# USE OF AUCTIONS IN WIRELESS SENSOR NETWORKS

By

Syed Yousaf Shah

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

Approved:

______________________________

Boleslaw K. Szymanski, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

July 2011
(For Graduation August 2011)

# CONTENTS


LIST OF FIGURES . . . v

LIST OF TABLES . . . vi

ABSTRACT . . . vii

1. Introduction . . . 1
   - 1.1 Motivation . . . 1
   - 1.2 Dissertation Structure . . . 2
     - 1.2.1 Keywords . . . 2

2. Market Mechanisms in Sensor Networks . . . 3
   - 2.1 Introduction . . . 3
   - 2.2 An Auction Mechanism for Distributed Congestion Control . . . 5
   - 2.3 Switch Options for Dynamic Service Composition . . . 10
     - 2.3.1 Methodology . . . 11
     - 2.3.2 Real-World Scenario . . . 12
   - 2.4 Decentralized Collaborative Markets for Wireless Sensor Network Resources . . . 14

3. Traveling Auction Mechanism for Distributed Congestion Control in WSNs . . . 18
   - 3.1 Introduction . . . 18
   - 3.2 Methodology . . . 19

4. Winner Election Strategies for Auctions . . . 23
   - 4.1 Introduction . . . 23
   - 4.2 Evaluation Methodologies . . . 24
     - 4.2.1 Evaluation under ideal conditions . . . 24
       - 4.2.1.1 Methodology Description . . . 24
     - 4.2.2 Evaluation under real-life conditions . . . 25
       - 4.2.2.1 Methodology Description . . . 25
   - 4.3 Efficient Winner Election . . . 28
     - 4.3.1 Description . . . 28
     - 4.3.2 Avoiding combinatorial comparisons . . . 29

5. Evaluation Framework and Results . . . 33
   - 5.1 JAVA Simulator . . . 33
     - 5.1.1 Results . . . 33
   - 5.2 Simulator in NS2 . . . 38

5.2.1 Framework . . . 38
5.2.2 NS2 Simulator Results for static Auctions . . . 39
5.2.3 NS2 Simulator Results for Traveling Auctions . . . 44
5.3 Discussion . . . 46
5.4 Findings . . . 48
6. Conclusion and Future Work . . . 51
BIBLIOGRAPHY . . . 52

# LIST OF FIGURES

2.1 Vehicals approaching to intersection . . . . . 6

2.2 Auction Based Bandwidth Allocation Framework . . . . . 10

2.3 Nodes publishing options . . . . . 11

2.4 A Real World-Scenario for Switch Options . . . . . 13

2.5 Networks publishing options . . . . . 16

3.1 Congestion traveling along with the target . . . . . 19

3.2 Packet Replacement . . . . . 22

4.1 Top elements of two consecutive vectors are not related . . . . . 30

4.2 Top elements of two consecutive vectors are related . . . . . 30

4.3 Top elements of two consecutive vectors are related . . . . . 31

5.1 Comparison of static Lookahead and Traveling Auctions . . . . . 50

# LIST OF TABLES


5.1 Loss under competitive election strategies in Auctions . . . . . . . . . . . . . . . . . 33

5.2 Loss under competitive winner election strategies in Auctions . . . . . . . . . . . . 39

5.3 Average VoI Loss Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

5.4 Average VoI loss for mission under different mobility models . . . . . . . . . . . . 44

5.5 Traveling Auctions vs Static Lookahead Auctions . . . . . . . . . . . . . . . . . . . . 45

5.6 Average VoI Loss Comparison . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

# ABSTRACT

The use of market mechanisms to solve computer science problems such as resource sharing, load distribution and network routing, is gaining significant traction. In this thesis, we investigate new market mechanisms to solve the problem of bandwidth sharing in wireless sensor networks (WSNs) in event-driven scenarios. We first demonstrate that a previously proposed strategy that greedily selects winners in repeated routing auctions is not globally optimal and then propose and evaluate a *lookahead* mechanism for winner selection in auctions. This mechanism improves upon the winner selection process for intelligently allocating network resources in WSNs. We experimentally show that there is a trade-off between the efficiency of the winner selection process and the depth of lookahead. Finally, we examine a frequent problem encountered by object tracking applications in WSNs, traveling congestion, in which congestion moves along with bursts of data from node to node. We design and evaluate a novel mechanism which we call *Traveling Auctions* to alleviate the problem of traveling congestion.

