Pervasive and Mobile Computing, vol. 8(3), June 2012, pp.346-357

Market Mechanisms for Resource Allocation in Pervasive Sensor Applications

Sahin Cem Geyik[a], S. Yousaf Shah[a], Boleslaw K. Szymanski[a], Sanmay Das[a], Petros Zerfos[b]

[a]*Department of Computer Science and Network Science and Technology Center*
Rensselaer Polytechnic Institute (RPI), Troy, NY, USA
Email: {geyiks,shahs9,szymansk,sanmay}@cs.rpi.edu
[b]*IBM T.J. Watson Research Center, Hawthorne, NY, USA*
Email: pzerfos@us.ibm.com

Abstract

This paper describes the use of market mechanisms for resource allocation in pervasive sensor applications to maximize their Value of Information (VoI), which combines the objectively measured Quality of Information (QoI) with the subjective value assigned to it by the users. The unique challenge of pervasive sensor applications that we address is the need for adjusting resource allocation in response to the changing application requirements and evolving sensor network conditions. We use two market mechanisms: *auctions* at individual sensor nodes to optimize routing, and *switch options* to optimize dynamic selection of sensor network services as well as switching between modes of operation in pervasive security applications. We also present scenarios of transient congestion management and home security system to motivate the proposed techniques.

Keywords: Pervasive Applications, Wireless Sensor Networks (WSNs), Value of Information (VoI), Quality of Information (QoI), Auctions, Switch Options.

1. Introduction

Deciding how to allocate resources to maximize Value of Information (VoI) for network users is a challenging problem in pervasive sensor applications. First, the value of information to each user may be subjective and user specific. Second, both the application needs and network conditions can evolve with time. A reasonable way of allocating resources in such domains is to deploy market mechanisms that take decisions based on the explicit valuation that users assign to information and apply them to the current network conditions. However, designing mechanisms that allow both the appropriate amount of information disclosure as well as efficient outcomes is a challenge in itself. As proof of concept, we present two market mechanisms that can be deployed in pervasive sensor applications to address aforementioned challenges.

The Value of Information is most easily thought of as a price that a user of network information is willing to pay for it. Therefore, it is reasonable to think about it as the *change in expected utility* of a decision-maker who receives the information. We distinguish between quality of information (QoI) [2], which can be represented by a vector of objective measurements, and VoI, which depends both on objective QoI as well as on the subjective assessment of the importance that an end-user assigns to certain information. Hence, QoI can be measured entirely within the system that produces information while VoI is dependent on both QoI and exogenous information usually provided by the information user.

This paper focuses on maximizing VoI in pervasive sensor applications. We propose to use two market mechanism methods[1]. Our first method deals with the low-level sensing and routing operations of a single sensor node. The goal at this level of WSN (Wireless Sensor Network) architecture is to collect the set of the most meaningful measurements or to transmit the most valuable packets first when congestion arises. This is particularly difficult in sensor networks because the congestion is often intermittent, associated with events that move through the network (like an edge of the forest fire, or an object tracked by the network). Therefore, traditional congestion control mechanisms that rely on feedback from the destination to the source do not work well. An important aspect of QoI in sensor networks is the time delay with which information is available to end-users. Thus, any congestion or packet collisions will lower QoI of the information carried by affected packets. We assume QoI is not affected by network utilization, making resource allocation approaches based on efficient network utilization not applicable [26, 6]. We demonstrate how the use of *auctions* on the very nodes where

[1]The initial description of our approaches as well as the initial experiments were published in [25] and [8].

temporary congestion occurs can resolve congestion in a way that minimizes the loss of VoI.

Our second method dynamically optimizes composition of services in pervasive sensor network applications. *Switch options* are often utilized in economics to decide the selection of mode of production for factories, or switching between investment options to maximize profit/minimize loss. When *Service Oriented Architecture* (SoA) is used in sensor networks, the more complex functions are composed of basic services that are often hosted by multiple nodes. Their execution commands significant resources (most importantly energy, but also bandwidth, computing power, and sensing modes). We use switch options to dynamically adjust the efficient composition of basic services to changing network conditions. We also demonstrate the use of this technique in a domestic pervasive security application in which false security-breach alarms can be prevented by cleverly switching between secure/insecure modes according to expected gains/losses of each mode.

The problem of decentralized resource allocation has been studied widely in communication networks, yet not in the context of both VoI and dynamically changing application needs and network conditions on which we focus here. Mainland *et al.* propose a decentralized reinforcement-learning based scheme for efficient vehicle tracking in a WSN under energy constraints [17]. Congestion based pricing mechanisms have also been featured prominently in the literature [16, 23], generalizing from the traditional domain of tollways and airports to communication networks. In such systems, the goal is to tackle the effects caused by increased traffic. Flat rate pricing, prevalent today, is not an optimal strategy for service providers [23]. Fixed pricing schemes can lead to overuse of bandwidth by exploitation of TCP at the user end, so using congestion pricing or differentiated QoS would help to regulate bandwidth use. Various mechanisms to avoid Internet congestion have been suggested (although none have thus far gained much real-world traction) [5, 13]. Most of these mechanisms rely on congestion feedback from the destination to the source, an approach that is not feasible for the intermittent congestion caused by event-driven network flows in WSNs. In the domain of WSNs, the works we would like to list are *RAP* [15], which prioritizes packets to reach a certain velocity to meet their deadlines, and *SPEED* [10] which aims to ensure a desired delivery speed along the sensor network. These papers aim to meet real-time packet urgency constraints as well as to deal with congestion in WSNs.

The rest of the paper is organized as follows. In the next section we discuss how auctions can be utilized to maximize VoI gain in sensor network congestion control. In Section 3, we present our switching options methodology applied to sensor network service selection as well as pervasive security applications. We conclude the paper in Section 4. Please note that apart from the related work given in the current section, the relevant previous work is discussed in the corresponding sections.

2. An Auction Mechanism for Distributed Congestion Control

As discussed above, resource allocation considered here is driven by VoI that combines an objective function that measures QoI and a subjective component that assesses how valuable information with the given QoI is to the end-user. We consider the low level management of resources in WSNs, as exemplified by bandwidth allocation in congestion scenarios. Following Chen *et al.* [4], we use public safety and emergency response needs of a VIP visiting a large city as a motivating example. The streets are equipped with a sophisticated WSN composed of acoustic sensors, closed circuit cameras, chemical fume sensors, and so on. Various state agents are present in the area to ensure the safety of the visitor and the public, using the deployed WSNs to monitor events.

