	200 m (1 hop)
Interference range	400 m (2 hop)

5. Results

5.1. Metrics for Analysis

We assume that routing would be pre-initialised using Djikstra's shortest path routing (through a simple pure ALOHA scheme). We plot results as

Cumulative Distribution Functions (CDF) over 1000 simulations using different random number seeds. We use CDF because it provides an informative statistical view of protocol operation. Mean and standard deviation tables are additionally provided. Three different performance metrics are evaluated. Throughput is measured in Erlangs and is calculated based on the number of successful packets received at the sink throughout each simulation. It can be expressed as:

$$Throughput = \frac{number\ of\ packets\ received\ at\ the\ sink\ *\ packet\ size/bitrate}{simulation\ time} \tag{4}$$

End-to-end delay is measured in seconds from packet generation to arrival at the sink. We also establish the throughput fairness for different sources using Jain's Fairness Index [27] which is expressed as:

$$J = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{n}^{i+1} x_i^2} \tag{5}$$

where, in this case, n is the number of source nodes and x_i is the throughput from i^{th} source node. The results range from $1/n$ (worst case) to 1 (best case).

5.2. Performance evaluation

Figures 6 & 7 and Table 2 show throughput performance of E-MAC and the comparison CSMA scheme. There are two sources on a chain - one at the end (node 11) and one in the middle (node 5). For the 2-hop interference model, in all the simulations E-MAC significantly outperforms CSMA. For 3-hop interference, it occurs 97% of the time. Even though CSMA employs interference detection on the channel and avoids the hidden node problem through RTS/CTS, it is still not fully able to exploit channel capacity. The significantly simpler E-MAC protocol nevertheless achieves much better results. Furthermore, results also incorporate the period during which E-MAC is self-organising and settling towards the best transmission delay. This self-organisation of transmission delay indirectly synchronises the network to source transmissions, thereby avoiding collisions. If a source places packets on the network at the correct rate, they will move sufficient hops downstream before the next packet is sent, thereby avoiding collisions. Through the emergence of rate searching, hop-by-hop flow control occurs. Once settled to the correct rate the end-to-end flow control becomes operational and throughput quickly rises close to the theoretical bounds. Under E-MAC, without the need for an explicit timing mechanism, the network achieves very good throughput performance.

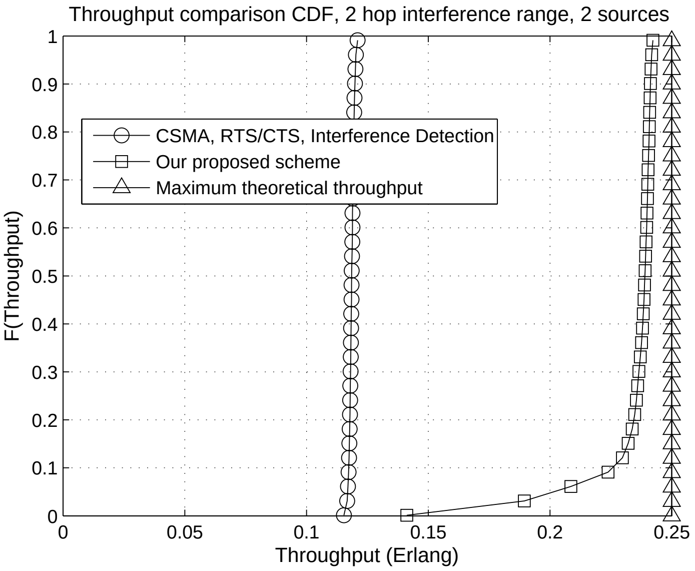

Figure 6: Throughput for system with 2 sources and 2 hop interference range

Table 2: Throughput (Erlangs) (mean ± standard deviation)

	CSMA	E-MAC	Maximum Theoretical
2 hop interference	0.1186 ± 0.0010	0.2344 ± 0.0140	0.25
3 hop interference	0.1015 ± 0.0009	0.1853 ± 0.0186	0.20