# CHAPTER 1
# Introduction

## 1.1 Motivation

We are no doubt living in the age of information explosion. Most of the experts agree that general knowledge is doubling every two years. Which of the information is of interest to an individual or a group of individuals depends on large number of factors. These factors eventually filter out for us the valuable information out of the information explosion. Some of these factors are the "timeliness" or "freshness" of information, accuracy of information and uninterrupted availability of information. The combination of these various factors define the value of the available information. Lets take an example from our everyday life. We pay for the latest news in different forms, e.g., newspaper, news cast etc, but as the news gets older it gets less valuable to us. News agencies, T.V channels, live feeds etc. do their best to get latest information and cash the freshness of the information accordingly. Similarly, we not only pay for the freshness of the news, but the accuracy of the news, and so on. In other words, the amount we pay for the news we get is a reflection of how we value the news, which means we do not pay for the information we get, we pay for the value we get out of certain information or news in our example. That is just one daily life example, there are large number of other fields, e.g. stock markets, flight/train schedules, currency exchange rates, surveillance and security etc. where various parameters collectively define the measure known as *Value of Information (VoI).*

The area of Wireless Sensor Networks (WSNs) is one of large number of areas which are sensitive to the value of the information. The fragile and resource constrained nature of WSNs makes it a challenging problem to deliver information with value that is as close as possible to the value expected by the end-user. As information flows through the network from source to destination, it incurs VoI losses in the process. The work presented in this thesis is motivated by the problem of reducing VoI loss and thus enabling efficient resource sharing in WSNs. The work focuses on design and development of mechanisms that will decrease the loss in the value of information in WSNs that may occur due to factors endogenous or exogenous to the network. For this purpose we introduce new market based strategies and improve upon already proposed strategies that we will discuss in this thesis.

## 1.2 Dissertation Structure

The first chapter of the thesis focuses on motivation and introduction to the problem. The second chapter presents approaches and applications of market mechanisms in WSNs, some of them are already being worked out and some of them proposed as research directions. Chapter3 discusses *Traveling Auctions* mechanisms for WSNs and in particular the *Traveling Auctions* mechanism for highly bursty event driven traffic in WSNs. Chapter4 compares already existing and new strategies for winner election in auctions. In Chapter5, we describe our simulation framework and present results for various experiments that we have conducted to test our approaches under various conditions. Chapter6 concludes the thesis and presents future directions of the work.

**Note:** The Chapter2 is collaborative work with other group members denoted on the first page of the chapter. Rest of the chapters show my own contributions to the area under the supervision of my academic and research advisor Prof. Boleslaw Szymanski.

### 1.2.1 Keywords

Wireless Sensor Networks (WSNs); Value of Information (VoI); Quality of Information (QoI); Auctions; Switch Options; Real Options; Destination-Sequenced Distance-Vector Routing (DSDV).

# CHAPTER 2
# Market Mechanisms in Sensor Networks

The problem of decentralized resource allocation has been studied widely in communication networks. Congestion based pricing mechanisms have been suggested in the literature [1, 2] to avoid congestion not only on tollways or in airports but also over communication networks. The goal for such systems is to balance externalities imposed by increased traffic. Flat rate pricing, prevalent today, is not an optimal strategy for service providers [2]. Fixed pricing schemes can lead to overuse of bandwidth by exploitation of TCP at the user end, and switching to congestion pricing or differentiated QoS would help in avoiding externalities. Various mechanisms to avoid Internet congestion have been suggested (although none have thus far gained much real-world traction)[3, 4]. Most of these mechanisms rely on congestion feedback from the network to the user, a mechanism that is not feasible in WSNs due to the intermittent nature of congestion caused by event-driven traffic in such networks.

In this chapter we discuss application of market mechanisms on various levels of the WSN architecture. We propose strategies for how market mechanisms can be applied in WSNs and we show the feasibility of the proposed mechanisms by examples.

## 2.1 Introduction

Deciding how to allocate resources to maximize Value of Information (VoI) for the users of the network is a challenging problem in WSNs. There can be many users with limited trust, and the value of information to each user may be subjective and user specific. A reasonable way of scaling up resource allocation in such domains may eventually be to deploy market mechanisms that take decisions based on the explicit worth that users are willing to assign to the information. However, designing mechanisms that allow for both the appropriate amount

---

* This chapter previously appeared as: B. Szymanski, S.Y. Shah, S.C. Geyik, S. Das, M. Chhabra and P. Zerfos. Market mechanisms for value of information driven resource allocation in sensor networks. In The Third International Workshop on Information Quality and Quality of Service for Pervasive Computing, IQ2S 2011, pages 62-67. IEEE, 2011.

of information disclosure as well as for efficient outcomes of the information disclosure is a challenge in itself. As proof of concept and to stimulate discussion in the community, we present three different market mechanisms that can be deployed at three different levels of WSN architecture.

The Value of Information is most easily thought of as the amount a user who uses information from the network to make decisions would be willing to pay for the information. Therefore, it is reasonable to think about it as the *change in expected utility* of a decision-maker who receives the information. For example, consider a police chief deciding how many cars to allocate and where to deploy them on a given night. A wireless sensor network could pick out threats, aiding in more effective deployment of vehicles, saving police department resources. The same level of crime prevention could be achieved using significantly fewer resources, and the savings in this case could be thought of as the value of information provided by the WSN.