Suppose a vehicle with a suspicious driving pattern is being tracked by the state agency, while a gathered mob is continuously monitored by local police. Both targets are getting close to the visitor's car as they enter a nearby intersection. Now, data packets are continuously sent to three different sinks (agencies), all with high priority, monitoring three different targets (visitor's car, suspicious mob and threatening vehicle). A noticeable problem occurs due to the fact that all three targets are physically close, causing congestion at nearby nodes that transmit packets to sinks. This congestion increases network delays and may even cause packet loss, decreasing QoI of the traces of the targets that are of high value at that point in time. This problem is even more complex when there are more targets with coordinated mobility patterns involved. Following Chen *et al.* [4], we formally define an auction mechanism to solve this problem.

While the value of information could be a very general function of QoI and user-specific importance of the information, for simplicity, we consider a case where the utility for an application i is a linear function of the QoI received about this application, $U_i = \alpha_i q(i, d)$, where α_i is the application specific multiplier and $q(i, d)$ is the QoI for the specific application i with the data transmission delay, d, experienced by this application[2]. We make one further simplification in what follows. We assume that each node maintains only one packet for each application because when two distinct packets of the same application are received at the destination, the one with the more recent target data brings VoI to the same value, regardless of whether the other one was received or not. Hence, each node's maximum queue length is limited by the number of active applications, a requirement easily satisfied by modern sensor nodes. Once congestion arises, the node needs to decide in what order the waiting packets will be transmitted, exposing them to different delays at the node. The standard congestion control techniques involve destination-to-source feedback therefore they are not applicable here because of the intermittent nature of congestion that keeps moving from one node to another in target tracking applications of WSNs.

To quantify the loss of information resulting from packet delays caused by congestion, we need to assess VoI of a piece of information. There are two related ways of thinking about this: first, how much the new information reduces uncertainty about the position of the object being tracked. A useful mathematical formulation in this case may be the reduction in entropy of the user's belief about the location and direction of the tracked object. If the user's belief is represented by a well-behaved distribution, like a Gaussian, this could be further simplified by considering perhaps only the variance of the distribution.

A related way of thinking about this issue is to quantify the "surprise" element of an observation as being key to the value of the information contained in the observation. For example, if a suspicious vehicle was traveling slowly in a particular direction, but suddenly made a U-turn and accelerated sharply, that information may be more relevant than if it just continued on its previous path. How can this be objectively quantified? Perhaps the difference in implied probability distributions of the location of the object could be important. A measure like KL-divergence of the two distributions (the post-observation distribution and pre-observation distribution) may be helpful in quantifying VoI in cases like this.

When congestion occurs in the network, the delay of packet delivery in-

[2]Although many more parameters may define the value of $q(i, d)$, we explicitly refer to d, as this is the parameter directly impacted by congestion, while other parameters, such as precision of target's position or precision of the time of measurement are not.

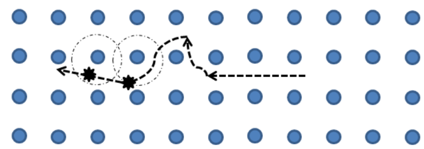


Figure 1: An object in pursuit of another while being tracked by sensing nodes (for simplicity sensing range of only two nodes is shown). The dashed line shows the path followed by objects.

creases which causes increases in the VoI loss. There are scenarios in which the congestion occurred on one node, disappears and reappears on a nearby node, thus giving an impression of mobile congestion. Such situations often arise in scenarios where the data streams are intensive but short. Consider a car chased by police; as the car moves along its path in the sensor network, the sensor nodes sensing the car generate bursts of data. As the followed and following cars move forward, they enter into the sensing ranges of new sensor nodes and exit the ranges of sensor nodes behind it. Consequently, the bursts of packets are generated by different sensor nodes as the cars continue the chase. These bursts generate congestion on the routing nodes with the congestion moving from one routing node to another as the bursts change their origins and trajectories. These bursts are short but produce intensive traffic. In such a situation the congestion moves from node to node. Previously developed solutions, such as [4], use, what we call *Static Auction* at each congested node that starts from scratch when the congestion arises there. Thus, they do not take into account the VoI losses accumulated by an application at previously congested nodes in the network. In particular, they initially assume that no packets were delivered yet to the destination. Hence, all packets will pass the congested node in a round robin fashion sequentially in the ordered of their priorities. Thus, packets of high priority will wait for such initial round robin round to finish before they can fully assert their priority. To avoid this problem, we introduce the *Traveling Auctions* in which the auctioneer node shares its local auction information with its neighbors. When congestion arises in any of the neighboring nodes, they can use the shared information to make more efficient decisions. The sharing is done via each node overhearing its neighbors. Accordingly, the packet header is extended to include auction specific information, including application id and the time of measurement of the packet.

In general, the Traveling Auctions approach applies to various mobile application scenarios in which congestion moves from node to node, such as: (i) unmanned vehicles performing joint operation and communicating over wireless network, (ii) rescuing workers using an ad-hoc wireless network in a disaster stricken area exchanging messages of various priorities with aid agencies, hospitals and general public, and (iii) tracking groups of mobile objects. All the aforementioned scenarios involve mobile objects with prioritized data traffic in which an efficient congestion resolution mechanism is needed to effectively use the WSN bandwidth.

In this paper, we demonstrate benefits of Traveling Auctions in the scenario of a tracking application. In this scenario, the goal of an auctioneer is to minimize the total VoI loss for all the applications. Let p_i be the priority of the application a_i. This priority is a function of the subjective measure of importance of the application. We assume TDMA as MAC layer protocol in our scenario under which each node receives a transmission slot in which it is able to transmit a new packet every t_c time units. Let tm_i denote the time at which the packet of application i competing for the current transmission slot was generated, which is also the time of measurement of the target object that this packet carries. Let tmp_i be the packet origination time at the source for the latest packet of application i known to be sent to the sink. As mentioned earlier, we also assume that the VoI loss caused by losing the next auction is a linear function of the delay[3].