Figure 8 and 9 show the packet end-to-end delay results for 2 hop and 3 hop interference respectively. Both graphs represent delay for packets arriving from 2 different sources (nodes 11 and 5) for both schemes. Again, sig-

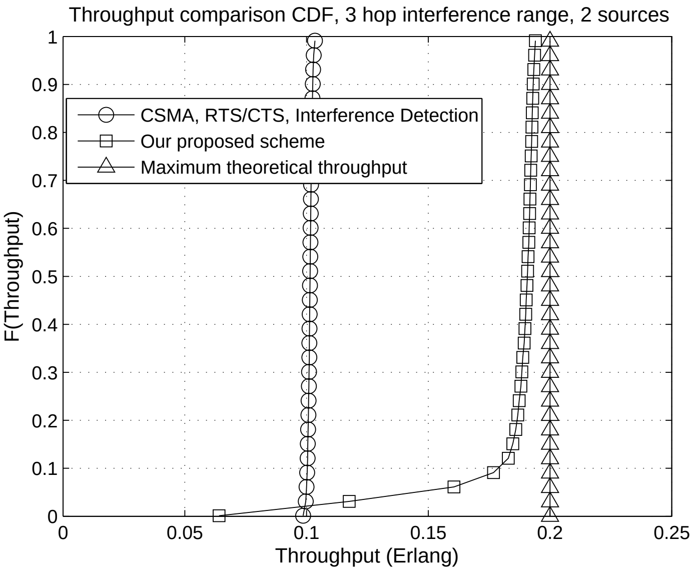

Figure 7: Throughput for system with 2 sources and 3 hop interference range

nificantly better end-to-end delay performance can be seen using the E-MAC protocol. Note, from Table 3, that end-to-end delay statistics for E-MAC and CSMA delay performance are both fairly consistent over the 1000 simulations. The minimal latency of E-MAC also arises through the same rate interactions. Once source nodes find a good transmission delay, the packets travels through the route with minimal collision or interference. This ensures that a packet is not held up at any node due to back-off or failure. The outcome is reduced end-to-end delay.

Figure 10 and Table 4 show fairness results (Jain's Fairness Index, as described in Section 5.1) for the 2 hop and 3 hop interference models using E-MAC and CSMA protocols. The results indicate ideal performance from the E-MAC scheme and near ideal performance from CSMA. Despite both schemes using the same fair queuing mechanism, some packets are lost under the CSMA protocol, due to collisions. This slightly reduces CSMA fairness.

To extend the scope of the results to show the performance of E-MAC with different numbers of source nodes ranging from 1 to 10, we consider the chain

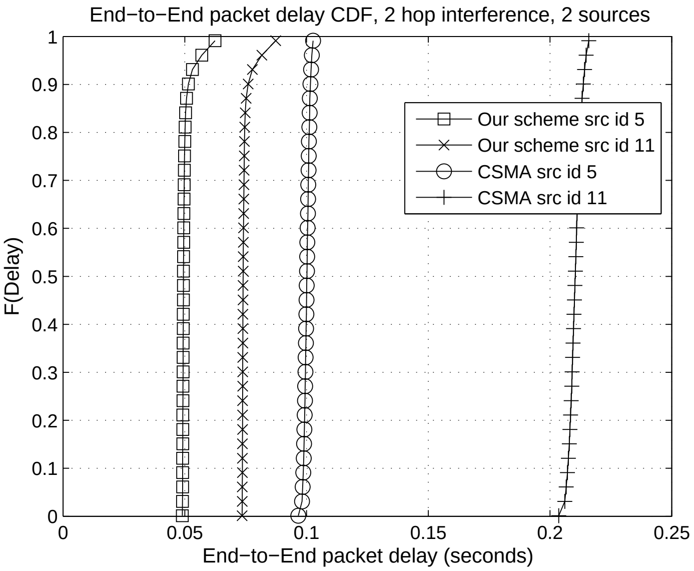

Figure 8: End-To-End delay for system with 2 sources and 2 hop interference range