We note some important preliminaries here. First, we distinguish between quality of information (QoI), which can be represented by a vector of objective measurements, and VoI, which depends both on objective QoI as well as on 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. For our purposes we assume that VoI is externally specified or provided by an oracle, while QoI is measured by the sensor network itself.

Second, we define three levels in WSN architecture at which we address the resource allocation problem using market mechanisms. These three levels also correlate with different time ranges for the solutions, varying from tens to hundreds of milliseconds at the first level, tens of seconds and minutes at the second level, and days and months at the third level. The lowest level is that of a single sensor node involved in sensing and routing. The goal at the node level is to collect the set of the most meaningful measurements or to transmit the most valuable packets first when congestion arises at a node. 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), and 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. We demonstrate how the use of auctions on the very nodes where temporary congestion occurs can help resolve congestion and therefore minimize the loss of VoI.

At the mid-level, networks perform more complex functions in the form of services. Basic services are often hosted by multiple nodes, and executing them commands significant resources (most importantly energy, but also bandwidth, computing power, and sensing modes). More complex applications can be composed from basic services by properly interconnecting the basic services, forming a service oriented architecture [5]. We suggest the use of switch options to decide the efficient interconnection of basic services as network conditions change dynamically.

Finally, at the high level of the WSN itself, market mechanisms may help to optimize network deployment and sharing. Sensor networks are spatial by nature and in cases where many different authorities are responsible for adjacent or overlapping spatial domains, the optimal deployment of sensor networks and access to the data that they collect are important issues. We show that the use of *real options* could lead to the creation of proper incentives for rational deployment and sharing of sensor networks.

## 2.2 An Auction Mechanism for Distributed Congestion Control

As discussed above, resource allocation is driven by VoI and VoI is subjective: it 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.

In this section, we consider the low level management of resources in WSNs, as exemplified by bandwidth allocation in congestion scenarios. Following Chen *et. al* [6], we consider public safety and emergency response needs of a VIP visiting a large city. 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. Every agency has its own mission and priority monitoring targets. Local police may be interested in monitoring traffic violations in the area as well as mob behavior, whereas a federal agency is tasked with detecting any kind of coordinated terrorist attack and dealing with high-level threats. State agencies may also be monitoring large vehicles entering the area, as well as individuals or cars with suspicious mobility patterns. Another state agency is monitoring the visitor's car, also a high priority mission. 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 causes packet loss, decreasing QoI of the traces of the targets that are of high value at that point in time. This problem will become more complex when there are more targets with coordinated mobility patterns involved.