Whenever the congestion arises on the node, the node itself computes the bids of all packets competing for the transmission slot and then allocates it to the highest bidder. The bid b_i for the current transmission slot for the application a_i representing this loss is computed as:

$$b_i = (tm_i - tmp_i)\ p_i\ t_c \tag{1}$$

From the above definitions, it is clear that when a new packet of the application for which we already have a packet in the queue waiting to be delivered arrives, only the packet with the latest origination time needs to be stored and the other one can be discarded. Figure 2 illustrates the VoI losses that are avoided when the latest available packet is delivered. In the current auction, the VoI loss for application 1 is larger than for application 2, so the

[3]If the VoI loss is a non-linear function of the packet delay, the bid computation will be just the difference between the predicted VoI loss after winning the auction and the loss after losing the auction.

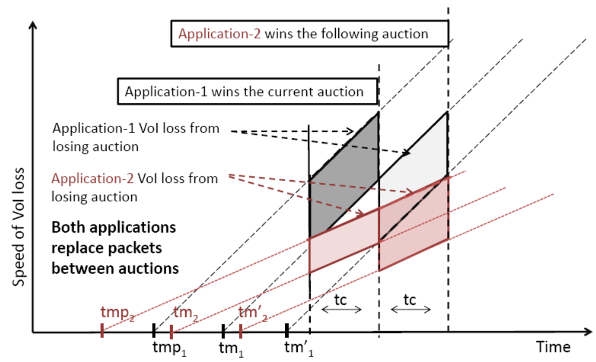


Figure 2: Replacing a packet with a fresher one; tmp_1 and tmp_2 are the origination times of the latest packets that passed through the node (or zeros where none passed yet), tm_1, tm_2 are the origination times of the packets competing for current transmission slot, while tm'_1, tm'_2 are those times for packets competing for the following transmission slot, for applications 1 and 2, respectively. Note that bids correspond to the sizes of shaded areas in the diagram.

packet of application 1 is selected. As a result, in the following auction, the time of the latest packet of application 1 raises to tm_1, so the VoI loss in that auction is larger for application 2 than for application 1, and therefore the packet of application 2 will be selected for the transmission slot.

2.1. Evaluations

To evaluate and compare Traveling and Static Auctions, we simulated the object tracking scenario using the Network Simulator *ns-2*. We used TDMA as the MAC layer protocol with the transmission slot being large enough to transmit one packet. Each node receives one transmission slot every $t_c = 0.1$ sec. We simulated a sensor network of 44 nodes deployed as a flat grid. The number of mobile objects (represented as special ns-2 nodes) varied between two and four. Each object was tracked by a separate application of the object tracking algorithm. The sensor nodes were stationary, while the tracked objects were moving according to mobility patterns mimicking real life objects movements. We used *ns-2* native DSDV routing protocol to route packets to sink. The simulation was run 5 times (each for 200 seconds) and at the end of simulation we calculated the total VoI loss at the sink for all applications.

We have developed a traffic generator attached to all sensing nodes, so each such node generates 10 packets per second for each object that is in its sensing range. The generated UDP packets are customized with extended header containing scenario specific information, such as the total number of applications, the associated application id, the time of the packet creation, and the application priority. Whenever the object enters the sensing range of a node, the node starts generating object measurement packets for the application associated with this object. The simulator is configured with *TwoRayGround* medium model [20]. Table 1 shows VoI loss (with the

Priorities	Trav.Auctions	Static Auction	Network Losses	Gain
9,36	1552 ± 8.4	1790 ± 10.3	1033 ± 0	45.8%
4,16,36	2991 ± 7.5	3724 ± 10.5	1359 ± 0	44.9%
2,4,9,16	2321 ± 5.3	2686 ± 3.7	1530 ± 0	46.1%

Table 1: VoI Losses

95% confidence intervals) for applications under: (*i*) Traveling Auctions, (*ii*) Static Auctions, and (*iii*) no congestion. The losses without congestion, referred to here as *network losses*, are accumulated by every application because of non-congested traffic delays and non-continuous measurements of the object movements. Hence, there are three types of losses that make up the total VoI loss:

1. Non-continuous measurements loss is incurred by lack of new data about the object between two subsequent measurements; the higher the frequency of measurement, the lower the VoI loss.
2. Non-congested traffic delay loss is caused by the delay the packet incurred traversing the non-congested path from the source to destination; the larger the hop distance from the source to destination, the larger the loss incurred by the application.
3. Congestion delay loss is incurred by the waiting that some of the packets experience at the routing nodes when the incoming transmission slots are taken by the packets of other applications. We can reduce this loss and thus the total VoI loss by efficiently managing the order in which packets of different applications are assigned to incoming transmission slots.

The network losses for two applications with priorities (9,36) are shown in Table 1. In the simulation, the sink received 2000 packets. The position of objects were measured every 0.005 sec, so loss caused by the non-continuous measurements was $0.005/2 \times 2000 \times (36 + 9) = 225$. Hence, the VoI loss due to the non-congested traffic delay is $1033 - 225 = 808$ and it bounds the delay that was at least $(1033 - 225)/2000/(36 + 9) = 0.009$ sec and at most $(1033-225)/2000/(36+9) \times 2 = 0.018$ sec. The delay measured in the simulation was 0.012 sec, so it was consistent with the bounds. Doing the same check for three applications with priorities (4,16,36), we find that the minimum cost caused by non-continuous measurements is $0.005/2 \times 2000 \times (4+16+36) = 280$ and the corresponding bounds on network delay are 0.0096 sec and 0.019 sec which again is consistent with the measured delay of 0.012 sec. The same consistency was observed for the case of four applications.

Since the auction impacts only the congestion caused losses of VoI, we measured how much higher was the Static Auction VoI loss relative to Traveling Auction VoI loss using the following formula:

$$Gain = 100 \times \frac{Loss\_Static\_Auctions - Loss\_Traveling\_Auctions}{Loss\_Traveling\_Auctions - Loss\_Network} \tag{2}$$

As shown in Table 1, Static Auctions incur around 45% higher losses compared to Traveling Auctions.

To confirm the scale of performance gains, we conducted tests for many different priority sets for two, three and four mobile objects under the same settings. Table 2 shows average of performance improvements (with the 95% confidence intervals) for different numbers of mobile objects tested. For brevity, the actual values of losses are not presented here, instead we computed the improvements for each individual case and then took their average. As discussed earlier, the Traveling Auctions are applicable to other scenar-

Priorities	Gain
(9,36),(2,8),(3,6),(4,16),(5,10)	43.5% ± 1.5%
(5,10,15),(2,4,9),(3,6,9),(4,16,36),(4,8,12)	43.8% ± 0.7%
(5,10,15,20),(2,4,9,16),(3,6,9,12),(4,16,36,64),(4,8,12,16)	44.0% ± 1.4%

Table 2: Average Gains

ios in mobile applications. We have described the use of Traveling Auctions with one specific utility function, but the auctions can be defined with any non-decreasing utility function of delay. As we have shown, Traveling Auctions perform significantly better than Static Auctions under highly bursty, short-term traffic.

