Allocating Sensor Network Resources Using an Auction-Based Protocol

Jarrod Trevathan[1], Lee Hamilton[2] and Wayne Read[3]

Griffith University, School of Information and Communication Technology, Brisbane, Australia,
[1] j.trevathan@griffith.edu.au, [2] lee.hamilton@griffithuni.edu.au
[3] James Cook University, School of Engineering and Physical Sciences, Townsville, Australia,
wayne.read@jcu.edu.au

Abstract

Wireless sensor networks are being increasingly used for remote environmental monitoring. Despite advances in technology, there will always be a disparity between the number of competing sensor devices and the amount of network resources available. Auction-based strategies have been used in numerous applications to provide efficient/optimal solutions for determining how to fairly distribute system resources. This paper investigates the suitability of using online auctions to allow sensors to acquire preferential access to network resources. A framework is presented that allocates network priority to sensor devices based on their characteristics such as cost, precision, location, significant changes to readings, and amount of data collected. These characteristics are combined to form the value for a particular sensor's bid in an auction. The sensor with the highest bid wins preferential access to the network. Priority can be dynamically updated over time with regard to these characteristics, changing conditions for the phenomenon under observation, and also with input from a back-end environmental model. We present an example scenario for monitoring a flood's progress down a river to illustrate how the proposed auction-based system operates. A series of simulations were undertaken with a preliminary auction structure to examine how the system functions under different conditions.

Keywords: Sensor network, Online auction, Prioritization, Resource allocation, E-Commerce

1   Introduction

Effective remote monitoring of sensitive ecosystems is vital to ensuring their sustainability into the future. *Wireless sensor networks* (WSNs) are gaining popularity for use in environmental measurement and monitoring initiatives due to their versatility and scalability. Advances in technology have led to high-speed broadband, cloud storage, pervasive computing, and semantically-enabled networked devices (i.e., "The Internet of Things") [3]. This makes sensor network systems more affordable and ubiquitous, thereby allowing the collection of more accurate and larger datasets on phenomena under study. Environmental scientists are set to greatly benefit from the increased knowledge brought about by the data gathered through sensor networks, which will facilitate the development of effective management policies – particularly those that address natural disasters.

However, despite increases in network capacity (i.e., *bandwidth*), the number of sensor devices also grows proportionally, which maintains an inequality between required bandwidth and actual bandwidth for transmitting large amounts of sensed data. This problem continues to limit the potential of environmental monitoring systems because the lack in required capacity leads to data loss at the point of collection. Therefore, dynamic decisions must be made about the allocation of scarce network resources amongst the competing devices. This predicament cannot be addressed through advancements in hardware technology alone – a smart software solution is required.

Trevathan et al. [24] describe an initiative to create intelligent, low-cost WSNs for environmental monitoring applications (referred to as *Smart Environmental Monitoring and Analysis Technologies* (SEMAT)). The SEMAT WSN infrastructure allows sensors to interface with a back-end environmental model in near real-time to facilitate reasoning over the collected data. As such, sensors can be dynamically re-tasked and the environmental model can evolve as new data is collected by the system. This enables the WSN to give priority to sensors collecting the most relevant data regarding a phenomenon, rather than being deluged with unimportant details. It is possible to integrate an intelligent WSN's infrastructure with an auction-based mechanism for resource allocation. Through dynamically updating sensor priorities, it is envisaged that the system could be extended to enable phenomenon tracking, which has potential applications in natural disaster management, environmental monitoring, and security.

This paper examines the applicability and suitability of using online auctioning mechanisms to improve resource-scheduling outcomes for WSNs. We present a basic framework for using auctions to allocate resources in a low-cost, intelligent SEMAT WSN [24]. Priority is allocated to devices based on their characteristics such as cost, precision, importance of location, significant changes to regular readings, and amount of data collected in terms of whether a sensor device's memory buffer is full. Individual priorities are dynamically updated over time (i.e., after each auction) with regard to these characteristics, changing conditions for the phenomenon under observation, and also with input from a back-end environmental model. This allows the WSN to potentially track the progress of a phenomenon throughout a geographical area [1], [2]. Note that this is different to other market-based approaches in that the auction process is linked to an environmental model, which can significantly bias the outcome in favor of certain bidders (sensors). Auction-based strategies for task allocation have been addressed in the literature before [9], [15]-[17], [27]. However, this is purely from a theoretical economic perspective. This paper focuses on the practical implications for an auction-based allocation scheme including the types of possible auctioning structures, the protocol, and the practical issues with implementing such a system using existing network technologies. We present an example scenario that simulates a flood moving down a river to illustrate how the system operates.

The paper is organized as follows: Section 2 presents the problem motivation and related work. Section 3 describes the framework for using an auctioning mechanism for resource allocation in sensor networks. Section 4 discusses the methodology and construction of the proposed system. Section 5 presents the results from simulations to evaluate the system's performance. Section 6 provides some concluding remarks.

2   Problem Motivation and Related Work

This section outlines the original problem motivation and existing literature related to our work.

2.1   Using E-Commerce for Dynamic Resource Allocation

Previous SEMAT [24] environmental monitoring deployments have been based on a hierarchical WSN model (see Figure 1). In the SEMAT WSN there is a central base station that communicates with a number of sensor devices. A sensor device in this context refers to an apparatus that takes a specific reading on one particular phenomenon (e.g., light, temperature, salinity, pressure, etc.). Each device has some form of onboard processing power, storage capacity and communication ability. For example, in the SEMAT deployments each sensor device is connected to a Gumstix computer-on-module (Site 1) or an Arduino micro controller (Site 2) that utilize wi-fi to communicate to the base station. A unique characteristic of SEMAT model is the presence of a back-end environmental model that consumes the collected data. This allows the environmental model to be updated over time as the data is collected (e.g., update conditional probability tables in a Bayesian network). For the purposes of this paper, it is assumed that

Jarrod Trevathan
Lee Hamilton
Wayne Read

the system has real-time, two-way communication between the base station and the sensor devices. Furthermore, it is assumed that power management is not a concern, and all network issues (e.g., routing, etc.) have been resolved.

Consider an example application with a safety critical sensor network for flood monitoring that contains thousands of devices across a river system. While each sensor may be trying to send data over the network (to a base station) about an approaching flood, the system must decide which sensors are the most important to allocate priority network resources to. If each sensor is simultaneously transmitting, network congestion will ensue. This directly impacts the timely dissemination of critical information. In a flood situation such information might dictate who should be evacuated first, which locations rescue crews should be assigned to, or what areas of economic interest should be protected first. (This scenario is applicable to bushfires, tsunami, earthquakes, lava flow, people/animal tracking, oil spills, bio-threat monitoring, etc.)
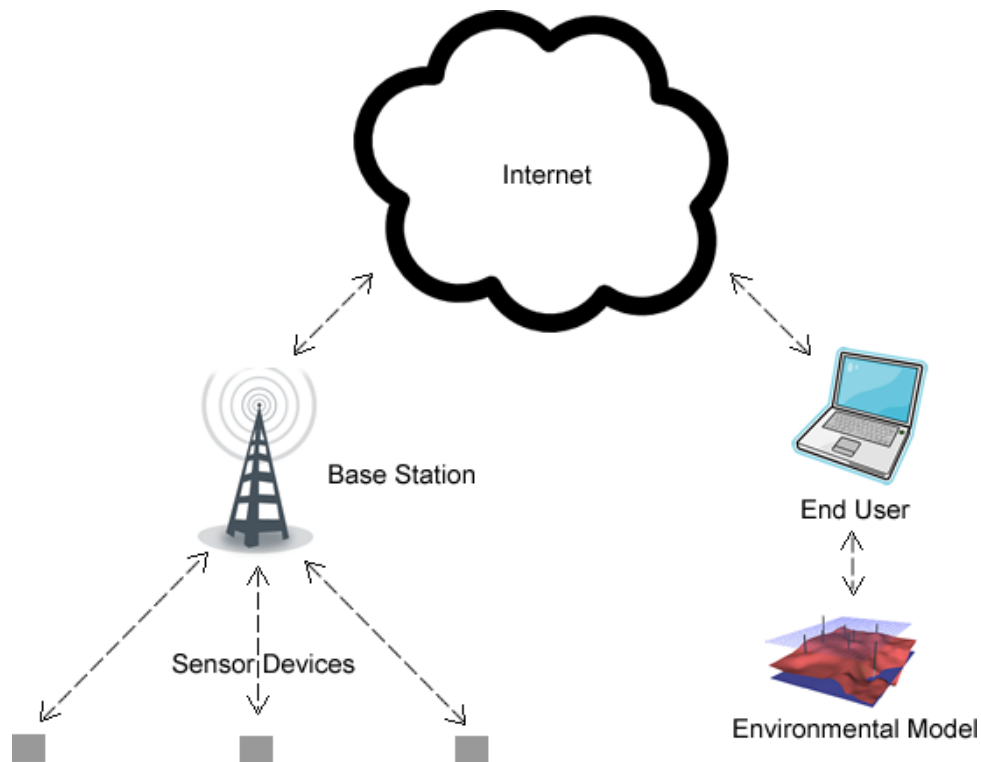

Figure 1: The intelligent environmental sensor network infrastructure

At present the only approach is to assign priorities before deploying devices [22]. For example, if there is a house in a particular area, sensors near the house are given higher priority, and therefore have first access to bandwidth. However, this does not allow dynamic updating of priorities in response to changing conditions. The other option is to wait until data is received at a centralized source, have a program or human make sense of it, and then reconfigure the sensors' priorities remotely (assuming the system is able to support remote operation). As such there are two types of sensor network deployments to consider:

- Static – These are one-off systems that will be deployed for a set period and then removed. No changes to the system in any form will occur during the period.

- Dynamic – A long-term sensor network deployment where new devices or modification to existing devices occurs. Furthermore, existing sensor devices may also be removed.

Static deployments establish the network allocation policy apriori. Therefore, the auction-based approach is not really relevant in their context. Instead, this paper will focus on dynamic deployments.

Consider the new upcoming generation of sensors that can do some data processing and have intermediate storage (as in the SEMAT project [24]). Access to the network resources could be modelled economically where the commodity being sought is bandwidth. The currency is *priority* – or the right to have first access to the bandwidth. The issue to be resolved is how to allow a sensor to determine what priority it has either independently and/or have its priority dynamically updated by the base station depending on the sensor properties (i.e., cost, precision, location, changes in readings, buffer size) and the observed actions of the phenomenon under study.

Assume all sensors are the same type and cost (e.g., all $20 water level sensors) and have onboard computing power. If one sensor starts reading significant changes to what it has regularly been monitoring, then its priority

43

Jarrod Trevathan
Lee Hamilton
Wayne Read

should increase. For example, if the flood raises the water level on a particular sensor dramatically, then it gains higher priority than another sensor that has not read any abnormal changes. This facilitates an auction-based approach where the highest priority sensor gets the first available access to bandwidth. Left over bandwidth then goes to the sensor with the next highest priority, etc.

Not all sensors are equal in terms of type, cost and position. In general, expensive sensors should be more accurate and therefore should have a higher priority. Furthermore, certain types of sensors might be more useful or suitable for tracking a given phenomenon than others, and should be given a higher priority. If there is a known model (e.g., a hydrodynamic model) that shows which way floods will most likely move, then sensors in certain positions should be given higher priority as they relate to the model.

Additionally, storage at the sensor-level is finite. If a sensor is unable to transmit its data in a timely manner, data loss will occur. This is either in the form of overwriting older data (i.e., the buffer employs a circular storage approach), or the sensor may not store additional readings (i.e., some loggers are designed to deactivate when their memory becomes full). Therefore, if a sensor device has a full memory buffer, or a buffer that has a significant amount of data, ideally that sensor device's priority should increase. Other devices with less data in proportion to their total buffer size could potentially afford to wait until a later time before transmitting their data.