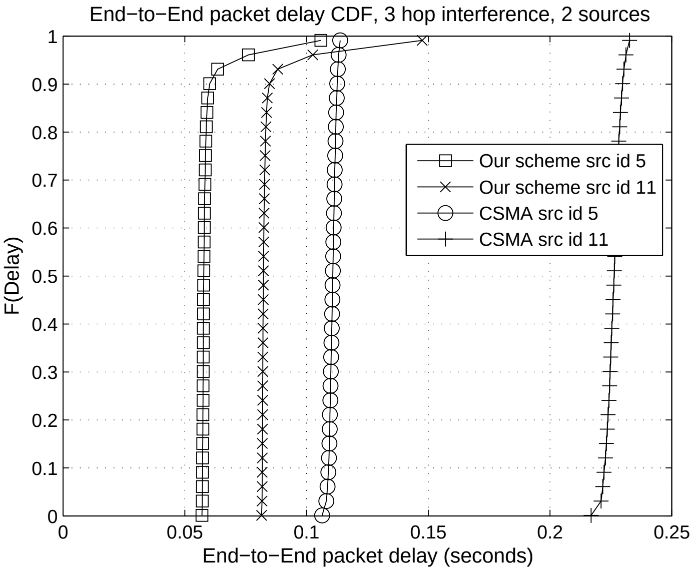

Figure 9: End-To-End delay for system with 2 sources and 3 hop interference range

Table 3: End-to-end delay (mean ± standard deviation)

		CSMA	E-MAC
2 hop interference	src 5	0.1002s ± 0.0011	0.0503s ± 0.0027
	src 11	0.2104s ± 0.0024	0.0748s ± 0.0027
3 hop interference	src 5	0.1108s ± 0.0014	0.0598s ± 0.0079
	src 11	0.2262s ± 0.0029	0.0847s ± 0.0104

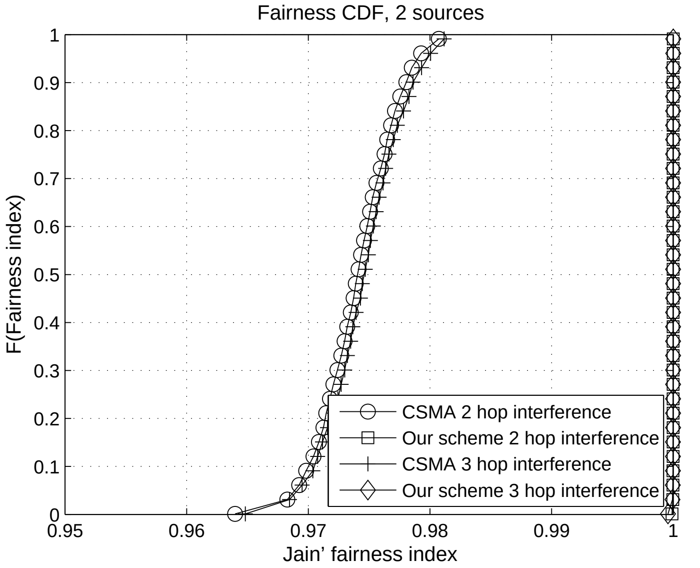

Figure 10: Fairness for system with 2 sources

Table 4: Fairness (Jain's Fairness Index) (mean ± standard deviation)