3. Switch Options for Pervasive and Mobile Applications

Switch options (for more details, see Chapter 10 in [22]) are a special form of real options that can be used to model the value of keeping multiple alternatives available. This applies to valuing general *process flexibility*, defined as the ability to switch among alternative inputs[4], as well as *product flexibility*, which refers to the ability of manufacturing multiple products in response to changing market demands. When switching costs are absent, exercising the option affects only the current payoff but not any subsequent (switching) decisions.

Switch options have been applied to the management of different modes of operation (e.g. ability to produce different materials according to market conditions) for investments [14][9]. The first of these two papers utilizes a stochastic dynamic programming to encapsulate the value of flexibility to switch between the modes. The second paper examines the option to change the quantity of resources used for production in response to changes in the product demand.

In this section we present the usability of switch options in pervasive sensor applications. We first describe our methodology for the switch options, and explain how the value of various alternatives can be found. We then evaluate the proposed methodology in two applications: (*i*) dynamic service selection in sensor networks, and (*ii*) switching operational modes in a pervasive security application.

3.1. Switch Options Methodology

Switch options constitute a promising modeling and decision making approach for service and operation mode selections that are made during the pervasive application's operation, in the presence of volatile network and environmental conditions. However, the application of this method requires a model for quantifying the value of an alternative service or operating mode.

3.1.1. Quantifying the Value of a Selection:

The value of a selection can be measured by the difference between its benefits and the costs incurred by its execution. In the case of service selection in sensor networks, the benefits rely on *Value of Information* (VoI) that is application-specific and depends on the importance, quality and security

[4] e.g. switching among various types of fuel that a factory may use for its operation

of a service's output [18]. On the other hand, costs of accessing a service include any energy spent in processing and communication with the provider of the service, as well as the delay for the transmission of the output that it provides. Hence, the *value* (V) of a service S is:

$$V(S) = V_{\text{inf}}(S) - \alpha(t)E_s(S) - \beta(t)D(S) \; , \tag{3}$$

where V_{inf} represents the VoI of the output that S produces, E_s represents the energy that is spent by this service, and D is the time it takes for the output of S to reach the requesting service. $\alpha(t)$ and $\beta(t)$ act as unifying parameters for the different units of the above components. They are also application specific and describe the relative importance of energy and delay to the application at a specific point t in time. For example, if the application becomes time-critical, the β value will increase to penalize the service instances with high delay. Similarly, for services with low energy left in the sensor on which they are implemented, the α value will increase. Furthermore it is entirely possible that the information value ($V_{\text{inf}}(S)$), energy spent ($E_s(S)$) and the delay ($D(S)$) vary during the operation; hence they are also time dependent, although this is not explicitly shown in the equation. In this work we are dealing with a simplified, linear valuation model but other valuation techniques can also be applied, a task that we leave for future work. Furthermore, in the case of switch options in other domains, different types of costs are more relevant. We will discuss these in more detail for the application of switching options in the pervasive security applications.

3.1.2. Switch Options Modeling

The value of being able to switch to different operating modes or actions during the development of a project is the extra value that can be gained once the switch is exercised. For example, for two operating modes A and B, of which the former is more valuable at the beginning of the project, the extra value of keeping B as an alternative, as long as there are no switching costs, is $V_{\text{Option}} = \sum_{t=t_0}^{t_n} V_{A\to B}(t)$, where $V_{A\to B}(t)$ is the expected value of the extra gain from using B instead of A when market conditions suggest so, and time series $t_0 \ldots t_n$ denotes the times at which the switch was made. Of course, this value is obtained only if choosing B is expected to be more beneficial than keeping A, otherwise the switch will not occur.

The above approach can readily be applied to sensor service selection or pervasive applications. However, it becomes apparent that the network and environmental conditions that drive the switching decisions should be assessed before any selection of sensor service instances or operating modes can take place. We name this task the *test phase*, which is followed by the actual *selection* and *execution* phases. These phases are discussed in the remainder of this section.

Test Phase: once the possible sensor services or the operation modes that provide the necessary functionality are detected, the test phase executes all alternative selections either all at once or one after the other, to estimate the costs that they incur and their value of information. The test run lasts for a set period of time, the length of which determines the cost of performing this evaluation. Often, the conditions for switching between the instances may require that the possible selection alternatives are run all at once.

The cost of a selection is determined during the test phase by: (i) accumulating the energy consumption and delay for processing and communication of the information packets that are relayed along the path that connects the service and the user, for the domain of service selection, and (ii) accumulating monetary and manual labor costs, for other application domains (such as modes of operation in a security application). The value of information that is provided by a selection is also assessed during the test phase, thus the value $V(S)$ (as in Eq. 3) can be computed.

From the above discussion, it is apparent that the test phase (and our proposed method based on switch options described herein) cannot be applied in cases when the sensor service, or a mode of operation, is time-critical and short-lived. For example, there is no opportunity to conduct a test phase when one needs to monitor the break-out of a fire in a forest. On the other hand, for long-lived sensor tasks such as temperature or soil contamination monitoring, the costs incurred by the test period are reclaimed by the future gains of switching from one selection to another.