Furthermore, as a flood is a dynamic phenomenon, its location (or front) will be constantly changing. The flood will sometimes be moving towards and then away from specific sensors (and in some cases even destroying them!). Given this scenario, priorities could presumably decrease for sensors that the flood has moved away from thereby freeing up access to those sensors that are coming into the path of the flood. A further complication to the flood scenario could be when there is more than one phenomenon of interest. For example, there may be two floods moving throughout an area. Sensors close to the flood need to obtain higher priorities, so now we have several sets of sensors competing for priority access to the network. Conceivably a prioritization scheme needs a way to both allocate and de-allocate resources within a sensor network.

2.2   Related Work - Network Congestion Management and E-Commerce Strategies

Prioritization techniques for event tracking are ultimately a subset of network congestion management. Congestion occurs when the amount of information being transmitted throughout a network exceeds the available resources. Congestion causes lost data from buffer overflows (i.e., *node-level* congestion) and/or dropped packets from transmission errors (i.e., *link-level* congestion). Congestion results in low data throughput, low quality of service, and excessive power consumption due to retransmission attempts. While congestion management has been studied in-depth for traditional networks, WSNs present new challenges given the finite nature of memory/computation and the environmental conditions under which sensor networks are deployed.

Congestion management approaches can be considered as either being *active* or *reactive*. In an active scheme, the system monitors its throughput status and tries to ensure that congestion does not occur. A reactive scheme deals with congestion once it has occurred. Such schemes can also be classified according to whether they are *global* or *local*. In a global scheme there is a central authority that is monitoring and managing congestion. Alternately, a local approach allows nodes within the network to collaborate and manage congestion at a local level. The type of approach employed could depend on the size of the network, the computational capabilities of the network nodes, and the application for which the network is being employed.

A key concept for congestion management is a *feedback loop*. Consider an analogy whereby a commuter is driving along a busy motorway. An authority is monitoring the motorway via video cameras and has a series of electronic sign posts at various locations throughout the motorway to notify motorists of important information. If the authority observes a crash that is resulting in traffic congestion for several kilometers, then the authority can notify motorists heading towards the affected area that there is congestion ahead and to proceed with caution. This analogy illustrates the function of a feedback loop. In a network sense, there is an application that is monitoring the system which provides information back to the network about the congestion status. This could operate at a global level (i.e., the motorway example) or at a local level whereby nodes transmit information about local conditions.

Chen and Yang [6] propose a congestion avoidance scheme based on light-weight buffer management. The approach adapts the sensor's forwarding rates to reduce congestion. They implement the buffer-based congestion avoidance with different *Media Access Control* (MAC) protocols including *Carrier Sense Multiple Access* (CSMA) and *Time Division Multiple Access* (TDMA). Their scheme is based on local feedback from intermediate nodes that use Hidden Markov Models and other probabilistic techniques. They evaluated their scheme via simulations.

Lim et al. [14] propose the *Adaptive Distributed Resource Allocation* (ADRA) scheme, an adaptive approach for distributed resource allocation in WSNs. Their scheme uses local actions performed by individual sensor nodes for node management. Each node adapts its operation over time in response to the status and feedback of its neighboring nodes. They study the effectiveness of the ADRA scheme for sensor mode management in an acoustic sensor network to track vehicle movement. The scheme is evaluated via simulations, and also prototyped it using the Crossbow MICA2 motes. They claim the ADRA scheme provides a good trade-off between performance objectives such as coverage area, power consumption, and network lifetime.

Jarrod Trevathan
Lee Hamilton
Wayne Read

Wang et al. [26] propose an upstream congestion control protocol for WSNs, called *Priority-Based Congestion Control Protocol* (PCCP). PCCP measures congestion degree as the ratio of packet inter-arrival time along over packet service time. Wang et al's proposal can be considered as a local scheme, however, it does not appear that any specific communication happens amongst nodes regarding congestion. They use simulations to test the performance of PCCP for scenarios involving single-path and multi-path routing.

Shi et al. [21] propose the receiver-assisted congestion control mechanism in which the sender performs loss-based control, while the receiver is performing delay-based control. This is a form of local congestion management where the two neighboring nodes (i.e., the sender and the receiver) work in unison to avoid congestion. They conducted a series of simulations and claim that their proposed scheme can outperform TCP/IP in high-speed and lossy environments. Their scheme does not utilize any form of auctioning.

Chen et al. [4], [5] examine congestion management in large-scale sensor network tracking applications using e-commerce mechanisms. The technique employs a *distributed second price auction-based* approach. A congested node acts as an auctioneer and the packets carrying target updates act as bidders. Each packet bids based on its predicted utility loss that will incur if it is delayed. The winning packet receives the current transmission slot of the auctioneer node. They conduct a series of simulations that are quantified according to a metric referred to as loss of information utility, which refers to how valuable the received information is to the application at that point in time.

Shah and Szymanski [18], [19] introduce the concept of a *travelling auction* for dynamic price-based routing to address transient congestion. A congested node acts as an auctioneer and runs an auction for each available transmission slot. Each packet bids its predicted quality of information loss arising from delay. The auctioneer can minimize the total quality of information loss for an application, or equalize the loss across all applications.

The existing congestion and object-tracking proposals focus on localized schemes. There is limited coverage on a global reactive approach. Schemes where prioritization is allocated apriori do not discuss how such allocation occurs or why. This paper investigates what auction structures are suitable for use in alleviating congestion. We also seek to integrate an environmental model to help inform the auctioning process on which sensors are most important at a particular point in time for the application.

3   Using Online Auctions for Resource Allocation in Sensor Networks

This section discusses the challenges for using auctioning mechanisms in WSNs and presents the justification for the auction structure proposed in this paper.

3.1   Contrasting Auction Types

Some of the major types of auctions include:

- English [10] – The price is competitively bid upwards until no new bids are submitted after a given time-out period (referred to as an open bid auction). The winner is the bidder with the highest price (referred to as a first price auction).

- Vickrey [25] – Bids are submitted in secret (referred to as a sealed bid auction). The winner is the bidder with the highest price. However, s/he only pays the second highest price for the item.

- Dutch [7] – The price is continually offered downwards by the seller (referred to as a reverse auction). The winner is the first bidder to accept the going price. This style of auction is commonly used as a tendering process for large projects.

- Continuous Double Auction (CDA) [8], [23] – There are many buyers and sellers who are continuously trading multiple items (e.g., a stock exchange).

In terms of WSN resource allocation, several rounds of bidding are not required as in an English auction. Having several rounds of bidding would complicate the system and use limited resources to conduct. There is no requirement that bids remain secret as in a Vickrey auction, therefore bids do not need to be sealed. Dutch auctions do not seem applicable in the context of WSNs and also require several rounds of bidding. While a CDA is the most versatile in obtaining the functionality raised in Section 2, it has more complicated rules than the other auction types and imposes more significant technical challenges for a WSN.

Due to the aforementioned reasoning, the auction type chosen for this paper is a form of first price, open bid auction (a mixture of English and Vickrey). Each sensor device can only bid once during an auction (i.e., the auction is a

Jarrod Trevathan
Lee Hamilton
Wayne Read

one-off procedure.). The bidder with the highest price wins first priority to network resources, the next highest bidder gets second preference, etc.

3.2 Auction Structures

As sensor networks can be constructed using various topologies, the exact structure of the auction can vary. The following are the structures that could facilitate the aforementioned auction types:

- Centralized auction house – There is a central auctioneer, A, to whom bidders *(b1, b2, ..., bn)* submit bids to (Figure 2 (a)).

- Distributed auction house – There are several auction houses, A1 and A2 (Figure 2 (b)). Each auction house could be responsible for a particular section of the WSN, or can be used in unison for implementing cryptographic protocols for security in electronic auctions [23].

- Peer auction house – The peer approach is the alternate extreme where all bidders can exchange resources amongst themselves (e.g., a clearing house or CDA) (Figure 2 (c)).

- Hybrid auction house – In a large WSN the entire community could be made up of all the aforementioned structures. The individual auctions could also be of varying auction types (e.g., English, CDA, etc.) (Figure 2 (d)).
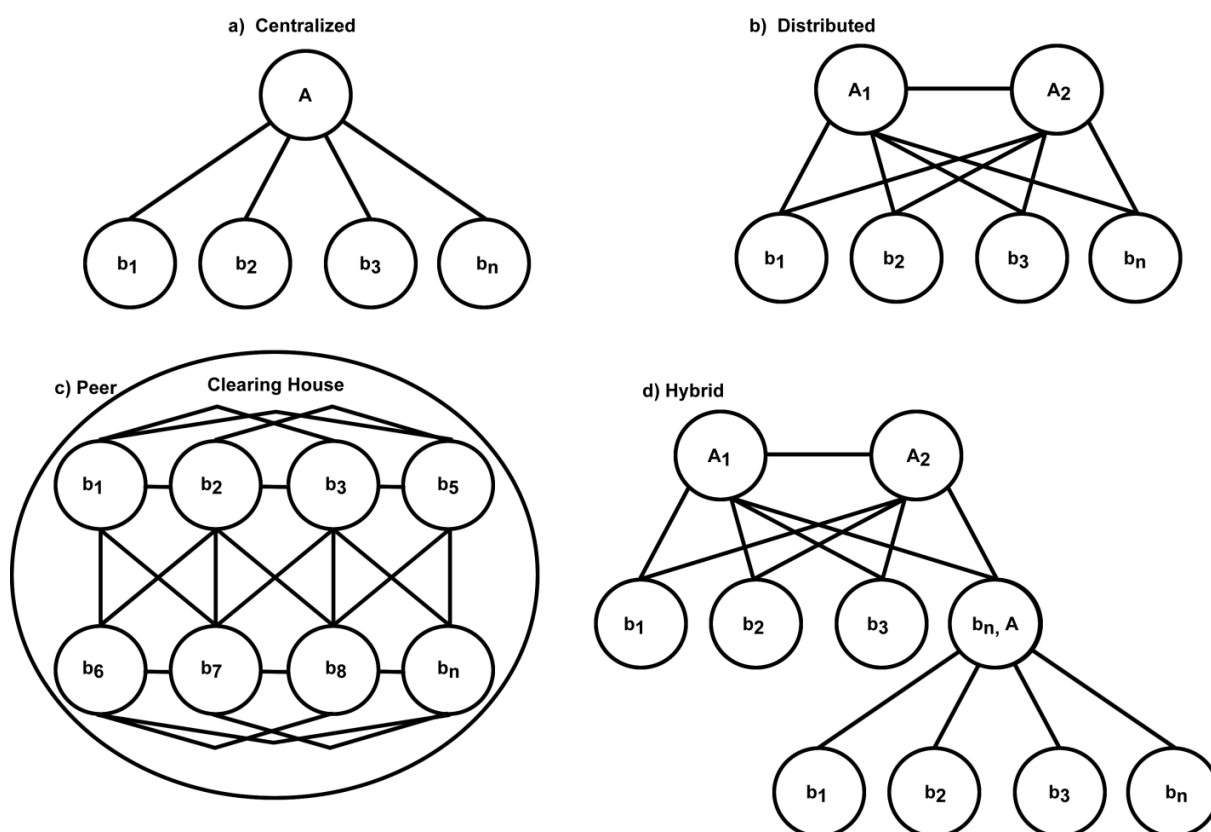

Figure 2: Differing auction structures that could be employed for a sensor network

For this paper, a SEMAT-style [24] deployment is used (e.g., hierarchical with a single base station and multiple sensor devices possessing some storage and computational ability). To keep things simple, the auction structure is a centralized auction house, where the auctioneer is the base station and sensor devices are the bidders. This approach facilitates a global, reactive congestion and object-tracking mechanism.