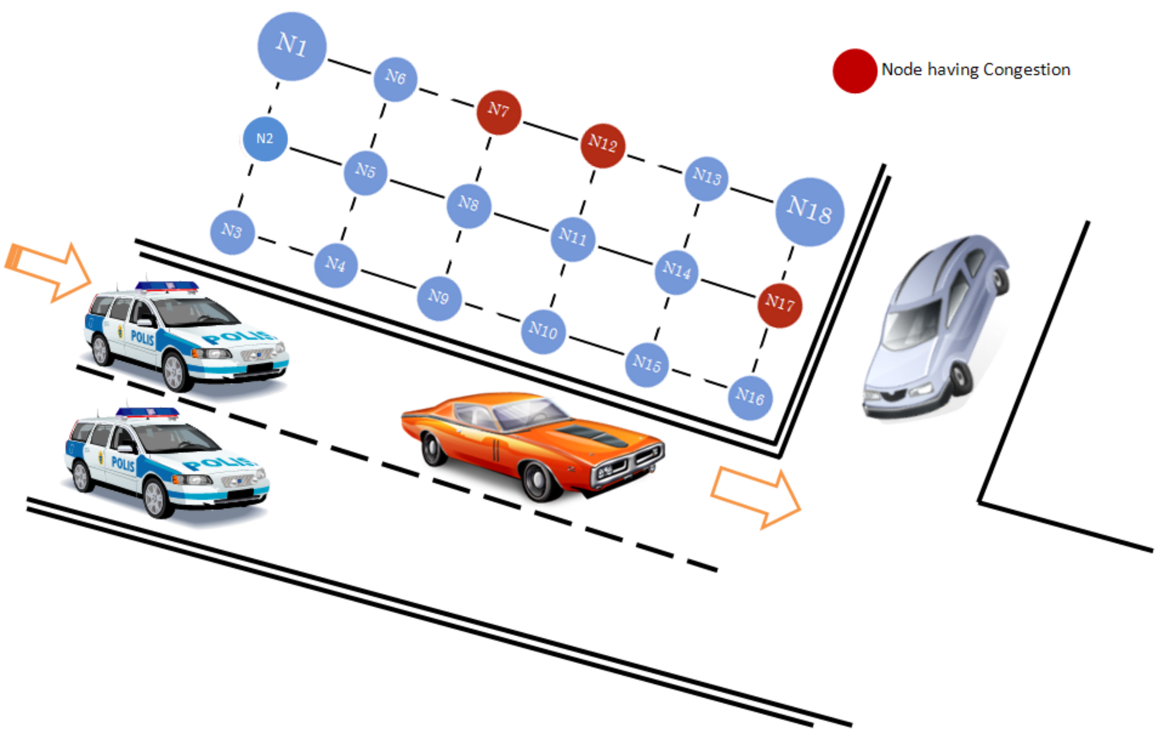



**Figure 2.1: Vehicals approaching to intersection causing congestion**

Chen at al [6] formally define an auction mechanism to solve this problem, which we summarize here. They consider two possible goals: efficiency (minimize total loss of information value) and equity (equalize the loss of information value for all missions). 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 a mission $i$ is a linear function of the QoI received about mission $i$, $U(i) = v(i)q(i,d)$, where $v(i)$ is the mission specific multiplier and $q(i,d)$ is the QoI for the specific mission with the data transmission delay for the mission of $d$ (although many more parameters define the value of $q(i,d)$, we explicitly refer to $d$, as this is the parameter directly impacted by congestion; other parameters, such as precision of target's position or precision of the time of measurement are not affected by congestion). Theoretically, even in this simple example, it would be useful to have $v(i)$ dependent on the information, for example, the closer the suspicious vehicle is to the visitor's car, the higher the VoI of its position. We make one further simplification in what follows. We assume that each node maintains only one packet for each mission because when two distinct packets of the same mission are received at the destination, the one with the more recent target data brings VoI to the same value, regardless if the other one was received or not. Hence, each node's maximum queue length is limited by the number of active missions, a requirement easily satisfied for 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. Since the standard congestion control techniques involve feedback loops, they are not applicable here because of the intermittent nature of congestion that keeps moving from one node to another in WSNs.

Now, we need to quantify the loss of information due to packets being delayed by congestion. To do so, we need to assess VoI of a piece of information. Often, VoI is most directly related to how much it changes the uncertainty or beliefs of the user of the information. For example, consider a police team monitoring potential threats to a VIP. A WSN reports the position and direction of movement of a suspicious vehicle, enabling the police to have a current estimate of the trajectory of this vehicle. How much does additional, new information about the position and direction of motion help?

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. Let us compare the case where the last observation by a node in the network was 10 minutes ago with a case where the last observation was 10 seconds ago. Clearly, new information will be more useful in the former case than the latter. Intuitively, this is because new information in the former case leads to a greater reduction in uncertainty about the object position than it does in the latter case. 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.

measure the loss of value as proportional to the additional delay incurred by the packet in congestion. To quantify, let $t_m$ denote the time at which target was sensed at $\mathrm{loc}(t_m)$. After time $\Delta t$, its new location $\mathrm{loc}(t_m + \Delta t)$ is uncertain and could be limited to a 2D circle whose radius depends on $\Delta t$ according to kinematics with the most recent speed $v_m$ detected at time $t_m$ and with acceleration limited to $a_{max}$. To compensate for the time delay, the destination can estimate the current position of the target as $\mathrm{loc}(t_m + \Delta t) = \mathrm{loc}(t_m)) + v_m \Delta t$. Hence, the increased uncertainty of the target position caused by packet delays is

$$|\mathrm{loc}(t_m + \Delta t) - \mathrm{loc}(t_m))| \leq \frac{1}{2} a_{max} (\Delta t)^2.$$

Hence, the VoI decreases quadratically with the delay. Denoting by $C_i$ all the constants of the proportionality of the VoI to the square of the packet delay, the loss of VoI for mission $i$, defined as the difference between the VoI of a packet not delayed by congestion and the packet arriving with such delay, is $L_i = C_i(\Delta t)^2$. A node experiencing congestion can choose one of the following two ways to decide the order of packets to be sent at the congested node:

1. Equalizing utility loss among all applications, i.e., $\forall$ applications $i$ and $j$ $\Delta L_i = \Delta L_j$

2. Minimizing the total utility loss, i.e., $\min \sum_i L_i$

Chen et al [6] considered only linear term of the VOI loss dominant when the prediction of the future target position is not made. The implementation of this approach done by Chen at al [7] is depicted in Figure 2.2. We assume that each packet carries with it the priority of the mission for which it is reporting data. As proposed in [6], a Second Price Auction is held at the point of congestion. The bids are entered by the target update packets that compete for transmission slots at the congested node, using the predicted utility loss as a form of currency. The auction winning packet will receive the currently available transmission slot while packets losing the auction will obtain additional funds for the future auctions of transmission slots on the same node, as the auctions repeat for every transmission slot of that the node until congestion is resolved. The additional funds received by each losing packet are proportional to its VoI loss incurred by the delay of its transmission. It should be noted, that the congestion is resolved locally at the point of congestion. Conducting such distributed auctions at each node whenever congestion arises there minimizes the overhead of overall congestion control. In particular, no feedback loop is needed between the destination and the source. Such feedback loops, which are used in traditional congestion control protocols, would delay the deployment of congestion control process, thus increasing the VoI loss caused by congestion.