The gains from different selections are estimated using the measurements of the test phase as follows. Let $C(t)$ be a random variable representing network and environmental conditions at time t that define *VoI* of options A and B at time t, denoted by $V_A(C(t))$ and $V_B(C(t))$, respectively. Also let J be the random variable representing switching signal to switch from option A to B ($J(t) = 1$) and from B to A ($J(t) = -1$), or no action ($J(t) = 0$), at time t. Most often, the switching signal is a function of $C(t)$, and its quality might depend on which options are active, as only currently active sensors can provide input to the computation of function J. Furthermore, let T denote the duration of the test phase. During that time, we make a series of n measurements of $C(t_i)$ and compute $J(t_i)$, where $t_i = i \times T/n$, as well as $V_A(C(t_i))$ and $V_B(C(t_i))$. Based on these measurements and computations, we create the piecewise linear approximations $v_A(t)$ and $v_B(t)$ of $V_A(C(t_i))$ and $V_B(C(t_i))$ for t in $(0, T)$. Likewise, we create a piecewise constant function $s(t)$ (stating currently active option, either A ($s(t) = 0$) or B ($s(t) = 1$)) defined for $t_i < t \leq t_{i+1}$ as 0 if $i = 0$ (we always start with option A active), or, for $i > 0$, $s(t_i) = \min(\max(J(t_i) + s(t_{i-1}), 0), 1)$, where min and max are used merely to keep the result 0 or 1. Then, the benefit per time unit of having an option to switch from A to B is given by the following integral:

$$V(A \rightarrow B) = \int_{t=0}^{T} \frac{s(t)(v_B(t) - v_A(t))}{T} dt \ . \qquad (4)$$

If amortization cost per time unit of having option B (which includes cost of switching between options as well as cost of measurements necessary to generate switching signals) is lower than $V(A \rightarrow B)$, option B is worth having. If c_{AB} and c_{BA} denote the switching costs from A to B and B to A, respectively, then the switching cost between options A to B normalized over time T is given by the following sum:

$$c(A \rightarrow B) = \sum_{i=2}^{n} \frac{J(t_i)}{2T} [(1 + J(t_i))c_{AB} - (1 - J(t_i))c_{BA}] \ .$$

Indeed, when $J(t_i) = 1$, we switch from option A to option B, and when $J(t_i) = -1$, we switch back, so at those times we need to add the cost of switching, and $J(t_i)(1 + J(t_i))/2$ yields 1 if and only if when $J(t_i) = 1$, and similarly $-J(t_i)(1 - J(t_i))/2$ yields 1 if and only if when $J(t_i) = -1$.

Finally, if the cost of measurements of switching signals at time t is $m(t)$, then the cost of measurements per time unit is:

$$m(A \rightarrow B) = \sum_{i=1}^{n} \frac{m(t_i)}{T} \ .$$

Often, $m(t)$ is independent of t, so $m(t) = m_c$ and then $m(A \rightarrow B) = \frac{n \times m_c}{T}$.

The above mathematical model captures a direct relationship between the signals used for switching decisions and the value of the switch option. The test phase is required for calibrating the values of switching option and of switching signals. Better these signals are, larger percentage of the benefit of switching to the best solution are. Furthermore, the cost of amortization and achievable benefits of switching determine whether the alternative services or modes of operation are worth keeping.

Execution Phase: once the feasible choices for a set of selections are narrowed down based on the lowest estimated cost from the test phase, then the chosen subset of possible selections are executed and the gains are computed according to Eq. 3, during the execution phase. Once conditions for replacement of services (or operating modes) are specified, they can be monitored continuously as the operation continues. Moreover, monitoring of the environmental phenomena may offer information about the kinds of events requiring switching. For example, this could be evident in the case of high humidity being detected during the test phase that affects one service more than another. Knowledge of such a result can be used to switch automatically when humidity increases in the monitored area. Note that such effects depend on the environment in which the sensor network is deployed, making the test phase essential. Furthermore, to be able to switch between services or operation modes, it is not necessary to run all alternatives during the execution phase. The environmental conditions that are necessary to decide on switching are monitored separately (which constitutes a cost as given in the above mathematical model). The switching points however are *learned* in the *test phase*, where multiple options are run together to see which is more advantageous at which condition.

Algorithm 1 Algorithm to Set Indicator Signals During Test Phase or Switch During Execution Phase

```
method set_indicator_switch(time t_i, condition C(t_i))
if t_i in Test Phase then
  Calculate V_A(C(t_i)) and V_B(C(t_i)) for options A and B at C(t_i)
  Update E[V_A(C(t_i))] and E[V_B(C(t_i))]
else
  if E[V_A(C(t_i))] > E[V_B(C(t_i))] then
    Switch to (or stay at) A
  else
    Switch to (or stay at) B
  end if
end if
```

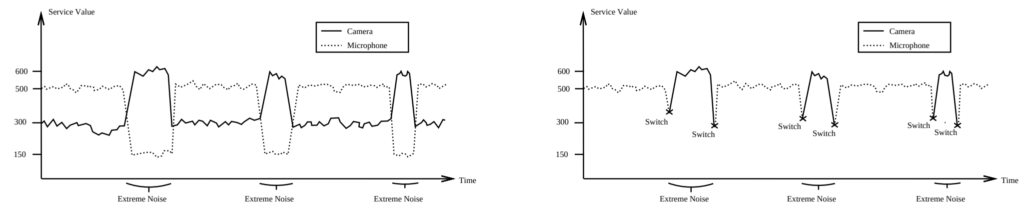
The pseudocode in Algorithm 1 highlights our switching methodology. This algorithm receives the current time (t_i) and environmental conditions ($C(t_i)$) and checks if the test phase is still in progress. If so, then the VoI is calculated for both options (since we assume simultaneous run during the test phase), and the expected value of VoI (denoted by $\mathbb{E}$) for both options (which will function as switching indicator signals later on) in the current conditions is updated. If the execution phase is in progress however, the expected values that were calculated during the test phase are examined to decide on switching between available options.

Our proposed methodology consisting of emphtest and *execution* periods relates to Machine Learning methods studied in literature [1]. The novelty of our approach consists of combining techniques for learning switching points during the test phase presented there with the theory of switching developed in Economy for evaluating the feasibility of alternative options (i.e. whether they should be kept at all).

3.2. Switch Options for Service Selection

A Service-oriented Architecture (SOA) approach [11] to wireless sensor networks abstracts them into a set of software services, each of which provides a well-defined functionality and might be deployed on one or more sensor nodes. Service selection is based on the assessment of the processing and communication costs that are incurred when a service instantiation in a given sensor node is chosen as part of a composition graph. There were previous methods proposed for making an efficient choice of operator for general distributed computations (e.g. [12][19][21][24]). However, these methods have not taken into account *the operational uncertainty* arising in sensor network deployments, which directly affects any estimates of service costs. Operational uncertainty in sensor networks arises from several causes, including (a) a lack of accurate knowledge of the computational and communication resources of sensor nodes and their residual energy that gets depleted over time, (b) changes in the environmental conditions in which nodes are operating, such as background noise that affects the quality of sensor readings (e.g. audio signals), and (c) the changes in the Value of Information (VoI) that the output of a composite service carries for a particular user at a given point in time. We propose the use of switch options to deal with such uncertainty in sensor services.