	CSMA	E-MAC
2 hop interference	0.9740 ± 0.0031	$\sim 1 \pm 0.00000797$
3 hop interference	0.9745 ± 0.0032	$\sim 1 \pm 0.00002833$

scenario where the specified number of source nodes are placed at the end of the chain. These results are shown in Figures 11 and 12, which exhibit the same trends as the previous results. Note that the throughput results for the two node case in Figure 11 differ slightly from the results presented in Table 2, due to the different placement of source nodes in the original topology (where one of the two source nodes is located in the middle of the chain). E-MAC clearly outperforms CSMA RTS/CTS and performance reaches very close to theoretical boundaries in the scenario. We can see a sudden variance in CSMA RTS/CTS fairness results. Even with a fair queuing policy CSMA RTS/CTS seems to become unstable once a clear dominating node appears in the network. Under 10 source operation, essentially every-node in the network is a source. The source closest to the sink is only 1 hop away. This source, due to its success and quick delivery, starts over-dominating the network, thereby operating as a single hop (breaking throughput bounds) and blocking out other transmissions (significant drop in fairness).

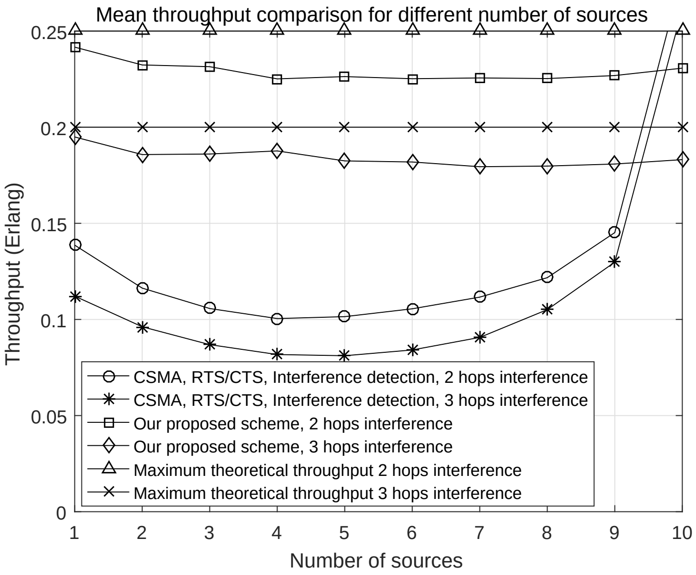

Figure 11: Mean throughput comparison for different numbers of sources

Overall, we have observed significant performance benefits of E-MAC over CSMA in two measured performance criteria (throughput and end-to-end delay) and better performance for fairness. The simplicity of E-MAC, in terms of

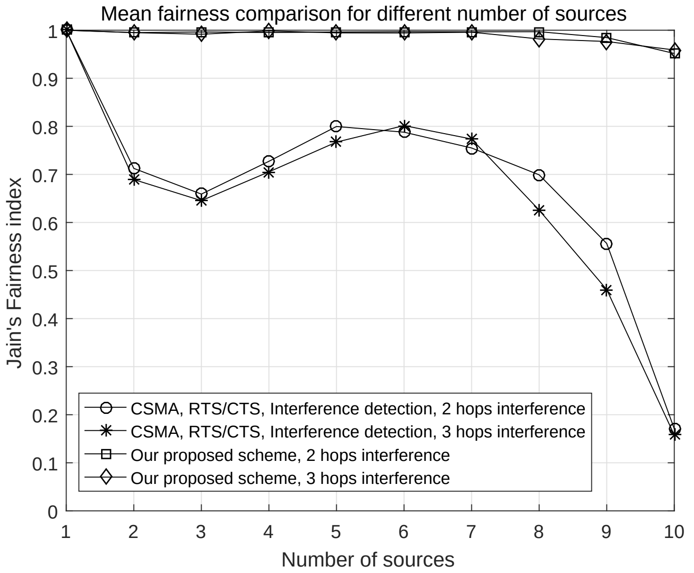

Figure 12: Mean fairness comparison for different numbers of sources

hardware and computational requirements, is truly encouraging. The basis for this is the exploitation of emergence through simple exchanges of information piggy-backing an otherwise trivial MAC protocol. The network is able to self-organise and adapt to different scenarios without requiring extra parameters or a shift in the simple agent behaviours. Emergence provides us with indirect synchronisation which boosts throughput and reduces end-to-end delay. Furthermore, the reduced number of collisions improves overall fairness.