3.3 Auction Phases

There are several phases common to all auction types. These include initialization (where the auction is set up), bidding (bids are submitted), and winner determination. Some auction types such as CDAs perform these processes repeatedly. The following describes how these processes relate to sensor networks and the proposed system:
Initialization – There are three initialization types in a dynamic deployment with regard to sensor devices:

Jarrod Trevathan
Lee Hamilton
Wayne Read

1. Deploying all sensor devices as part of establishing the WSN;

2. Deploying new sensor devices throughout the WSN's life span; and

3. Modifying existing sensor devices

Bidding – Sensor devices submit bids to the auctioneer.

Winner Determination – The auctioneer determines the winner and allocates network resources accordingly.

With regard to this paper, initialization is performed as a one-off process at the outset of the WSN's deployment. We will not consider physical additions or modifications to the WSN after deployment. The only modification is updating each device's priority for network resources (hence this is really a semi-dynamic deployment). A series of auctions will be conducted throughout the deployment's life at predefined intervals. After winner determination, sensors will have their priorities updated with regard to their placement in the auction process. Note that all sensors will continue to transmit their sensor data in-between auction rounds. However, an individual sensor's priority to transmit will change from round to round. The specifics of this process are described in the next section.

4   An Auction-Based Protocol for Allocating Sensor Network Resources

Figure 3 illustrates the WSN auction structure. There are multiple sensor devices (i.e., the bidders). The base station (i.e., the auctioneer) collects bids submitted from the sensor devices. During the winner determination stage, the base station consults with a back-end environmental model to determine the priority ordering for the sensors.
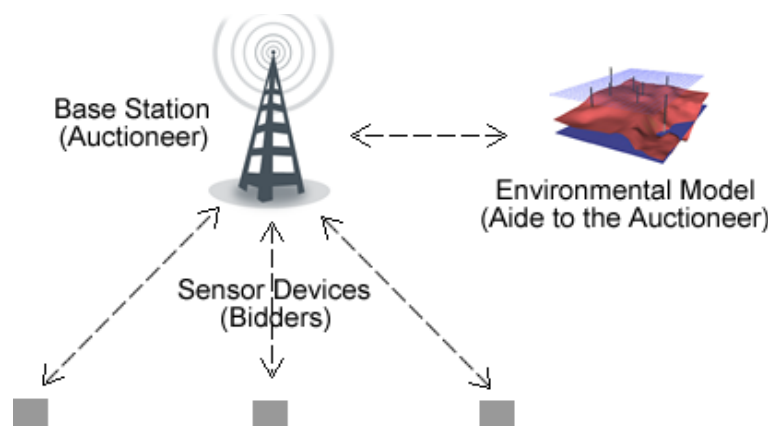

Figure 3: The proposed WSN auction structure

4.1   Mathematical Model

The goal for the simulated results (presented in Section 5) is to observe how the system performs while monitoring a flood's progress as it moves down a river (see Figure 4). Conceivably such a scenario could alert authorities to areas where people and property are in danger, or trigger flood gates to take action to avoid economic loss.

Geographical area *L* is divided into a *2*-dimensional grid, *(M, N)* where *(i, j)* denotes the location (positioning) of the cell in the *i*th row and *j*th column.

$$L = M \times N = \{(i, j), | i \in 1, ..., M, j \in 1, ..., N\}$$

(1)

The phenomenon under study (such as a flood or bush fire) is denoted as

$$P = \{(p_1, q_1), (p_2, q_2), ...\}$$
$$P \subseteq L$$

(2)

47

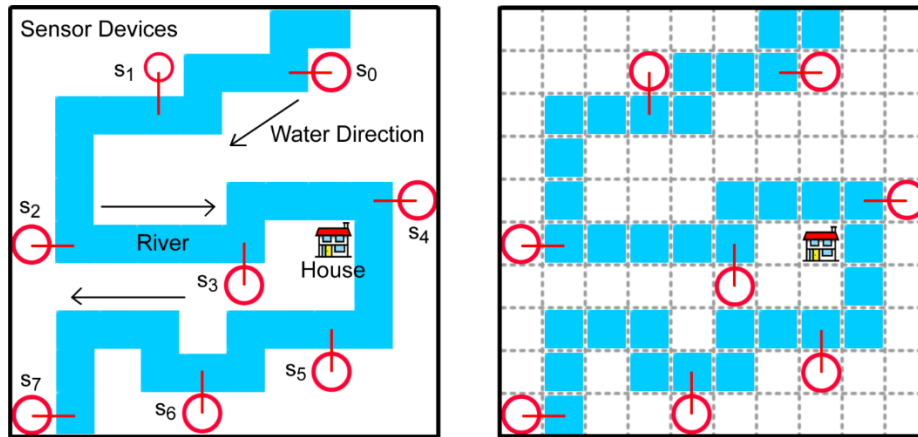Jarrod Trevathan
Lee Hamilton
Wayne Read

Figure 4: An example scenario with monitoring the progress of a flood down a river

In the proposed example a river *R*, flows through the area. Assuming that a flood that is moving down a river is completely contained within the river, then $P \subseteq R$. There is a set of sensor devices $S = \{s_1, s_2, ..., s_n\}$. s_i ($1 \le i \le n$) indicates the ith sensor's geographical location within *L*. The commodity up for auction is priority – the right for a sensor device to acquire first access to the network for the required resources it needs to satisfy its currently assigned peak load. All sensor devices have balance of currency, denoted as *$*, with which to bid for bandwidth. *$* is allocated to, or taken from devices by the bank based on a specific sensor device's characteristics.

4.2 Experimental Setup

In the proposed example there are 100 cells arranged in a square array (i.e., M = N = 10). The river occupies 34 cells with R = $\{(i_1, j_1), ..., i_{34}^R, j_{34}^R)\}$. The flood (or phenomena) is represented as a software agent. The flood occupies k cells with $k \le 34$, F = $\{(i_1^F, j_1^F), (i_2^F, j_2^F), ..., i_k^F, j_k^F)\}$. Note that F $\subseteq$ R. Time is denoted as T = $\{t_1, ..., t_i, ..., t_l\}$. Figure 3 shows the direction the river is moving in. The front or leading edge of the flood changes position over time. This is randomly allocated to be between 1 unit and 4 units per time measure. I is determined by how quickly the flood moves downstream (e.g., based on the random allocation of the leading edge during each t_i). The auction house is represented as a software agent that resides on the base station. An auction is conducted for each unit of time. j is bounded by how fast the flood progresses down the river. The simulation terminates once the flood has progressed past point p_m in L.

In this paper we assume that all sensor devices are homogeneous and sample the same thing (i.e., water level). Each device takes samples according to a sensing interval, which is identical for all sensors in this example. Each sensor device is represented as a software bidding agent. There are eight sensor devices that occupy one square each (i.e., *n = 8*). We assume that two sensor devices cannot occupy the same location.

4.3 Sensor Characteristics

Each sensor device has the following characteristics:

- Precision α ($0 \le \alpha \le 1$) – This refers to how many significant digits the device measures. The sensor device's importance will depend on the application (as evaluated by the environmental model).

- Cost β ($0 \le \beta \le 1$) – In general, given two devices that measure the same characteristic, the data collected from the more expensive device would be more valuable/accurate for the application. Cost can be normalized between the least expensive s_i^{min} and the most expensive s_i^{max} sensor devices as follows:

$$\beta = \frac{s_i - s_i^{min}}{s_i^{max} - s_i^{min}}$$

(3)

- Change in Readings γ ($0 \le \gamma \le 1$) – When a significant change to the currently observed readings occurs, this suggests that there has been a change in the phenomenon under study and therefore this sensor should be given attention. Each sensor collects one reading per time unit, R = $\{r_1, r_2, ..., r_j\}$. This can be represented by a threshold Δ which indicates how far from the mean reading the newly observed reading is: $r_i > r_{i-1} \pm \Delta$. For the purpose of this example, Δ is binary in that either there is a change or no change in the latest observed reading in time period t_i.

Jarrod Trevathan
Lee Hamilton
Wayne Read

- Priority δ (0 ≤ δ ≤ 1) – Sometimes a sensor's particular location may be deemed more important than the placement of other sensors. For example, a sensor near a house might be deemed to be particularly important in a safety critical application (such as a bush fire alert system). Priority is determined by a sensor's proximity to an economic asset, or its significance as determined by the environmental model (described in Section 4.4). Alternately, priority could be purchased between auction rounds by allowing a company to bid for the right for a particular sensor(s) to be given first preference to network resources.

- Quantity of Data Collected ε (0 ≤ ε ≤ 1) – Over time, a lower priority sensor may buffer a significant amount of data while it waits for its turn to transmit. Therefore, rather than lose the data, ε can gradually increase over time flagging that the sensor needs to transmit. Assume that each s_i has a buffer B = {b_1, b_2, ..., b_n} where b_j denotes the jth sensor reading and n is the buffer's capacity. J is the total number of readings for a sensor where J ≤ n. ε is calculated as follows: ε = J/n

The aforementioned five characteristics are combined to compute *$* as follows:

$$\$ = \frac{\omega_1\alpha + \omega_2\beta + \omega_3\gamma + \omega_4\delta + \omega_5\varepsilon}{\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5} \tag{4}$$

where ω_i (1 ≤ i ≤ 5) is the weighting associated with each rating. The bidder with the highest *$* has the highest priority. The weightings are not used in this paper; however, future work will involve determining the best values to assign to the weightings. For now, each weighting is given a value of *1* (i.e., all characteristics are weighted evenly).

4.4 The Environmental Model

The environmental model dictates the rules for the WSN and focuses on a specific set of features. For example, in the [24] deployments the environmental model was examining causal factors for algal blooms in marine environments [11]. The environmental model accepts input from the sensors in the form of sensor readings and also the *γ* and *ε* factors. The model uses its underlying algorithms to deduce values for *α, β* and *δ*. These are passed back to the auctioneer to evaluate *$* for each sensor device. The rule for the environmental model in the flood example is:

$ should be highest for the current s_i observing the phenomenon and the next one down river expecting to encounter it.

A decision-maker, a government agency, or a company with an interest in the data from the WSN can pay money to increase the priority of a sensor (or group of sensors). Furthermore, the environmental model can be extended by tracking algorithms such as those proposed by [1], [2], [13].

4.5 Implementing a Priority-Based Auction Scheme

The properties of the CSMA protocol can be used to implement a priority-based auction scheme. CSMA is a probabilistic media access control protocol that allows all devices to transmit at any time on a network medium. If two devices attempt to transmit at the same time, then a collision occurs. The devices are then forced to *back off* for a random time period and attempt to re-transmit later. This protocol can be used to implement prioritized access to the network. The priority of the device determines how long a device is forced to wait before attempting a retransmit. For example, there are two devices s_1 and s_2, where s_1 has higher priority. If a collision is detected, both stop transmitting and wait. As s_1 is of higher priority, the amount of time it has to wait before retransmitting is lower than that of s_2. The priority approach can be implemented using the *P*-persistent mode in CSMA/CD. When the sender is ready to send data, it checks continually if the medium is busy. If the medium becomes idle, the sender transmits a frame with a probability *p*. If the station chooses not to transmit (the probability of this event is *1-p*), the sender waits until the next available time slot and transmits again with the same probability *p*. This process repeats until the frame is sent or some other sender stops transmitting. In the latter case the sender monitors the channel, and when idle, transmits with a probability *p*, etc.