To demonstrate the application of switching options theory to the domain of sensor service selection, we first provide a real world scenario, and then



(a) Service Value Fluctuations for Two Types of Monitoring Service

(b) Value of Switching Option During the Test Period

Figure 3: Expected Test Period Results for Parking Garage Monitoring Application

present the benefits of the switch options methodology on a simulation based on this scenario.

3.2.1. Real-World Scenario

As a real world application of switch options to service selection, we chose a covered parking garage monitoring network. There are two types of services in this network: (*i*) a microphone service of readings from an acoustic sensor to monitor the sound volume, and (*ii*) a camera service that provides views of the area covered by the microphone monitors. We consider a pervasive monitoring application during which automated service selection may choose to utilize one or both of the services in monitoring.

An illustration of expected test period results for this application is shown in Figure 3a. The test period measures how external factors affect the VoI and benefits produced by each service. Clearly, the camera view for an area will produce the best results, but running a camera is a costly operation (due to its energy consumption and maintenance costs) in mid-term applications, so the benefit of using this service is low when no events are happening in the garage. Figure 3a also shows that when the VoI of the microphone service drops below a certain level, there is loud noise in the garage, and the VoI provided by the camera service increases. Of course, loud noise often signifies an important event in which case the VoI (the first factor in Eq. 3) increases even more, giving the camera service a higher benefit than normal. On the other hand, the microphone gives faulty measurements when the noise level is high. This example shows how the switch option balances the *Value of Information (VoI)* provided by the services with the cost incurred by it.

Once the costs of running multiple services in each area during the test period have been incurred, the service selection mechanism can switch between the microphone and the camera service, and can do it with increased efficiency based on information about the conditions that are beneficial for switching gathered during test period. In the long run, the costs of the service test phase is compensated by clever switching actions, which it enables with the information it acquires. Figure 3b shows the advantage that could have been gained had switching information been available during the test period shown in Figure 3a. The area difference between the curves of Figure 3b and the curve of the microphone service (since it is best on average) in Figure 3a quantifies the extra value of the switching option.

3.2.2. Evaluations

We conduct a simulation-based evaluation of our service selection method via switch options, which is based on the parking garage example described in Section 3.2.1. Our goal is to assess the relative gains in service value obtained by using this new approach compared to the naive method of selecting a service based on its current value, as well as the optimal approach that always has complete knowledge of the value of services in a noisy environment. The setup that we simulate includes a microphone and a camera monitoring service for one area of the parking lot; events triggering system responses are set up to investigate how the service value of the microphone or camera changes. A test period was established to measure the switch points according to the sound levels in the environment. These levels are hence associated with the signaling function $J(t)$, as given in Section 3.1.2. Here, $C(t)$ (i.e. the environmental conditions) is the sound level in the system at time t. This can be measured by both the microphone and the camera service, since we assume the camera service also encapsulates a microphone; hence the conditions of the environment can be measured by the chosen service during execution phase. This allows for the chosen service signal that it should be switched with the other alternative during execution. In another application with different environmental indicators, extra effort may be needed to monitor conditions which signal the switching of service selections. The cost of this extra effort was also mentioned in Section 3.1.2, which may make keeping the option infeasible if too costly. While we try to make reasonable assumptions regarding the evolution of noise level in our target environment as well as the characteristics of the process that generates events of interest, our goal is not to exhaustively study all possible statistical distributions and their parameters for these components, but rather gain a qualitative understanding of the performance of these strategies. A realistic assessment of the service selection approaches can only be performed in a deployed sensor

Type	Inter-arrival (secs)	Sound Level	Length of Period (secs)
Event	exp(mean=100) $\rightarrow$ av: 100	200+[40 x (Γ(k=90,θ=1/3))] $\rightarrow$ min: 200, av: 1400	20 x $\mathcal{N}$(μ=1,σ^2=0.1) $\rightarrow$ av: 20
Non-Event	exp(mean=40) $\rightarrow$ av: 40	200+[40 x (Γ(k=10,θ=0.5))] $\rightarrow$ min: 200, av: 400	20 x $\mathcal{N}$(μ=1,σ^2=0.1)] $\rightarrow$ av: 20

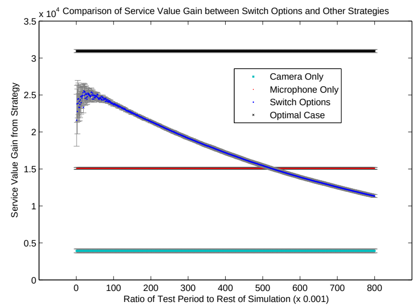
Table 3: Sound Change Properties for Service Selection Experiment

network, a task that we leave as future work.

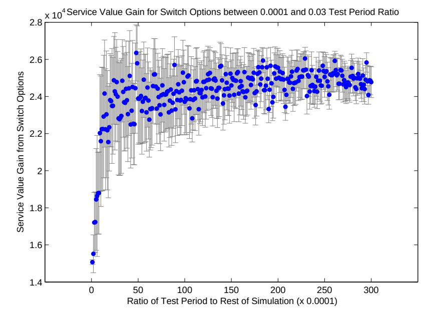
In our experiment, the results of which are shown in Figures 4a and 4b, the simulation time is fixed at 100,000 seconds, and the ratio of the test period to the remainder of the simulation time is varied among the runs (we ran 10 cases of 100,000 seconds for each test period length), within the range of values that is depicted on the x axis in the figures. We introduce two types of sound level changes to the environment, on top of the ground noise sound level, which is constant with a value of 200. The first type represents an increase in the environmental noise without any actual event of interest taking place (for example no car driving in or out of the parking lot but outside noise due to nearby construction is increasing). The second type of sound level change that we simulate represents a significant event of interest that occurred in the environment, for example detection of car movement in the parking garage. The properties of these sound level change types are given in Table 3.

As discussed above, the value of information (VoI) consists of two components: one that is subjective and describes the utility as assessed by the user, and another one that denotes the objective quality of information (QoI) that data carries. VoI is then represented as the product of these two components. For our experiments, a constant QoI of 0.9 is assumed for the camera sensor, while the microphone has a QoI that changes according to $1 - (\frac{SoundLevel}{1000.0})^2$, which accounts for loss of quality with high sounds. The utility for the camera sensor service is set to 1.0 upon occurrence of an event, while for the microphone it is constant at 0.2 when no event of interest occurs. Regarding cost, we set it to 0.45 and 0.15 for the camera and the acoustic sensor respectively, which includes both energy and delay as per Equation 3. The value $V(S)$ of each service is computed by the simulator as $V(S) = utility \times QoI - cost$ for each time unit.