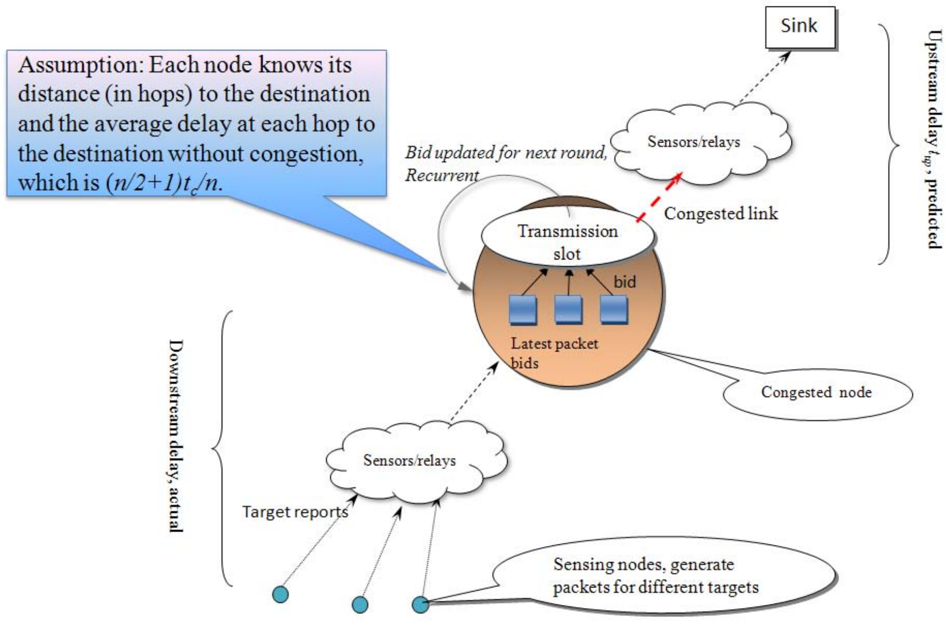



**Figure 2.2: Auction Based Bandwidth Allocation Framework**

Chen et al [7] show that the auction allocation mechanism for congestion control performs better than equal allocation or mission priority proportional allocation.

## 2.3 Switch Options for Dynamic Service Composition

Dynamic service composition for sensor networks was originally introduced by Geyik et al. [8] who characterized a service abstractly as a program running on a sensor node that requires a certain set of inputs and produces some data (output) characterized by a set of meta-data. An automated composition of higher functionality service creates a dataflow graph by interconnecting a set of services together. Choosing recursively the lowest cost of input providers (that is other services that are able to provide some of the required inputs) is termed *service selection*. This problem can be seen as the optimal selection of services rather than the generation of the optimal dataflow graph [8] because optimal interconnections can easily be chosen once the participating nodes are selected.

*Switch options* are usually defined as having multiple plans readily available for execution and draw a direct analogy to the service selection task that was described above. When making investments decisions, there could be multiple options (e.g. *expand*, *scale down* or *shut down* investment) to proceed, or to use different operating modes for the investment, e.g., having the option to produce or use different materials in the same factory, according to market conditions. The extra cost needed to make these options available should be taken into account when valuing a project. The *value* of a switch option is calculated as the sum of the benefits of switching from one option to another, whenever it is profitable.

We propose to use the switch option approach when choosing the input providing services. The currently selected services are considered an investment. The risks involved in choosing them are reduced by *periodically re-examining* their selection in view of the current network conditions and switching to a new service composition when profitable. Initially different compositions of services are considered but not chosen as suboptimal are kept as *switching options* and may be chosen in the future.