4.6 The Auction Protocol

Figure 5 illustrates the auction protocol and each phase is described below. Initialization is performed once at the deployment of the WSN. The remaining phases are performed repeatedly. There is one auction for each $t_i \in T$. Prior to each auction, the environmental model is updated with the sensor readings received in that round (which influences a sensor's *α, β, δ* values). There are two types of data transmitted by a sensor. The first is the sensor's readings. A sensor will attempt to transmit this data continually regardless of its priority. The second type of transmission is a sensor's bid data. A sensor's bid data is only transmitted as needed to conduct an auction round.

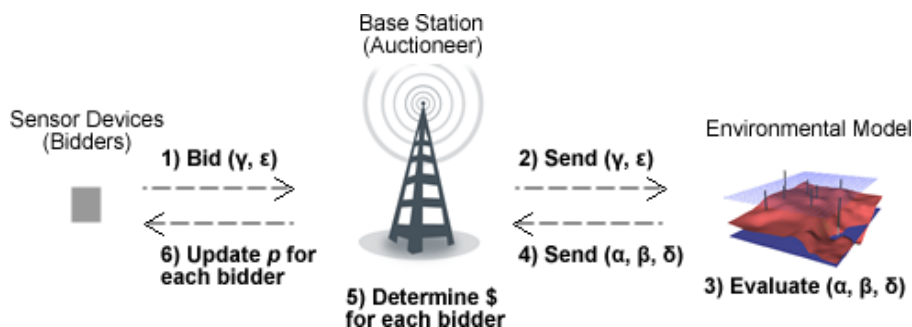Jarrod Trevathan
Lee Hamilton
Wayne Read

Figure 5: The auction protocol

The auction protocol contains the following steps within the three major auction stages:
Initialization – Each sensor device is assigned its $ value according to the environmental model's evaluation of (α, β, δ). γ and ε are initially excluded (i.e., set to 0) as no data has been sampled yet.

Bidding – Bidders submit (γ, ε) to the auctioneer (Step 1 in Figure 5). (γ, ε) is then forwarded to the environmental model (Step 2) and evaluated based on a sensor's importance to the model (Step 3). The environmental model determines (α, β, δ) and forwards this to the auctioneer (Step 4).

Winner Determination – The auctioneer combines the bid data received from each bidder and the environmental model's evaluation (i.e., (α, β, δ, γ, ε)) to determine $ for each bidder (Step 5). Each bidder is allocated p for its CSMA wait time as follows: p = 1 - $. p is sent to each particular sensor. Sensors then resume transmitting their sensor data according to their received priority. If a lower priority sensor transmits its sensor data when a higher priority sensor is trying to transmit, the lower priority sensor is forced to back off. The environmental model is updated by the new sensor readings it receives from the prioritized devices. This data influences future auction rounds based on its significance to the environmental model and/or whether someone is willing to pay money in an attempt to boost a sensor's priority.

5   Results and Discussion

The OmNetpp network simulator was used to test the auction method. Performance is dependent on the core MAC layer method that has been designed to control CSMA/CA network contention. The simulations use the default IEEE 802.15.4 MAC layer as a baseline for comparison, and tests how the proposed auction method behaves with continuous traffic congestion and temporal (event driven or converging) traffic. To test congestion/contention, the simulations vary the traffic loading (packets generated per second). The simulation results metrics (throughput, delivery (errors), and delay) provide evidence of the auction method's quality and performance improvements.

5.1   Performance Criteria

The quality and performance criteria involved in evaluation of the auction method are:

- Baseline vs. MAC layer method vs. auction method vs. real world scenario phases – These simulation phases allow for comparisons to validate the method within different system modules and conditions; and

- Node-based vs. network-based simulations – Permits analyzing traffic performance results between low-level individual node prioritization compared to continuous higher-level network loading (congestion);
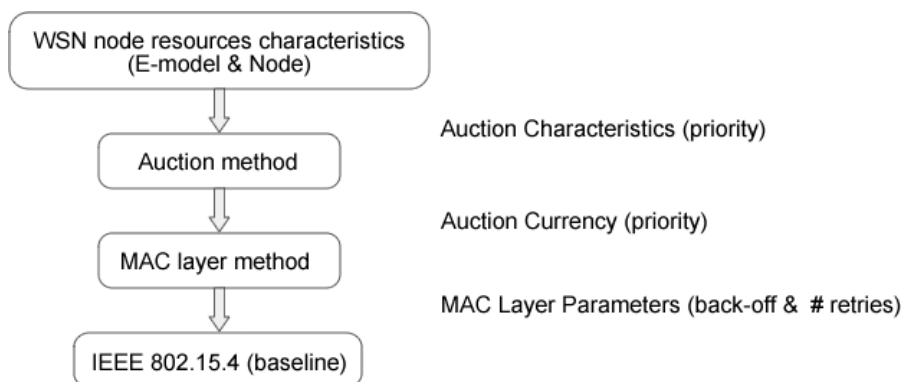

Figure 6: Simulation phases and hierarchy

50

Figure 6 illustrates how the comparison of different phases for evaluating the network results requires building a baseline default network, which all other method results are compared against. The MAC layer phase simulates the low-level (localized nodes) network contention of the IEEE 802.15.4. The auction method phase simulates the high-level (nodes resources) auction control which controls the MAC layer method. The real world phase simulates the flood scenario of each node in sequence will generate data with temporal high priority traffic. These simulation phases provide an incremental evaluation of the proposed method:

- Base-line phase – Build a network simulation with static default MAC layer parameters for all nodes to control network congestion;

- MAC Layer method phase – Configure static MAC layer parameters (macMinBE, macMaxBE and macMaxCSMABack-offs) for prioritizing all nodes to control network congestion;

- Auction method phase – Develop an auction method to use characteristics to control the MAC layer which in turn controls network congestion; and

- Real world scenario phase – Dynamically change auction characteristics to prioritise each node over time to monitor the delivery and delay performance results.

Testing the node-based (local) performance is more important compared to network-based (global) performance tests. This allows evaluation of how each node is performing individually and relative to other nodes in the network. The network-based simulations are beneficial for a final validation of the whole method performance. The different simulation levels of testing the performance of the WSN include:

- Node-based simulations – Evaluates how the auction method affects each node to prioritize its traffic while maintaining a static 80% network loading (packet rate) to provide network contention. The 80% network loading was chosen to provide enough congestion, but not affect the results through saturation and temporal congestion. Simulations results are averaged over multiple runs; and

- Network-based simulations – Evaluates how the auction method affects the network congestion due to changing network loading (packet rate). This mainly tests the method's performance near the start of network saturation (100% congested) and is used for indicating the method's effectiveness on continuous congestion. The network-based single run simulation is used to compare the method results to the baseline results, in order to provide an overall gauge of the method's performance. The network performance test is carried out only after the entire node simulations have been completed, and has the same setup of network parameters (different priorities between nodes) as the node simulations. The network performance simulation is used to validate the node simulations.

The throughput and delivery rate performance metrics gauge the method's reliability and bandwidth utilization, whereas, the delay metric helps to determine the fairness and Quality of Service (QoS) of the network traffic:

- Throughput (bps) – Number of bits (low level) received over a period of time;

- Delivery Rate (%) – Number of valid packets (high level) received compared to what is transmitted. This indirectly indicates the number of receiving errors (i.e., dropped packets); and

- Delay period (s) – Time between creating and receiving a valid packet.

5.2 Auction Simulation Goals

The goals and expectations for analyzing the auction method's simulation results are:

- To focus on improving network performance, i.e., not simulating all real world resource situations (location dependencies, buffering levels, and changes between auction periods);

- To successfully simulate a WSN and contention representative of real world network behavior and traffic;

- To improve method performance compared to the baseline default IEEE 802.15.4 CSMA/CA;

- Prioritized traffic will improve throughput and delay depending on node priority;

- Effects of increasing congestion (traffic load) are: increased contention which increases collisions, and increased throughput, dropped frames, and delay; and

Jarrod Trevathan
Lee Hamilton
Wayne Read

- The low priority node's traffic will not have a protracted period (delay metric) in transmitting waiting packets. The packet buffering required should not be too large for the embedded memory resources and there should be no significant delay in notifying the end user of an environmental event.

5.3   Assumptions

The reliability of the throughput and delivery performance is a priority over the delay metric (end-to-end latency) of network traffic. The method has trade-offs between certain performance metrics, which results in improved delivery and reduced delay metric results.

Network traffic prioritization can be tested based on the foundation of the MAC layer phase. The auction phase is a confirmation the auction method can control the critical MAC layer method. The majority of the simulations statically set the MAC layer parameters and the auction characteristics to thoroughly test the effects on the network.
The simulation number of test runs and run time assist in removing the randomness of node and network results. The randomness is a result of the varied packet transmits period and the baseline default MAC layer contention (IEEE 802.15.4) back-off random generated values.

The expected improvement in the bandwidth utilization is located around the start of network traffic saturation. At full saturation the bandwidth is fully utilized with no improvement. Another expected improvement is a reduction in the delay in receiving traffic from the high priority nodes in comparison to the low priority nodes.

5.4   Simulation Results

The simulation tests involved comparing the baseline vs. auction method. This included establishing a static network which dynamically adjusts the packet generation interval and back-off values. For each baseline and method simulation there are individual tests that allow performance comparisons between node-based vs. network-based results. Each of the node-based vs. network-based simulation test records the network performance metrics.

The simulation results and analysis are grouped by each network performance metric (i.e., throughput, delivery, and delay). The results grouping allows for an easy means of analyzing the influences the MAC and auction method properties are having on the different network metrics, in order not to favor one metric (delivery) over another metric (delay) results. The order and details of the simulation tests include:

- The baseline node-based simulation running for 10 separate test runs for 30 seconds;

- The baseline network-based simulation running varied network traffic loading (0.25 – 4 packets per second) test runs and each for 30 seconds;

- The MAC layer method is tested by running node-based simulations with the same default baseline simulation parameters and adjusting with a combination of macMinBE, macMaxBE and macMaxCSMABack-offs parameters over multiple different simulation tests. This is in order to improve all the performance metrics;

- The network-based performance simulations are run to validate the node-based results; and

- The auction method characteristics simulations confirm the control of the MAC layer method will provide the same results for prioritising the nodes.

In the MAC layer phase tests, each of the parameters (macMinBE, macMaxBE, macMaxCSMABack-offs) were adjusted within range, amount of scaling, and any prioritization between each node. Each adjustment required the node-based performance tests to be run over a number of times to provide averaged metric results.

The MAC method simulations involve prioritizing nodes by setting the MAC layer parameters (macMinBE, macMaxBE, and macMaxCSMABack-offs), with each parameter having the same importance as the other. As the macMinBE (back-off duration) decrease, the macMaxCSMABack-offs (number attempts) is increased proportionally. This is to ensure the total period of time to finally transmit is appropriate between different priorities, to reduce dropped frames and extensively long delays for low priorities. The parameters have very small ranges and the node's auction priority value is required to be scaled to fit within the range.

The simulations involved changing the MAC layer macMinBE (initial random exponent) parameter with different values and the prioritization between nodes that are within the range of macMinBE = [0 – 7]. The MAC layer macMaxBE (final random exponent) parameter is altered with different limits and prioritization between nodes that are within the range of macMaxBE = [5 – 8] and macMaxBE >= macMinBE. The MAC layer macMaxCSMABack-offs (number of transmit attempts) parameter is changed with different limits and prioritization between nodes that are within the range of macMaxCSMABack-offs = [2 – 8].