In Figure 4a, the results for service values corresponding to different lengths of the test phase (an average over 10 cases, with 95% confidence intervals as error bars) are given for four strategies: the switch options ap-



(a) Comparison of Switch Options for All Test Period Lengths



(b) Gain from Switch Options for Test Period Lengths up to 3% of the Rest of Simulation Period

Figure 4: Service Selection Experiment Results

proach, two strategies that select either the camera or the microphone service exclusively, and the optimal approach that has complete and accurate knowledge of the VoI and the costs. Evidently, choosing either the microphone or the camera service provides less value than switching between them during the system's run time. This is to be expected, since a system that does not make use of the test phase does not know if and when it should switch between alternatives. Such decisions in our experiments are made via the expected value by each service at a given sound level. Although the test phase is indeed useful, as it gets longer, the value gains decrease due to the excess cost of running (and testing) both services. Figure 4b shows the results (again with 95% confidence intervals as error bars) of a closer look to the varying lengths of the test phase to see the value that peaks the gain for this setting (via comparing them with shorter ones for the same experiment). From the graph, it appears that when the length is too short (for example only 0.1% of the actual system run time), the gain in value is rather small. This is due to the switching options approach not being able to determine when to make a switch, as a consequence of limited experience with events.

3.3. Switch Options for Pervasive Security Applications

In this section, we consider another application of switch options, the selection of operation modes in a pervasive security system. A house, a gallery or any other secure space where certain valuables are stored are often monitored by a security firm which charges a certain amount of money per unit of time for its services when the security system is in alarm mode.

We assume that in addition to the per time fee, the security firm charges also switch costs, a fixed cost fee charged when the alarm is raised and the response crew is dispatched to the monitored area. This is a perfect case for applying switch options because being able to set good indicator values of when to alarm the security firm is crucial to protecting valuables with acceptable cost of protection. False alarms cost money for security charges, but missed security breaches result in the loss of valuables. This increases the importance of *test period* during which the alarm indicators are learned.

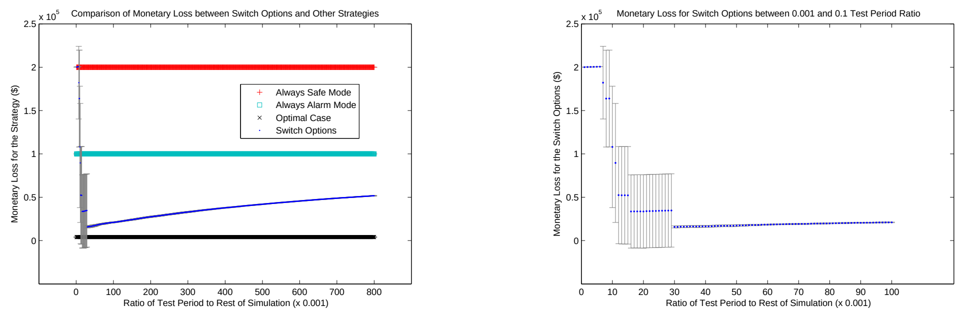
We assume a security application where a microphone is used to measure sound levels in the system. The changes of sound levels in the monitored area indicate a potential security problem (e.g. theft). We simulated the security application for 100,000 minutes, and again introduce two types of sound level changes to the system: (*i*) security breach, and (*ii*) safe situation whose properties are detailed in Table 4.

Type	**Inter-arrival (min)**	**Sound Level**	**Length of Period (min)**
Security Breach	600+exp(mean=600) → min: 600, av: 1200	200+[40 x (Γ(k=90,θ=1/3))] → min: 200, av: 1400	5+[35 x $\mathcal{N}$(μ=1,σ^2=0.1)] → min: 5, av: 40
Safe Event	300+exp(mean=300) → min: 300, av: 600	200+[40 x (Γ(k=20,θ=0.5))] → min: 200, av: 600	5+[15 x $\mathcal{N}$(μ=1,σ^2=0.1)] → min: 5, av: 20

Table 4: Security Application Sound Change Parameters

The costs of running the alarm operating mode is set in our simulation as follows. According to the sound level in the area as an indicator, the security firm charges $50 for switching to alarm mode (i.e. initial service fee), and $1 for each minute that it checks upon the monitored area. Furthermore, we have set the total value of valuables in the monitored area to be $200,000; hence, the losses amount to this value if the switch to alarm mode does not happen during a security breach. These values stay the same during the test phase, meaning that for each minute spent in test phase, the security firm charges $1 to learn the parameters for the indicators (i.e. the sound levels that indicate a security breach, hence the switching to alarm mode is necessary).

The results of the experiment are presented in Figure 5 (with 95% confidence intervals as error bars). We varied the length of the test period to see how it affects the cost of lost valuables (i.e. being less-than-necessary cautious to switch to alarm mode) and the payment to the security firm (i.e. being too cautious). We ran 10 cases of 100,000 minutes for each test period length. Figure 5a shows that initially losses are very high because the learn-



(a) Monetary Loss for All Test Period Lengths

(b) Monetary Loss for Test Period Lengths up to 10% of the Rest of Simulation Period

Figure 5: Comparison of Switch Options to Other Strategies in Terms of Monetary Loss for the Pervasive Security Application

ing of necessary indicator values (i.e. sound values that require a switch) has not been completed yet. However, once the test period length reaches about 3% of the total simulation time (as shown in Figure 5b that is the close-up of Figure 5a), it again starts to increase with the length of test period. This is because the security firm is paid \$1 for each minute of the test period and this cost is not recovered by the slightly better tuned indicator values. Hence, there is an apparent trade-off between the costs of the test period and the gains received by learning when to switch. Furthermore, Figure 5b shows the *step* stair shaped cost curve and the sudden drops of losses happening after a sequence of the same monetary losses for different test period lengths. Those appear because certain lengths of the test period were necessary to learn important switching cases for the rest of the simulation.