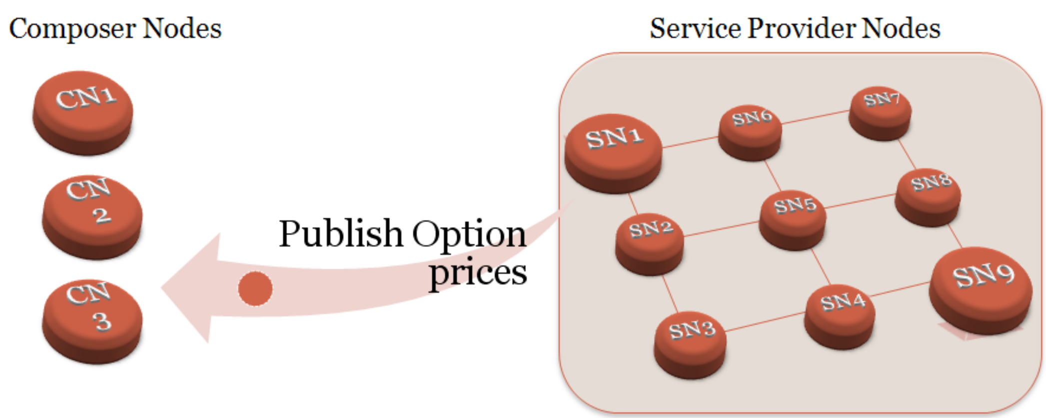



**Figure 2.3: Nodes publishing options**

### 2.3.1 Methodology

We propose two phases for service selection: (*i*) Test Phase and (*ii*) Service Selection Phase. The test phase runs multiple possible services and evaluates at what conditions switching between the input providers improves the VoI attained. Furthermore, the probabilities of these conditions to arise are also calculated, so we can determine how profitable it is to keep different selections available. Since an extra effort is expended during the test phase and the cost of this phase needs to be made up by the future gains, the length of this phase is limited by this requirement.

Service selection phase starts with the selection that has the highest expected gain. A switch to another selection is guided by the data collected during the test phase. Let the random variables $V_A, V_B$ denote the VoI of services $A$ and $B$ with the same functionality. Further, let $P(V_A < v_A, V_B < v_B)$ denote the probability that the network is at the state in which those random variables are less then or equal to $v_A, v_B$, respectively, and $p(V_A, V_B)$ denote the corresponding probability density. Then, the expected value of ability to switch from service $A$ to service $B$ is:

$$V(A \rightarrow B) = \int_{V_B=0}^{\infty} \int_{V_A=0}^{V_B} (V_B - V_A)\; p(V_A, V_B)\; dV_A\; dV_B\;,$$

which is in effect a summation over all $V_A > V_B$

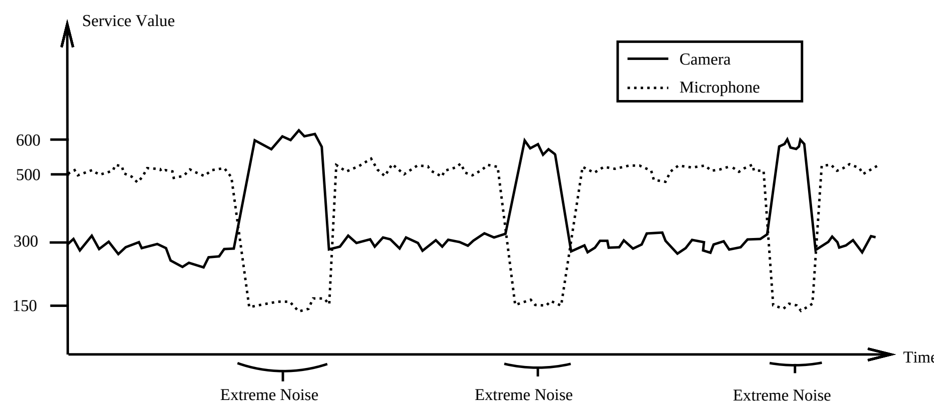
### 2.3.2 Real-World Scenario

Figure 2.4a shows a sensor network implemented in a covered parking garage. 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 long term monitoring mission during which automated service selection may choose to utilize one or both of the services in monitoring.

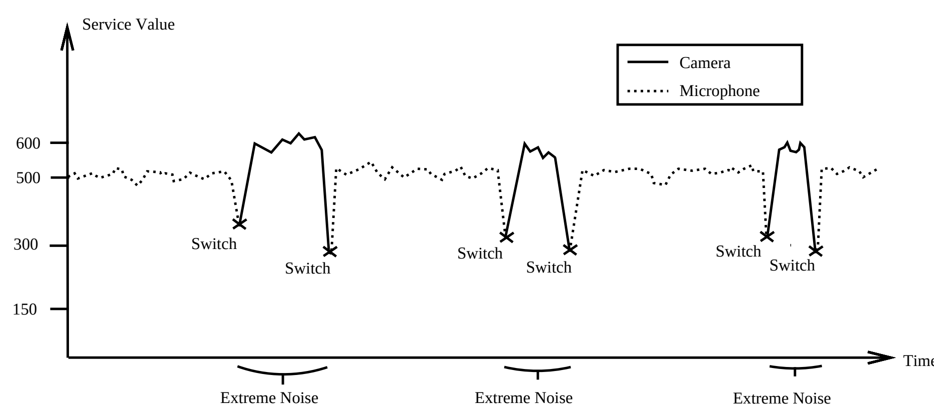
An illustration of expected test period results for this application is shown in Figure 2.4b. The test measure how external factors affect the VoI produced by each service. Clearly, the camera view for an area will produce the best results, but running a camera is a costly operation (due to its energy consumption and maintenance costs) in mid-term applications, so the gain from using this service is low to account for the cost. Figure 2.4b also shows that when the value of the microphone service drops below a certain level, there is loud noise in the garage, and the gain from using the camera service increases. Of course, loud noise often signifies an important event. Hence, the VoI (the first factor in the above equation) increases even more, giving the camera service a higher VoI 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