Jarrod Trevathan
Lee Hamilton
Wayne Read

Allocating Sensor Network Resources Using an Auction-Based Protocol

As expected, the node-based simulation results for both the auction and MAC layer methods are similar, as the auction characteristics set the node's priority, which controls the MAC layer parameters with the same values. The auction method simulations involve:

- Utilizing the MAC layer method to create parameters to prioritize the nodes;

- Simulation with each node set to one characteristic, with unique priority covering the entire range;

- Simulation with each node set to the same characteristic, with unique priority covering the entire range;

- Simulation with each node set to one characteristic with the same priorities and weightings; and

- Simulation with each node set to one characteristic with multiple priorities and weightings.
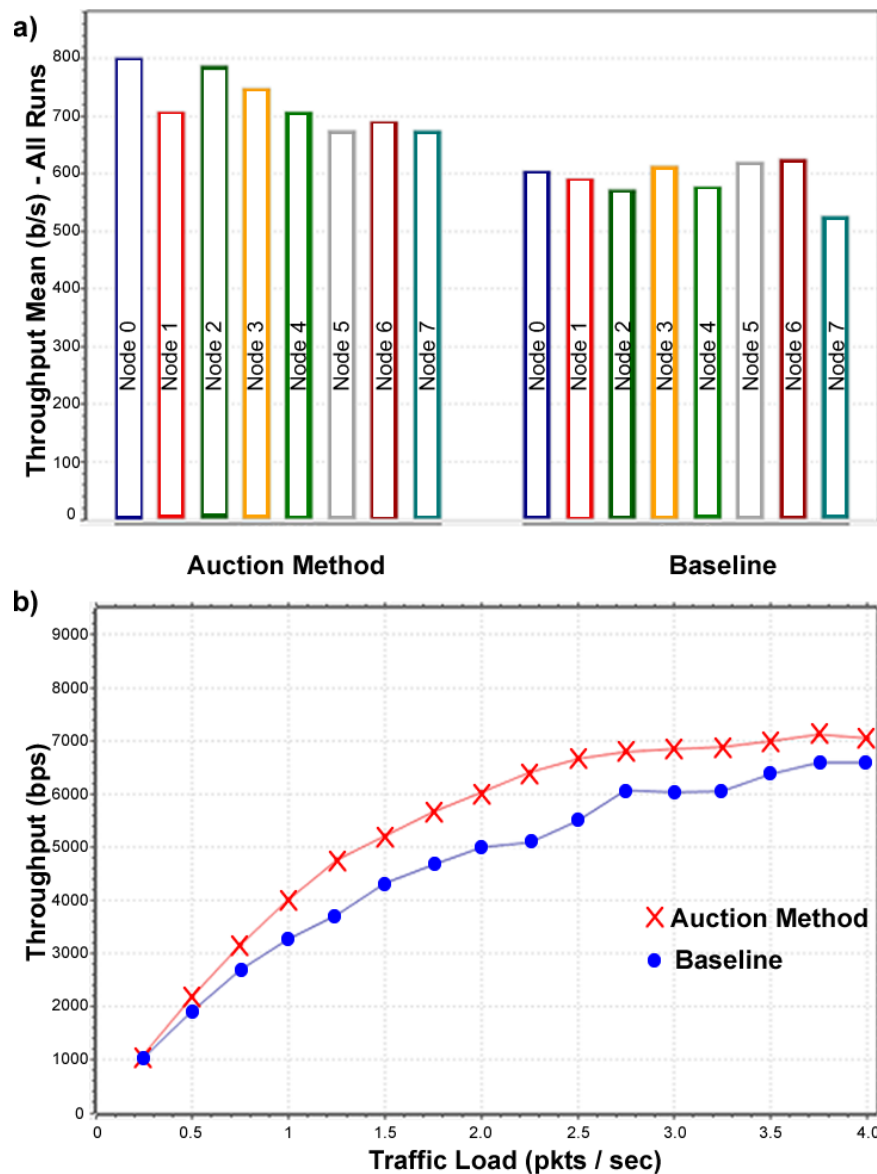

Figure 7: Node-based throughput mean results (a) and network throughput results (b)

5.4.1 Throughput

The throughput metric defines the receiving congestion rate over time for each node and the entire network. The resolution chosen for the throughput metric is Bits Per Second (bps) and not the packet per second. As this is not the

appropriate scale, the packets can have different lengths. The number of received (RX) bits is counted during the simulation. When the simulation finishes, the throughput is calculated as:

$$\text{Throughput (bps)} = \frac{\text{RX bits}}{\text{Simulation Time}} \tag{5}$$

Node-Based Simulation Results – The node's throughput result values for each simulation test run are averaged over all runs and the node's throughput mean is compared with the same nodes between the baseline and method. All test runs have randomness in generating packets and this variation can be viewed in the baseline results between nodes (Figure 7 (a)):

$$\text{Node Throughput Runs Mean (bps)} = \frac{1}{n} \sum_{i=1}^{n} \text{Throughput per Run} \tag{6}$$

The method results have all nodes with a higher throughput compared to the baseline by an average increase = method - baseline / baseline = (684.13 - 591.96) / 591.96 = **15.5%**. This results from increased bandwidth utilization of each node being prioritized and the increased number of retries. As expected, the high priority nodes (node 0 being the highest) will have higher throughput compared to the low priority nodes (node 7 being the lowest). This is due to the prioritization of all MAC layer parameters between nodes and a decrease in dropped frames.

Network-Based Simulation Results – The network-based traffic throughput simulation (Figure 7 (b)), tests from the low, through to the high rates of network traffic being generated. Ideally, the network throughput has a linear gradient increase up to saturation and plateaus with saturation. The baseline does not have this behavior, as the baseline result is more curved before reaching the start of the plateau (start of saturation). The method results demonstrate that throughput is increased around the start of the network loading, through to saturation. The results have slight fluctuations, as the network simulations have a single run per traffic load setting (packet/second). The method's network throughput is increased because of the following reasons: the nodes do not have similar back-off timeout values, an increased utilization of the bandwidth (each node takes turn in accessing), less dropped packets and the high priority nodes wait less (increased number of small back-offs). The best case throughput improvement of ((6000 – 5000) / 5000) * 100 = 20% is at the mid-point difference between the baseline and method (traffic load of 2.0 packets per second).

5.4.2 Delivery

The delivery metric defines the received (RX) vs. transmitted (TX) rate (useful for indicating error rate) for each node and the entire network. The delivery metric is critical compared to the delay metric, as it is more effective to have reliability (quality) over response time (performance). The number of received and transmitted bits is easily counted during the simulation, and when it finishes the delivery is calculated. The delivery metric is closely related to the throughput (received over time) metric and provides a similar increase in results:

$$\text{Delivery Ratio } \% = \frac{\text{RX bits}}{\text{TX bits}} * 100 \tag{7}$$

Node-Based Simulation Results – The node's delivery result values for each simulation test run are averaged over all runs (Figure 8 (a)). The node's method delivery ratio mean values over a number of runs are compared to the same nodes of the baseline.

$$\text{Node Delivery Runs Mean Ratio (\%)} = \frac{1}{n} \sum_{i=1}^{n} \text{Delivery per Run} \tag{8}$$

The delivery results show that all nodes have a higher delivery compared to the baseline, average increase = (method-baseline) / baseline = (69.9 - 60.4) / 60.4 = 15.7%. This is caused by the increased duration and retrying to transmit the frames before being dropped, resulting in a decreased number of dropped frames. The high priority nodes (node 0 being the highest) have a slightly higher delivery compared to the low priority nodes (node 7 being the lowest). This is a result of the prioritization of all MAC layer parameters between nodes.

Allocating Sensor Network Resources Using an Auction-Based Protocol

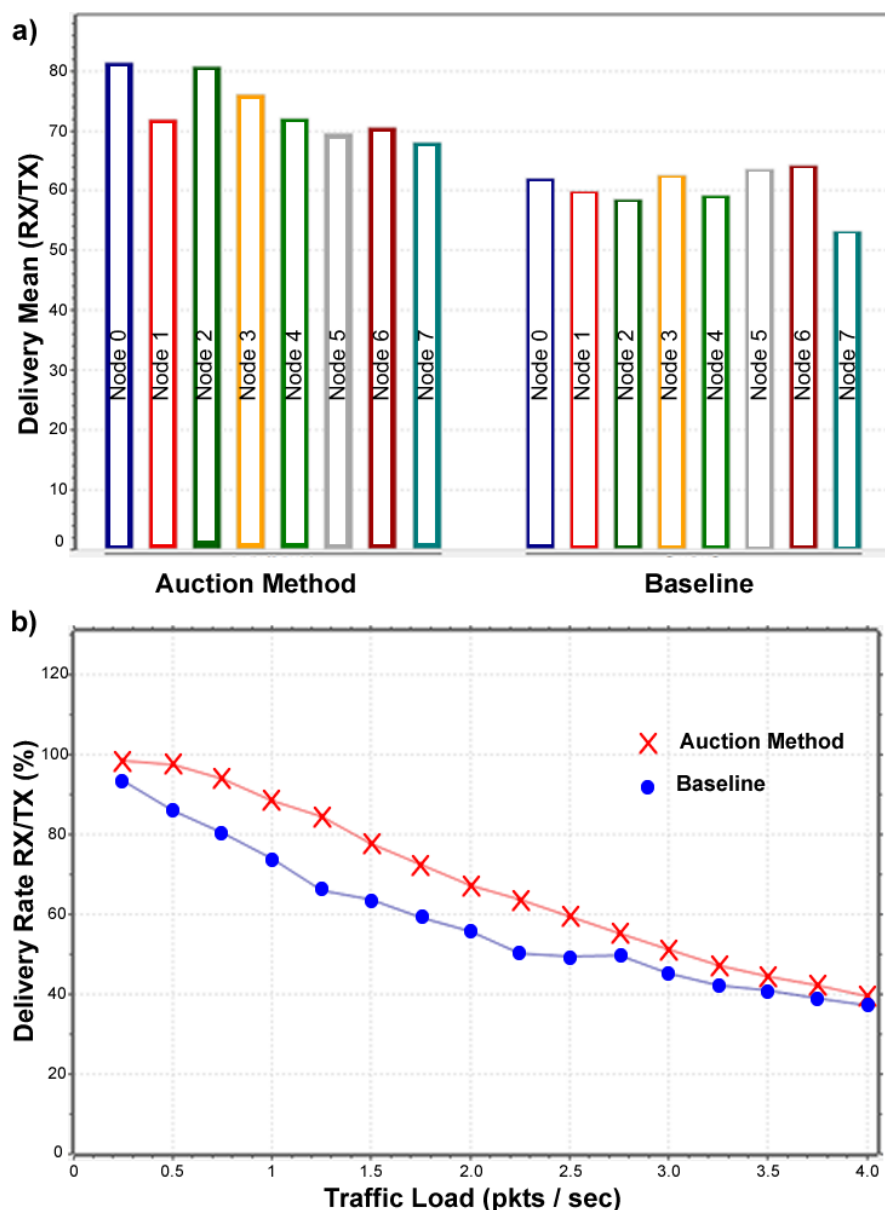Jarrod Trevathan
Lee Hamilton
Wayne Read

Figure 8: Node delivery mean results (a) and network delivery results (b)

Network-Based Simulation Results – The network-based traffic delivery simulation (Figure 8 (b)), tests low through to high rates of network traffic. Ideally, the network delivery has an increase in a linear gradient from between no network load and saturation. The method improvement reduces the number of dropped packets the more saturated the network becomes when compared to the baseline as there less ability to fit more traffic during saturation (> 2.0 packets per second). This explains why there is greater improvement compared to the baseline at the lower network loading. The simulation of network contention for controlling sequential access to a single channel bandwidth is apparent, by observing the delivery rate change as the network load is increased up to saturation. As a result the nodes will increase the randomly dropped packets. The dropped packets are the effect of the number of retries being exceeded between the competing nodes, when attempting to access the channel. The efficient use of bandwidth and the decrease in dropped frames, results in increased delivery performance. The best case delivery improvement of $((65.7 – 55.7) / 55.7) * 100 = 17.9\%$ is at the mid-point difference, between the baseline and method (network traffic load of 2.0 packets per second).