5.3. Parameters

Earlier, we stated that a property of biological systems is a lack of scenario-specific parameter tunings. The reader will have noted that two parameters seem to abuse this notion in E-MAC: α and $change\_scale$.

Figure 13 and 14 are contour plots which show the variation of throughput when α and $change\_scale$ are varied. Actual $change\_scale$ values are related to packet length. It is clear that performance is generally insensitive to these parameter values. However some trends can be observed.

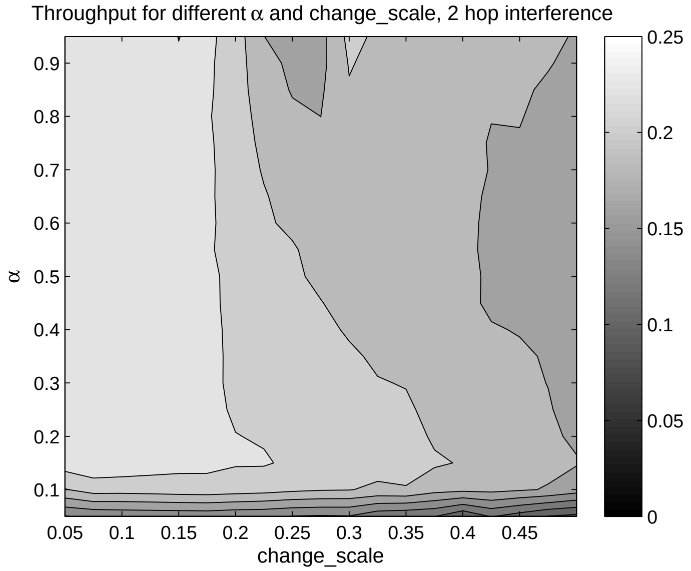

Figure 13: Throughput variation when α and $change\_scale$ are varied, 2 hop interference

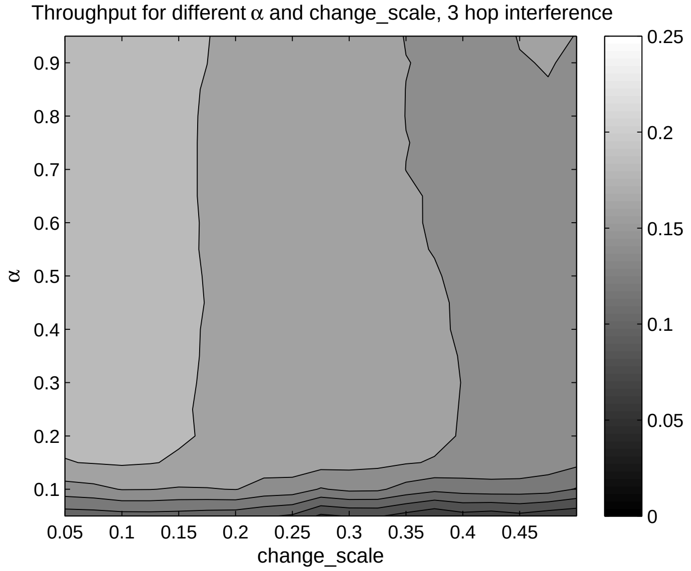

Figure 14: Throughput variation when α and $change\_scale$ are varied, 3 hop interference

We have set $change\_scale$ values to be 10% of packet length, where E-MAC performs well. Greatly increasing the value causes the resolution of Transmission Delay to be too coarse so that E-MAC does not perform well. An excessively low value causes very slow settling and adaptive response.

We also see from Figure 13 and 14 that the value of α should be in the general region of 0.2. Choosing more extreme values will cause the transmission delay to settle very slowly (low α), or away from a value commensurate with good throughput and reduced ability to adapt (high α). In fact when α approaches value of 1, the Robbins Monro algorithm no longer tracks past values and essentially only line 12 in Algorithm 1 remains active. The protocol will only respond to the last acknowledgement outcome, leading to unstable behaviour.