Figures also show that staying always in the safe mode causes the loss of valuables (\$200,000 average monetary loss), while keeping always the alarm mode on incurs the payment of \$100,000 to the security firm. The switch options may result in higher losses than alarm always on strategy when the very short test periods causes the loss of valuables. However, with longer test periods, the switch option strategy yields losses very-close to optimal (lowest loss line in Figure 5a corresponding to the optimal switch decision, therefore unattainable in reality) results are achieved.

4. Conclusion

This paper discusses resource allocation in pervasive sensor applications. We propose the application of market mechanisms to address problems of resource allocation from the very low level of a sensor node to the high level pervasive sensor applications. We have simulated an application of auctions at the sensor node level and found it to be very efficient in resolving congestion. We furthermore presented the application of *switch options* to dynamically select advantageous services during the operation of sensor applications, and changing modes of operation in pervasive security applications. Our future work includes improvement and real world application of the proposed methods.

5. Acknowledgments

This research was sponsored by the US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, Cambridge, MA, 2004.

[2] C. Bisdikian, L. Kaplan, M. Srivastava, D. Thornley, D. Verma, R. Young, *Building Principles for a Quality of Information Specification for Sensor Information*, in: Proc. Int. Conf. on Information Fusion, 1370-1377, 2009.

[3] L. Chen, L., B. K. Szymanski, J. Branch, *Auction-Based Congestion Management for Target Tracking in Wireless Sensor Networks*, in: Proc. IEEE PERCOM, 2009.

[4] L. Chen, Z. Wang, B.K. Szymanski, J. Branch, D. Verma, and R. Damarla, *Dynamic Service Execution in Sensor Networks*, The Computer Journal, 53(5): 513-527, 2010.

[5] C. T. Ee and R. Bajcsy, *Congestion Control and Fairness for Many-to-One Routing in Sensor Networks*, in: Proc. ACM Int. Conf. on Embedded Networked Sensor Systems (SenSys), 148-161, 2004.

[6] D. Gesbert, S. G. Kiani, A. Gjendemsjo, and G. E. Oien, *Adaptation, Coordination, and Distributed Resource Allocation in Interference-Limited Wireless Networks*, in: Proc. of the IEEE, 95(12): 2393-2409, 2007.

[7] S. C. Geyik, B. K. Szymanski, P. Zerfos, and D. Verma, *Dynamic Composition of Services in Sensor Networks*, in: Proc. IEEE Int. Conf. on Service Composition (SCC), 242-249, 2010.

[8] S. C. Geyik, B. K. Szymanski, P. Zerfos, and A. Mowshowitz, *Sensor Service Selection through Switch Options*, in: Proc. IEEE Int. Conf. on Service Composition (SCC), 717-724, 2011.

[9] J. M. Harrison, J. A. Van Mieghem, *Multi-Resource Investment Strategies: Operational Hedging under Demand Uncertainty*, European Journal of Operational Research, Vol. 113, Issue 1, 17-29, 1999.

[10] T. He, J. A. Stankovic, C. Lu, T. Abdelzaher, *SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks*, in: Proc. IEEE Int. Conf. on Distributed Computing Systems, 46-55, 2003.

[11] J. Ibbotson, C. Gibson, J. Wright, P. Waggett, P. Zerfos, B. K. Szymanski, D. J. Thornley, *Sensors as a Service Oriented Architecture: Middleware for Sensor Networks*, in: Proc. IEEE Int. Conf. on Intelligent Environments, 2010.

[12] S. Kalasapur, M. Kumar, B. A. Shirazi, *Dynamic Service Composition in Pervasive Computing*, IEEE Trans. on Parallel and Distributed Systems, Vol. 18, No. 7, 2007.

[13] K. Karenos, V. Kalogeraki, and S.V. Krishnamurthy, *A Rate Control Framework for Supporting Multiple Classes of Traffic in Sensor Networks*, in: Proc. IEEE Int. Real-Time Systems Symp. (RTSS), 2005.

[14] N. Kulatilaka, *Valuing the Flexibility of Flexible Manufacturing Systems*, IEEE Trans. on Engineering Management, Vol. 35, Issue 4, 250-257, 1988.

[15] C. Lu, B. M. Blum, T. Abdelzaher, J. A. Stankovic, T. He, *RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks*, in: Proc. IEEE Real-Time and Embedded Technology and Applications Symp., 55-66, 2002.

[16] J. MacKie-Mason and H. Varian, *Pricing Congestible Network Resources*, IEEE Journal on Sel. Areas in Communications, 13(7): 1141-1149, 2002.

[17] G. Mainland, D. C. Parkes, and M. Welsh, *Decentralized, Adaptive Resource Allocation for Sensor Networks*, in: Proc. Symp. on Network System Design & Implementation (NSDI), 2005.

[18] A. Mowshowitz, *On the Market Value of Information Commodities: III. Demand Price*, Journal of the American Society for Information Science, Vol. 43, 242-248, 1992.

[19] E. Park, H. Shin, *Reconfigurable Service Composition and Categorization for Power-Aware Mobile Computing*, IEEE Trans. on Parallel and Distributed Systems, Vol. 19, No. 11, 2008.

[20] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Englewood Cliffs, NJ, 1996.

[21] T. Repantis, Y. Drougas, V. Kalogeraki, *Adaptive Component Composition and Load Balancing for Distributed Stream Processing Applications*, Peer-to-Peer Networking and Applications, Vol. 2, No. 1, 60-74, 2009.

[22] E. S. Schwartz, L. Trigeorgis, *Real Options and Investment Under Uncertainty: Classical Readings and Recent Contributions*, MIT Press, Cambridge, MA, 2001.

[23] J. Shu and P. Varaiya, *Pricing Network Services*, in: Proc. IEEE Int. Conf. on Computer Communications (INFOCOM), 1221-1230, 2003.

[24] S. Soro, W. Heinzelman, *Camera Selection in Visual Sensor Networks*, in: IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, AVSS 2007, 81-86, 2007.

[25] B. K. Szymanski, S. Y. Shah, S. Geyik, S. Das, M. Chhabra, P. Zefros, *Market Mechanisms for Value of Information Driven Resource Allocation in Sensor Networks*, in: Proc. Int. Workshop on Information Quality and Quality of Service for Pervasive Computing, IQ2S 2011, IEEE Percom, 62-67, 2011.

[26] Y. Xue, B. Li, and K. Nahrstedt, *Optimal Resource Allocation in Wireless Ad-Hoc Networks: A Price-based Approach*, IEEE Trans. on Mobile Computing, 5(4): 347-364, 2006.