5.4.3 Delay

The delay metric defines the duration (end-to-end latency) between the timestamp for creating a packet to be transmitted, and the timestamp of the received valid packet for each node. The delay metric helps to identify the total period of back-off re-attempts to transmit the packet for each prioritized node. For each packet being generated that is sent (no hops) to the gateway node, the packet delay is averaged for each node over the whole simulation run. The delay metric represents the response time for detecting an event in the problem domain.

Analysis of the delay metric performance is important. If only the throughput and delivery results were considered when evaluating the method's performance, the delay (mainly affecting low priority nodes) could increase substantially to have noticeably long responses (possibly in the seconds) in detecting events in the WSN. The negative effect is node buffering overflow problems occurring while packets are waiting to be transmitted.

When comparing the delivery and delay, the overall increase in delivery (and throughput) of all nodes compared to the baseline will increase the overall delay of all nodes as the number of prioritized transmissions increase. The method causes a large delivery increase while only causing a slight delay increase. This is a result of the increased number of transmissions (throughput) and the default behavior of the MAC layer being less likely to successfully transmit with small back-off timeouts (high priority nodes). The cause of this problem is that even though the back-off is initially set low for high priority nodes, the transmit attempts fail (back-off) and the BE increments (back-off range increases) for each attempt. The frame is transmitted at the larger back-off which is similar to the baseline. None of the MAC layer parameters can be adjusted to resolve this problem. When comparing each node's delays, the high priority nodes will have a shorter delay (end-to-end latency) compared to low priority nodes, as the MAC layer parameters are prioritized to favor a shorter back-off timeout, to be capable of transmitting sooner.

$$\text{Packet Delay } s = \text{RX Timestamp} - \text{Create Frame Timestamp}$$

$$\text{Node Packet Delay Mean (s)} = \frac{1}{n} \sum_{i=1}^{n} \text{Delay per Packet} \tag{9}$$
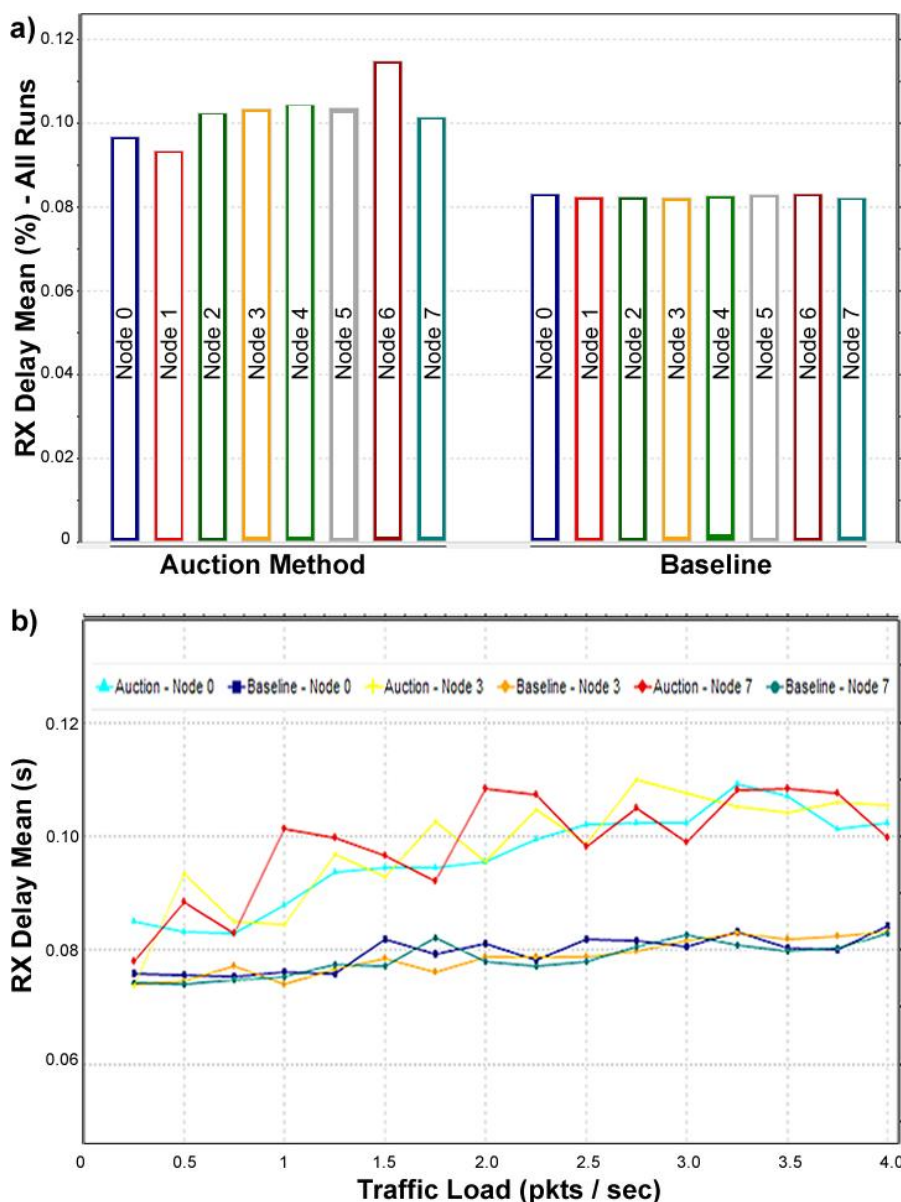

Figure 9: Node delay mean results (a) and network delay results (b)

Node-Based Simulation Results – The node's delay result values are averaged over all packets and simulation runs (Figure 9 (a)). The node's method delay result (over a number of runs) is compared to the same node in the baseline results. The randomness of generating packets and back-off timeouts in the delay results needs to be considered when comparing nodes. The node delay results have an incremental increase between the high priority nodes (node 0) through to the low priority nodes (node 7). This is due to the prioritized number of transmit retries for each node. The extra delay for low priority nodes is a result of the increased back-off range, the IEEE 802.15.4 transmits back-off slot period (20 symbols) and the increased total attempts period to transmit the same frame. The method's results have a relative delay difference between high to low priority nodes with a delay increase = node 7 – node 0 = (0.1133 - 0.0846) = 42ms. This is an acceptable amount of delay for the lowest priority node when compared to the transmit rate for a low priority node (approx. packet per 1 second). Each transmit attempt has no guarantee the actual back-off timeout is relative to the node's priority.

$$\text{Node Runs Delay Mean (s)} = \frac{1}{n} \sum_{i=1}^{n} \text{Node Packet Delay Mean per Run} \tag{10}$$

The baseline delay result has the similar delay between nodes, as all nodes have same MAC layer parameters. As expected, the auction and MAC layer methods have similar results, as the auction characteristics set the node's priority, which controls the MAC layer parameters with the same values.

Network-Based Simulation Results – The network-based traffic delay simulation (Figure 9 (b)) tests how three representative nodes are prioritized while adjusting the network traffic load being generated from low to high rates. The network delay has a slight increase in the results as the network traffic load is increased. This slight increase in delay as traffic loading increases is the same rate for both the method and baseline. This demonstrates the method has the same performance as the baseline, apart from the relative increases for low priority nodes. The reason for this steady increase is the average delay of packets will increase as more packets are waiting to be transmitted.

The average delay has an extra increase for the highest priority node of (0.0815 - 0.09) = 8.5ms (resulting from the extra throughput) and the lowest priority node of (0.0805 - 0.1275) = -47ms - -8.5ms (extra throughput) = 38.5ms, which is captured at the mid-point of the network traffic loading (packets per second = 2.0). The network-based results are similar to the node-based results and confirm that the delay results are valid. The average extra (added to the baseline average) delay is consistent throughout the network traffic loading range. This demonstrates the node delay will be consistent as congestion increases for each node. Unlike the throughput and delivery metrics, the network-based delay metric in Figure 10 shows the prioritization effects on each node to allow for analysis of the method's performance of each node throughout the network loading range.

5.4.4 Results Validation

The simulation times are validated by comparing different simulation times, without changing other settings with a single run of the baseline network. The test settings include: Bit Rate = 9600, Traffic Network Load = 1.7. The test monitors the network status (Table 1) to verify there is no large standard deviation in the results. The results have a small standard deviation for all the different simulation times. The ideal Sim Time is 30 seconds.

Table 1: Validating simulation times

Sim time	Network Load	Errors	Delivery	Channel Usage	Throughput
5	79.3	**39.7**	60.3	47.8	4592
10	80.5	**40.6**	59.4	47.8	4592
20	81.95	**38.8**	61.2	50.1	4816
30	82.25	**40.9**	59.1	48.6	4666
40	82.1	**39.7**	60.2	49.4	4746
50	82.0	**39.4**	60.6	49.7	4771
60	82.2	**39.2**	60.7	49.9	4797

The number of runs per test is validated by comparing the different number of runs in the node performance results (Table 2), and by averaging each metric result values over all runs. The test settings include: Sim Time = 15 and Network Load = 1.75 for node 0. This validates the mean of each metric that is fairly consistent with the number of runs increase and the standard deviation of each metric is decreasing, the more run data points are averaged. The ideal number of runs = 10 simulations.

Jarrod Trevathan
Lee Hamilton
Wayne Read

Table 2: Validating simulation number of runs

Runs	Throughput		Delivery		Delay	
Baseline – Node 0	Mean	Std Dev (+/-)	Mean	Std Dev (+/-)	Mean	Std Dev (+/-)
5	612	**60**	63	**5.91**	0.0793	**0.00184**
10	605	**57**	62	**6.03**	0.079	**0.000767**
15	596	**45**	60	**4.85**	0.079	**0.000518**
20	594	**39**	61	**4.2**	0.079	**0.000482**

The traffic network load per test is validated by comparing different traffic network loads to generate a baseline of the IEEE802.15.4 to determine packet rate. This is so the network is congested without the network being saturated (Table 3). The test settings include: 8 nodes, Bit Rate = 9600 and Sim Time = 10. The ideal traffic load is 1.75 (packet/sec) and this provides a ~80% network loading for all nodes transmitting to the gateway node.