The same observations can be seen in the end-to-end delay performance for different parameter values given in contour plots in Figures 15, 16, 17 and 18.

It is important to note from the contour plots that, given almost *any* values for these parameters, in the scenarios presented, E-MAC will perform better than CSMA.

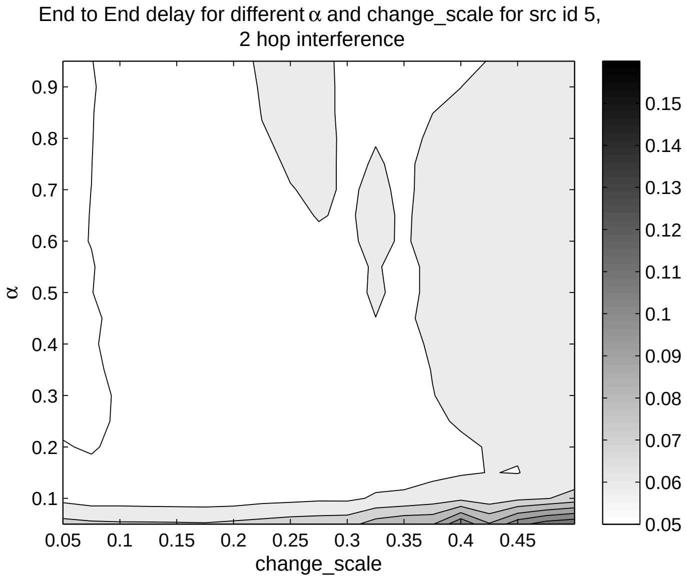

Figure 15: End-to-end delay variation from source 5 when α and $change\_scale$ are varied, 2 hop interference

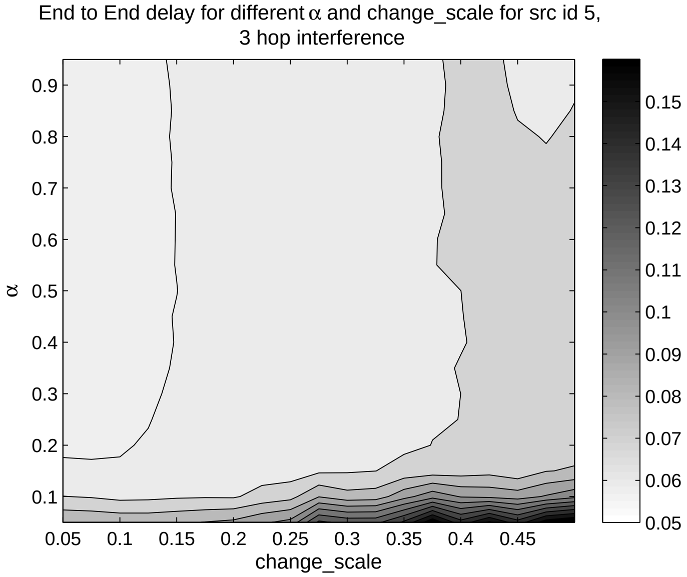

Figure 16: End-to-end delay variation from source 5 when α and $change\_scale$ are varied, 3 hop interference