(a) A Parking Garage Example to Explain Switch Options Methodology




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




(c) Value of Switching Option During the Test Period

**Figure 2.4: A Real World-Scenario for Switch Options**

incurred, the service selection mechanism can switch between the microphone and the camera service, and can do it efficiently since it has information about the conditions that are beneficial for switching. In the long run, the costs of the service test phase will be compensated by clever switching actions, which it enables with the information it acquires. Figure 2.4c shows the advantage that could have been gained had switching information been available during the test period shown in Figure 2.4b. The area difference between the curves of Figure 2.4c and the curve of the microphone service (since it is best on average) in Figure 2.4b quantifies the extra value of the switching option.

## 2.4 Decentralized Collaborative Markets for Wireless Sensor Network Resources

Wireless sensor networks (WSNs) are typically deployed for the purpose of providing better information to a decision-maker. For example, consider a WSN deployed in a forest: the primary purpose of the WSN may be to monitor temperature and environmental conditions to detect forest fires and prevent them from spreading. Extreme temperature readings are a critical but rare event. It is absolutely necessary that the network performs well at detecting fires when they occur. At the same time, given the rarity of such a circumstance, the sensors in the network could also serve other roles as perhaps monitoring wildlife activity, rainfall levels, etc. Deploying the network is costly, so the benefits must outweigh the cost of deployment. These benefits are again measured as the Value of Information (VoI) provided to the decision-makers who use the network.

As WSNs start being deployed systematically, we must reason explicitly about the costs, benefits, and trade offs inherent in the process, as well as the sharing of the deployed network among multiple entities and for different purposes. The natural language for this reasoning is the language of economics, and we need to think about the market for information that is provided by a WSN, and how it can interact with the design of such a network. The key players in the marketplace are the service provider, who deploys and maintains the WSN, and users who value the information provided by the WSN and are willing to pay for it. Often one of the users may actually be the service provider. Continuing with our forest-monitoring WSN example, suppose the state of New York deployed a WSN in its forests, and the state of Massachusetts deployed one in its own. Many fires are capable of spreading across the Massachusetts-New York border, so it is important for each state to be aware of what is going on in the forests of its neighbor. Instead of deploying its own networks in the neighboring states, Massachusetts can acquire access to data from New York when it is critical, and vice versa. Yet, a blanket agreement to share all data is not necessarily a good idea, because producing and sending data is costly, and non-critical data that has value to New York may not necessarily be as valuable to Massachusetts. Obviously what is needed is a dynamic market mechanism capable of allowing trades between users and service providers, and allocation of resources that can be relied upon in emergency situations. While non-trivial, no other decentralized approaches can efficiently arrive at the right trade off between the value of information provided by WSNs and the costs of their deployment and use. In this section, we sketch a vision for some possible ways in which market mechanisms may help to solve the problem of resource allocation in WSNs, leading to appropriate payments in exchange for the VoI provided to users. For our purposes, we assume, as before, that the decision-makers who use the information provided by the WSN are in possession of well-calibrated estimates of the value of the information the WSN provides to them. Therefore, in our models we assume that the utility to the user is exogenously specified.

**Real Options for Emergency Information.** Let us return for a moment to the emergency forest-fire scenario discussed above. For simplicity, assume that Massachusetts and New York only have interest in each other's WSN information if it is potentially predictive of the existence of a forest fire in the neighboring state, and that the rest of the time they have no value for information from the other. However, since a forest fire is a critical emergency situation, the value of information related to forest fires is high. One possibility would be for the states to write and buy real options on the transmission of their data. For example, New York could write, and Massachusetts purchase, an option that enables Massachusetts to use up to 40% of the WSN resources available to New York.