Table 3: Validating simulation traffic loading

Traffic load	Network Load	Errors	Throughput
0.5	24	14	2016
1.0	46	12	3920
1.5	**68**	**37**	4144
2	**93**	**38**	5488
2.5	118	41	6720
3.0	140	50	6720

5.5   Real World Flood Scenario

This test simulates a flood progressing down a stream with the water height is changing over time. The simulation will confirm how changing each node's priority improves the performance relative to the neighboring nodes, and improves the reliability of the auction method's performance while controlling all the nodes. The simulation is focused on increasing the priority of three nodes, as the expected results can be repeated for other nodes. The contrast in the results between the nodes will be maximized and the simulation complexity will increase requiring the simulation to be run for a much longer period of time. All nodes are within range of each other and have a static traffic loading of around 80% to create network contention.
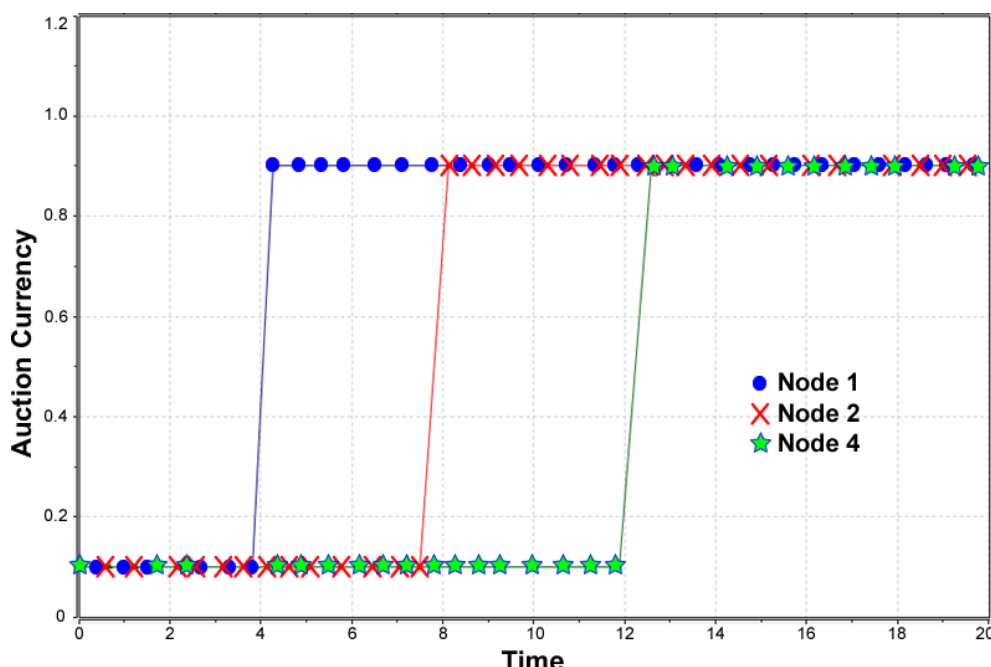

Figure 10: Real world flood auction currency simulation timeline

As time passes (each auction period), a selected node processes the auction round with the new increased value for the auction's sensor sample change characteristic. This changes the node's priority for a period of time. For each auction period the simulation records, the results since the last auction period are compared to the baseline results. The auction period is set to four seconds and the period will be of long enough duration to detect any changes in the

Jarrod Trevathan
Lee Hamilton
Wayne Read

results at the end of the period. The baseline results are the same simulation, except the node's priority remains unchanged for the auction period. The simulation is run several times to reduce randomness errors in the results. The simulation timeline has auction update points which increases a selected node's priority. Only three nodes have their auction currency (priorities) change (from 0.1 to 0.9), which start at 4 seconds and the preceding auction update points in the timeline (Figure 10). The higher priority nodes are 1, 2, 4, and are updated in the respective order.

The attempt to run the simulation to capture discrete results (vectors) throughout the simulation did not work as there was excessively random information for each message sent. The simulation and captured results were altered to gathering statistical summation results at the end of the simulation (end of the auction period). This allows for a single mean value (Figure 11 (a)) to determine the performance of the auction method at the end of each auction period (grouped nodes results per auction period) labelled *RealWorldAU-Cx* and x = [1-4]. The auction period results x = 1 occur when the simulation has run for 4 seconds until the other results for the incremental values of the x variable reach the end of the simulation. The auction period results (*RealWorldAU-Cx*) are the accumulation (incremental) of the entire simulation run time up to each auction update point, compared to one node with high priority for the entire simulation (Figure 11 (b)). This also affects the final results in Table 4 for the auction update results at the 8, 12, 16 seconds (auction points). The difference results in the table decrease over time.
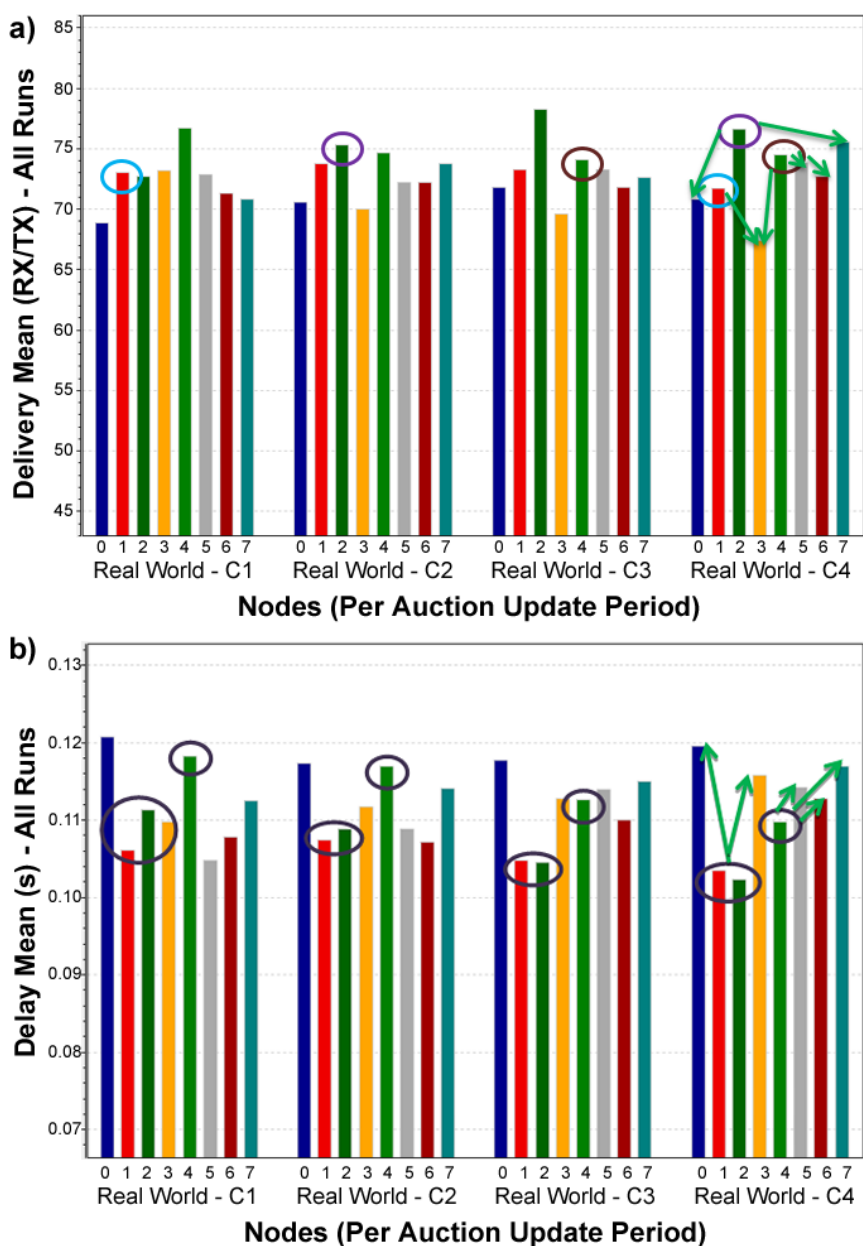

Figure 11: Real world flood simulation delivery results (a) and delay results (b)

The real world simulations results have randomness errors (i.e., no guarantee the actual back-off per transmit attempt is relative to the node's priority). The trend in the metric over time demonstrates the results are improving. As expected, the delivery metric results do not change a large amount. The simulation tables (Tables 4, 5 and 6) group

Jarrod Trevathan
Lee Hamilton
Wayne Read

the results into delivery and delay performance metrics. The result tables show the relationship between the auction update periods and nodes in comparison to the baseline results. The comparison against the baseline helps to evaluate the method results, as if the method had not done anything for each auction period (round).

Table 4: Real world flood simulation delivery results

Auction Update #		Node 1	Node 2	Node 4
8s	Baseline	70.7±9.73		
	Method	73.0±8.61		
	Difference	**2.3**		
12s	Baseline		74.1±7.81	
	Method		75.3±7.24	
	Difference		**1.2**	
16s	Baseline			72.9±8.10
	Method			74.1±9.30
	Difference			**1.2**

Table 5: Real world flood simulation delay results

Auction Update		Node 1	Node 2	Node 4
8s	Baseline	0.123±0.0192		
	Method	0.106±0.0118		
	Difference	**0.017**		
12s	Baseline		0.114±0.00744	
	Method		0.109±0.00769	
	Difference		**0.005**	
16s	Baseline			0.117±0.00762
	Method			0.113±0.00905
	Difference			**0.004**

Table 6: Real world flood simulation results with changing priority per node over time

Auction Update	Description	Delivery (%)	Delay (s)
8s	Node 1 changed priority	2.3	0.017
12s	Node 2 changed priority	1.2	0.005
16s	Node 4 changed priority	1.2	0.004
Mean		**+1.57**	**-0.009**

The simulation results are compared between the method prioritization changes (Table 6) to the baseline (no priority change) to confirm that each node's performance improves for each packet being sent with a higher priority. The high priority nodes results have an increase in delivery (average 1.57%) and a decrease in the delay (average 9 ms) performance metrics for each auction period. As expected, each high priority node's metric improves, the longer the node is set to a higher priority (for each auction update). As mentioned, these metrics for each auction period will decrease as a result of the amount of time the node is at high priority, compared to the accumulated simulation time. These values still provide a good indication the method improves the performance, even in the worst-case situation with the network not having all neighboring nodes with unique priorities.

The simulations results are also compared with the method prioritization changes between the nodes at each auction period (Figure 12 (a)). The high priority node has a slightly increased delivery metric result compared to the lower priority nodes for each auction period. The higher priority node has a noticeable decrease in the delay metric result compared to the slight change in the delivery. The greater the number of nodes becoming higher priority will have less impact on the improvements in the metrics results of the higher priority nodes, compared to the lower priority nodes. As a consequence, the duration (number of auction periods) while being set to a higher priority, will need to be considered especially if a substantial amount of other neighboring nodes are also set to a high priority. Each node's results improve at different rates relative to other nodes (they have different starting metric values when increasing their priority). The individual node's results improve over time.

The real world simulations have the auction period results (*RealWorldAU-Cx*) accumulate the entire simulation run time up to each auction update point, which dynamically change the nodes priority throughout the simulation. As a comparison, a separate simulation test (Figure 12 (b) has only one node with higher priority (node 2 – dark green bar)) compared to all other nodes for the entire simulation run (not incrementally changing like the Real World auction update test). The results demonstrate how the length (30 seconds) of being high has more of an impact on the delivery and delay metric results. The delivery increase = (node 2 – initial low priority nodes mean) / low priority nodes mean = (81 - 73) / 73 = 10.9% improvement compared to the average value of other low priority nodes. The