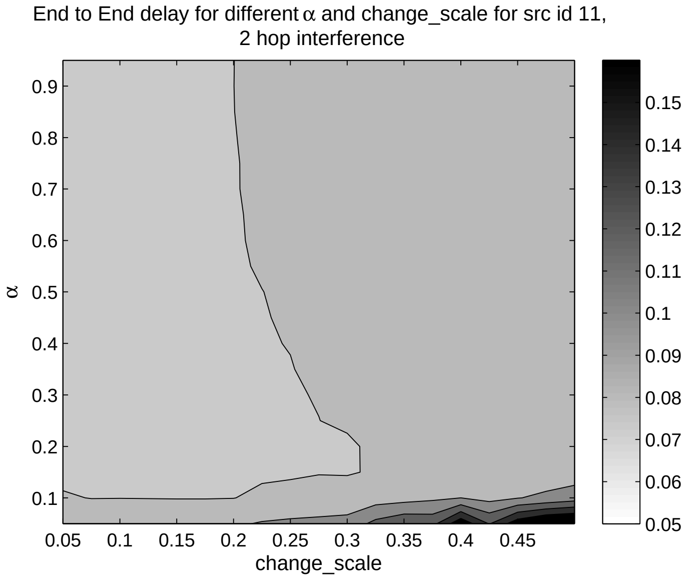

Figure 17: End-to-end delay variation from source 11 when α and $change\_scale$ are varied, 2 hop interference

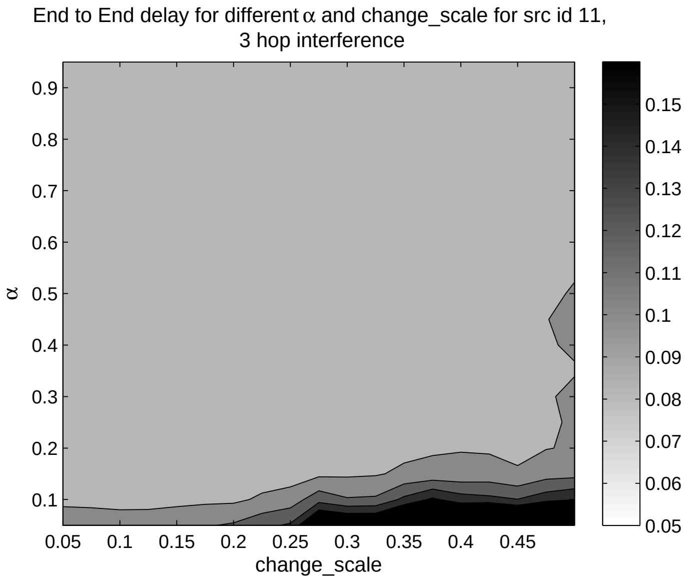

Figure 18: End-to-end delay variation from source 11 when α and $change\_scale$ are varied, 3 hop interference

6. Conclusion

We have discussed some notions derived from a biological metaphor and applied them to the development of a new type of MAC protocol for WSNs. E-MAC follows very simple rules based on the reaction of social agents to the intensity of a localised environmental stimulus. Without explicit synchronisation and using very simple hardware it is able to out-perform its comparator, the widely-known IEEE 802.11 CSMA/CA RTS/CTS scheme. Throughput, end-to-end delay and fairness were compared using multi-hop chain networks. E-MAC exhibits self-organisation, flow control on both hop-by-hop and end-to-end basis, indirect synchronisation between the nodes as packets are relayed and minimal latency. Its parameter insensitivity means that it can be adopted in different environmental conditions without the need for specific set-up tuning.

7. Acknowledgements

We would like to acknowledge EPSRC (Engineering and Physical Sciences Research Council) for funding this research.

References

[1] E. Maltby, C. Legg, M. Proctor, The ecology of severe moorland fire on the north york moors: effects of the 1976 fires, and subsequent surface and vegetation development, The Journal of Ecology (1990) 490–518.

[2] K. Sha, W. Shi, O. Watkins, Using wireless sensor networks for fire rescue applications: Requirements and challenges, IEEE International Conference on Electro/information Technology (2006) 239–244.

[3] S. Du, A. K. Saha, D. B. Johnson, Rmac: A routing-enhanced duty-cycle mac protocol for wireless sensor networks, INFOCOM. 26th IEEE International Conference on Computer Communications (2007) 1478–1486.