Jarrod Trevathan
Lee Hamilton
Wayne Read

delay decrease = (node 2 – low priority nodes mean) / low priority nodes mean = (0.094 – 0.115) / 0.115 = 18.3% improvement compared to the average value of other low priority nodes. The large percentage change value of the delay metric is mainly due to the simulation only having one high priority node.   All nodes metrics results are averaged over 10 runs and the delay metric is averaged for all packets for the entire simulation run.
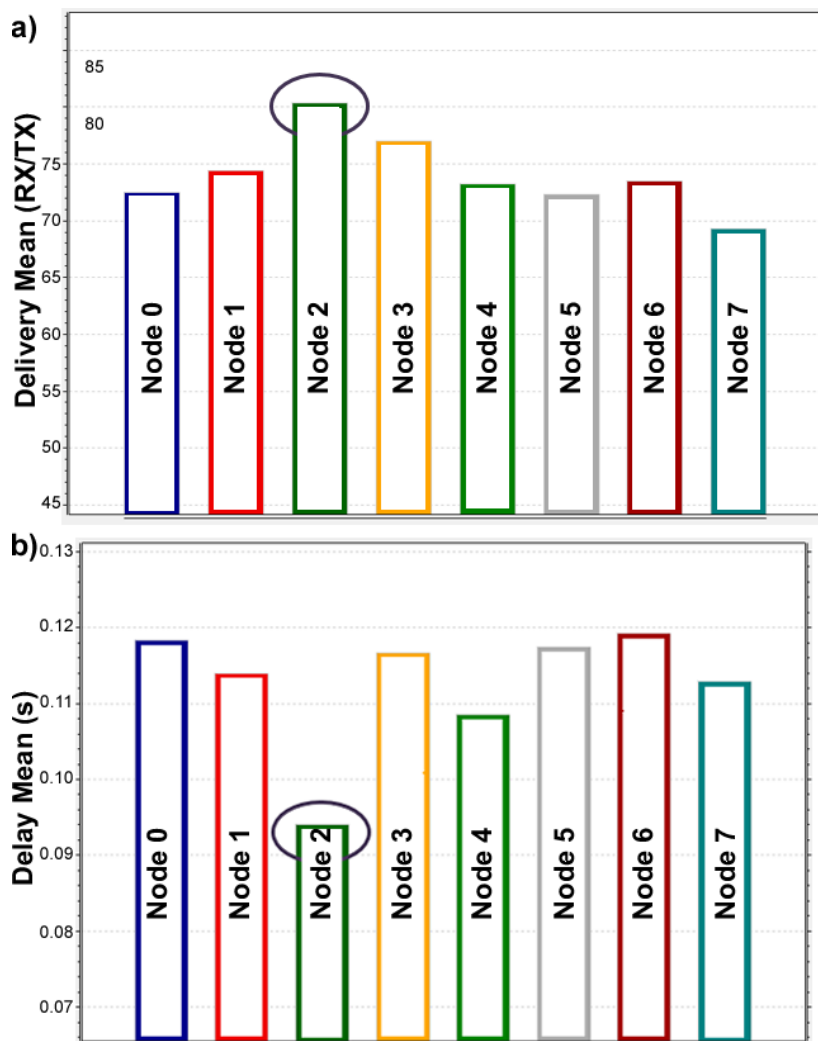

Figure 12: Delivery results for prioritizing only node 2 (a) and delay results for prioritizing only node 2 (b)

6   Conclusions

This paper presents a framework for smart sensors that participate in online auctions to acquire WSN resources. A sensor's *bid* value is based on its characteristics – cost, precision, importance of location, changes in regular readings, and amount of data collected. Priority is dynamically updated over time with regard to these characteristics, changes in the phenomenon under observation, whether someone is willing to pay for priority, and also with input from an environmental model. The environmental model allows the system to reason over collected data, and provides near real-time analysis for decision makers (who can also influence the priority based on economic decisions). We present an example scenario for monitoring a flood's progress down a river to illustrate how the system operates. A series of simulations were undertaken to examine how the system functions under different conditions.

The network control outcomes from the simulations are: 1) A configurable network simulation to test the proposed methods; 2) The ability to capture scalar and vector network results to calculate the performance metrics, in order to analyze and validate the auction method results; 3) Demonstration that the proposed auction method and MAC layer method produced similar improved network performance compared to the baseline 802.15.4. We found there is a trade-off between delivery and delay; and 4) The node simulation results improve the design and prove the MAC layer method to prioritize the Chile nodes during contention.

The performance metrics outcomes from the simulations indicated: 1) The auction method increases the delivery while decreasing the delay results for the high priority nodes. The opposite applies for low priority nodes, based on

Jarrod Trevathan
Lee Hamilton
Wayne Read

the node-based results; 2) Node-based (constant traffic load) simulations provide a means to evaluate each node's performance relative to others and verifies that one metric improvement is not adversely affecting other metrics; 3) Network-based (variable traffic load) simulations show that the method can handle varied traffic loading and node simulation results have improved the network performance; 4) The node-based averaged throughput increased by 15.5% between the baseline and method results; 5) The network-based maximum throughput increase is 20.0% between the baseline and method around 2.0 packets per second network loading; 6) The node-based average delay compared between high priority nodes and low priority nodes is 42ms. This increase for the low priority node is acceptable in the problem domain; 7) The network-based delay results are similar to the node-based results between the baseline and method. The delay results provide a consistent value as traffic loading increases, indicating the auction method will behave the same as congestion increases; 8) The network throughput performance criteria has consistent improvements with the throughput metric and the delivery rate metric; and 9) We validated the method does not have excessive delays.

The simulation phases outcomes show: 1) The MAC layer method vs. baseline comparison indicates that the low level MAC layer control improves the local network performance, while independently not having to consider the auction method (i.e. WSN resources management); 2) The auction method vs. baseline comparison suggests the high level auction method improves the global network performance by generating similar improved results as the MAC layer method; 3) The real world simulations tests proved to be difficult to bias the MAC method to produce noticeable results for the higher priority nodes in the shorter durations; and 4) The real world vs. baseline results comparison indicates the individual node's method performance is improved over time (auction periods).

The real world simulations suggest that the higher priority nodes have improved method performance when compared to lower priority nodes. The same amount of data sent with increased delivery (less dropped) and decreased delay (shorter back-off). The simulation induces the worst case situation with most nodes, with the same priority.
Future work involves investigating different auction types (such as CDAs) and structures (e.g., peer-based) and their implications for sensor networks. Other work includes integrating more representative environmental models that operate in conjunction with semantic annotation and inference engines [12], [20] for event tracking throughout a geographical area. Additionally, more truly dynamic systems need to be examined where the sensor network configuration can change during the course of the deployment. Furthermore, starvation in a network is a serious issue that needs to be addressed. Finally, the proposed scheme does not address what to do in a tie-situation where two or more sensors have the same $ value.

Websites List

Site 1: Gumstix, Inc
http://www.gumstix.com

Site 2: Arduino
http://www.arduino.cc

References

[1] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, Tracing a moving object with a binary sensor network, in Proceedings of the 1st International Conference on Embedded Sensor Systems, Los Angeles, CA, 2003, pp. 150-161.
[2] C. Beard, Preemptive and delay-based mechanisms to provide preference to emergency traffic, Journal of Computer Networks, vol. 47, no. 6, pp. 801-824, 2005.
[3] T. Berners-Lee, J. Hendler and O. Lassila, The semantic web, Scientific American, vol. 284, no. 5, pp. 34-43, 2001.
[4] Chen, L., B.K. Szymanski and J.W. Branch, Auction-based congestion management for target tracking in wireless sensor networks, in Proceedings IEEE International Conference on Pervasive Computing and Communications, Galveston, TX, 2009, pp. 1-10.
[5] L. Chen, Z. Wang, B. Szymanski, J.W. Branch, D. Verma, R. Damarla, and J. Ibbotson, Dynamic service execution in sensor networks, The Computer Journal, vol. 53, no. 5, pp. 513-527, 2010.
[6] S. Chen and N. Yang, Congestion avoidance based on lightweight buffer management in sensor networks, IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 9, pp. 934-946, 2006.
[7] R. Comment and G.A. Jarrell, The relative signalling power of Dutch auction and fixed price tender offers and open market share repurchases, Journal of Finance, vol. 46, no. 4, pp. 1243-1271, 1991.
[8] R. Das, J. Hanson, J. Kephart, and G. Tesauro, Agent-human interactions in the continuous double auction, in Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI), Seattle, 2001, pp. 4-10.
[9] N. Edalat, C. K. Tham and W. Xiao, An auction-based strategy for distributed task allocation in wireless sensor networks, Computer Communications, vol. 35, no. 8, pp. 916-928, 2012.
[10] F. Gul and E. Stachetti, English and double auctions with differentiated commodities, Journal of Economic Theory, vol. 92, no. 1, pp. 66-95, 2000.

Jarrod Trevathan
Lee Hamilton
Wayne Read

[11] G.S. Hamilton, F. Fielding, A. Chiffings, B.T. Hart, R.W. Johnstone, and K.R. Mengersen, Investigating the use of a bayseian network to model the risk of lyngbya majuscula bloom initiation in deception bay, Queensland, Australia, Human and Ecological Risk Management, vol. 13, no. 6, pp. 1271-1287, 2007.

[12] V. Huang and M. K. Javed, Semantic sensor information description and processing, in Proceedings of the 2nd International Conference on Sensor Technologies and Applications, Cap Esterel, 2008, pp. 456-461.

[13] H.T. Kung and D. Vlah, Efficient location tracking using sensor networks, in Proceedings of the 2003 IEEE Wireless Communications and Networking Conference, Cambridge, MA, 2003, pp. 1-8.

[14] H.B. Lim, M.Ch. Foo and Y. Zeng, Adaptive distributed resource allocation in wireless sensor networks, in Proceedings of the 2nd International Conference on Mobile Ad-hoc and Sensor Networks, Hong Kong, China, 2006, pp. 770-781.

[15] G. Mainland, D.C. Parkes and M. Welsh, Decentralized, adaptive resource allocation for sensor networks, in Proceedings of the 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation, Berkeley, CA, 2005, pp. 315-328.

[16] J. Ostwald, V. Lesser and S. Abdallah, Combinatorial auctions for resource allocation in a distributed sensor network, in Proceedings 26th IEEE International Real-Time Systems Symposium, Miami, FL, 2005, pp. 274.

[17] D. Schrage, C. Farnham and P.G. Gonsalves, A market-based optimization approach to sensor and resource management, in Proceedings in Defense and Security Symposium, International Society for Optics and Photonics, Orlando, FL, 2006, pp. 62290I-62290I.

[18] S.Y. Shah and B.K. Szymanski, Transient traffic congestion control with traveling auctions, in Proceedings IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Lugano, 2012, pp. 14-19.

[19] S.Y. Shah and B.K. Szymanski, Price based routing for event driven prioritized traffic in wireless sensor networks, in Proceedings IEEE Network Science Workshop, West Point, NY, 2013, pp. 1-8.

[20] A. Sheth, C. Henson and S.S. Sahoo, Semantic sensor web, IEEE Internet Computing, vol. 12, no. 4, pp. 78-83, 2008.

[21] K. Shi, Y. Shu, O. Yang, and J. Luo, Receiver-assisted congestion control to achieve high throughput in lossy wireless networks, IEEE Transactions on Nuclear Science, vol. 57, no. 2, pp. 491-496, 2010.

[22] N. Tabassum and G. Lu, Data-Driven MAC for efficient transmission of spatially correlated data in WSNs, Communications: Wireless in Developing Countries and Networks of the Future IFIP Advances in Information and Communication Technology, vol. 327, pp. 146-157, 2010.

[23] J. Trevathan, H. Ghodosi and W. Read, An anonymous and secure continuous double auction scheme, in Proceedings of the 39th Hawaii International Conference on System Sciences, Kauia, HI, 2006, pp. 1-10.

[24] J. Trevathan, R. Johnstone, T. Chiffings, I. Atkinson, N. Bergmann, W. Read, S. Theiss, and T. Stevens, SEMAT – The next generation of inexpensive marine environmental measurement and monitoring systems, Sensors, vol. 12, no. 7, pp. 9711-9748, 2012.

[25] W. Vickrey, Counter speculation, auctions, and competitive sealed tenders, Journal of Finance, vol. 16, no. 1, pp. 8-37, 1961.

[26] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, Upstream congestion control in wireless sensor networks through cross-layer optimization, IEEE Journal on Selected Areas in Communications, vol. 25, no. 4, pp. 786-795, 2007.

[27] Z. Zhang, W. K. Chan, T. H. Tse, H. Lu, and L. Mei, Resource prioritization of code optimization techniques for program synthesis of wireless sensor network applications, Journal of Systems and Software, vol. 82, no. 9, pp. 1376-1387, 2009.

Allocating Sensor Network Resources Using an Auction-Based Protocol

Jarrod Trevathan
Lee Hamilton
Wayne Read