[4] T. Van Dam, K. Langendoen, An adaptive energy-efficient mac protocol for wireless sensor networks, ACM Proceedings of the 1st international conference on Embedded networked sensor systems (2003) 171–180.

[5] I. Demirkol, C. Ersoy, F. Alagoz, et al., Mac protocols for wireless sensor networks: a survey, IEEE Communications Magazine 44 (4) (2006) 115–121.

[6] C. Li, H. Zhang, B. Hao, J. Li, A survey on routing protocols for large-scale wireless sensor networks, Sensors 11 (4) (2011) 3498–3526.

[7] J. N. Al-Karaki, A. E. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE wireless communications 11 (6) (2004) 6–28.

[8] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebber, M. D. Fricker, K. Yumiki, R. Kobayashi, T. Nakagaki, Rules for biologically inspired adaptive network design, Science 327 (5964) (2010) 439–442.

[9] F. G. Mármol, G. M. Pérez, Providing trust in wireless sensor networks using a bio-inspired technique, Telecommunication systems 46 (2) (2011) 163–180.

[10] R. V. Kulkarni, G. K. Venayagamoorthy, Bio-inspired algorithms for autonomous deployment and localization of sensor nodes, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40 (6) (2010) 663–675.

[11] P. G. Espejo, S. Ventura, F. Herrera, A survey on the application of genetic programming to classification, IEEE Transactions on Systems, Man, and Cybernetics, Part C 40 (2) (2010) 121–144.

[12] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm intelligence 1 (2007) 33–57.

[13] S. Garnier, J. Gautrais, G. Theraulaz, The biological principles of swarm intelligence, Swarm Intelligence 1 (2007) 3–31.

[14] J. Camhi, G. Sumbre, G. Wendler, Wing-beat coupling between flying locust pairs: preferred phase and lift enhancement, The Journal of experimental biology 198 (4) (1995) 1051–1063.

[15] M. Dorigo, E. Bonabeau, G. Theraulaz, Ant algorithms and stigmergy, Future Generation Computer Systems 16 (8) (2000) 851–871.

[16] E. Bonabeau, Social insect colonies as complex adaptive systems, Ecosystems 1 (5) (1998) 437–443.

[17] E. Bonabeau, G. Theraulaz, J.-L. Deneubourg, S. Aron, S. Camazine, Self-organization in social insects, Trends in Ecology & Evolution 12 (5) (1997) 188–193.

[18] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm intelligence: from natural to artificial systems, no. 1, Oxford university press, 1999.

[19] R. C. Plowright, C. M. Plowright, Elitism in social insects: a positive feedback model, Westview Press, 1988.

[20] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L. Chrétien, The dynamics of collective sorting robot-like ants and ant-like robots, in: Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats, 1991, pp. 356–363.

[21] H.-F. Chen, Robbins-Monro Algorithm, Stochastic Approximation and Its Applications, ser. Nonconvex Optimization and Its Applications, Springer, vol. 64, pp. 1-24, 2002.

[22] I. Rhee, A. Warrier, M. Aia, J. Min, M. L. Sichitiu, Z-mac: a hybrid mac for wireless sensor networks, IEEE/ACM Transactions on Networking (TON) 16 (3) (2008) 511–524.

[23] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies., Vol. 3, IEEE, 2002, pp. 1567–1576.

[24] M. Yigit, E. A. Yoney, V. C. Gungor, Performance of mac protocols for wireless sensor networks in harsh smart grid environment, in: First International Black Sea Conference on Communications and Networking (BlackSeaCom), IEEE, 2013, pp. 50–53.

[25] C. Suh, Y.-B. Ko, A traffic aware, energy efficient mac protocol for wireless sensor networks, in: 2005 IEEE International Symposium on Circuits and Systems, IEEE, 2005, pp. 2975–2978.

[26] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd annual Hawaii international conference on system sciences, 2000., IEEE, 2000, pp. 10–pp.

[27] R. Jain, D.-M. Chiu, W. R. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer system, Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 38 (1984).