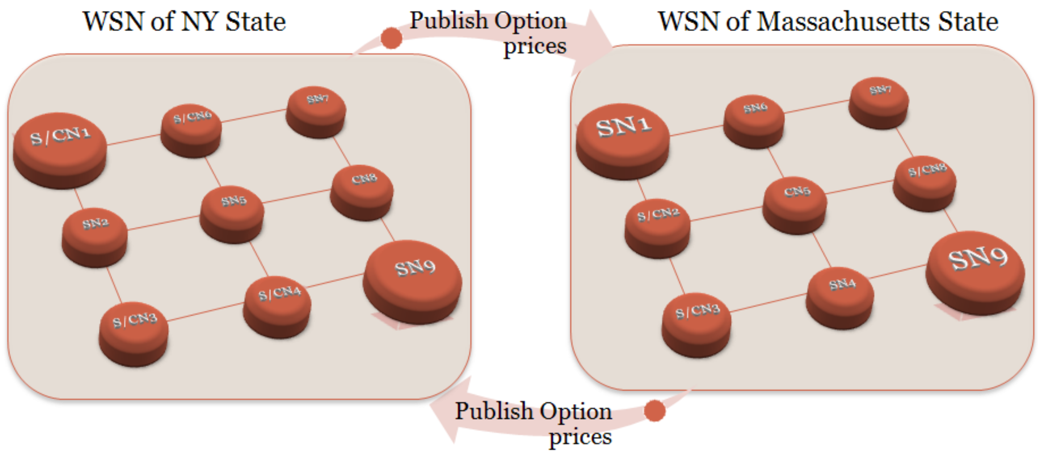



**Figure 2.5: Networks publishing options**

The option would be American style, so that Massachusetts could exercise it at any time that it becomes necessary, triggered by the existence of a forest fire that may spread to Massachusetts. Several interesting issues arise once we think of the option in these terms. (i) New York state would have to provision extra resources so that it can always reliably provide the services it is writing the option on. It must never be the case that Massachusetts cashes in on the option and New York is unable to provide the service. Therefore, the WSN may need to be over-provisioned for emergency situations when thinking about only the needs of the state itself. However, the cost of this over-provisioning can be made up for through the payments made to New York by Massachusetts when Massachusetts buys the options. (ii) Massachusetts must become aware of information that may be critical to its decision-making about whether or not to exercise the option. This leads to issues of trust between New York and Massachusetts –for example, if Massachusetts were to completely trust New York's words, and New York deceived Massachusetts into thinking a situation was an emergency when it was not, the option would be needlessly exercised. (iii) Standard option pricing methodologies may have to be extended to deal with the idiosyncrasies of real, deployed WSNs, for example the information needed to even decide whether to exercise the option, as discussed above. Further, especially when thinking of emergency response, which is necessary when *rare* events occur, standard methodologies based on assumptions of normality that use the variance of expected returns (or return-equivalents) may no longer be the right approach.

While the research questions raised are significant, real options methodology has the potential to deliver real value in several ways. It could lead to the establishment of a decentralized collaborative market that will guarantee the availability of information whenever it is needed. Options provide a market-based solution for resource allocation and management between competing or cooperative WSN service providers, potentially allowing them to recover infrastructure and resource costs encountered in deployment. If the market succeeds, the cost that each state will pay in the form of options in practice is less than the managerial and deployment cost of a full fledged WSN. Moreover this price will also be lower than the price a state will have to pay in case an event occurs across the border and the state wants to buy services at the time of event, when the seller will charge significantly higher prices in response to demand because the information has the highest value to safety. While we have discussed this so far in the context of two parties, the extension to multiple service providers and users is obvious.

# CHAPTER 3
# Traveling Auction Mechanism for Distributed Congestion Control in WSNs

## 3.1 Introduction

We have shown in Chapter2 that running distributed auctions on individual nodes prove to be efficient mechanism for resolving congestion. In the next chapter, we propose improvements to the exiting auction mechanisms. We have seen in the Chapter2 that in the proposed auction mechanism the congested nodes do not share any information about auctions held, so the auctions on one node are independent of the auctions on the other nodes; we call such auctions *static auctions* because the auction information stays on local node. We have worked out case scenarios in which event triggering objects generate burst of traffic along the line of their motion causing congestion to follow a trajectory in the network. The problem of congestion resolution becomes more interesting when the congestion systematically moves from one node to another. Consider an example of 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 traffic data. As the car moves forward, it enters into the sensing range of new sensor nodes and exits the range of sensor nodes behind it. This way the burst of data generated by sensor nodes moves in the network along with the target car moving on its way. These bursts of data generate congestion on the nearby nodes and this congestion moves along as the burst moves from point to point in the network. These bursts are of short duration. But we have seen that in typical static auction the nodes do not take into account the VoI loss accumulated by a mission on the other nodes, thus every node starts auction based on the information locally available on the node. This favors high priority missions on every node because one node has no information of which mission has won auction in its neighborhood. The missions other than of the highest priority tend to suffer compared to the scenario in which a node running auction would have known the VoI loss accumulated by the missions on other the nodes. To capture such concepts, we propose

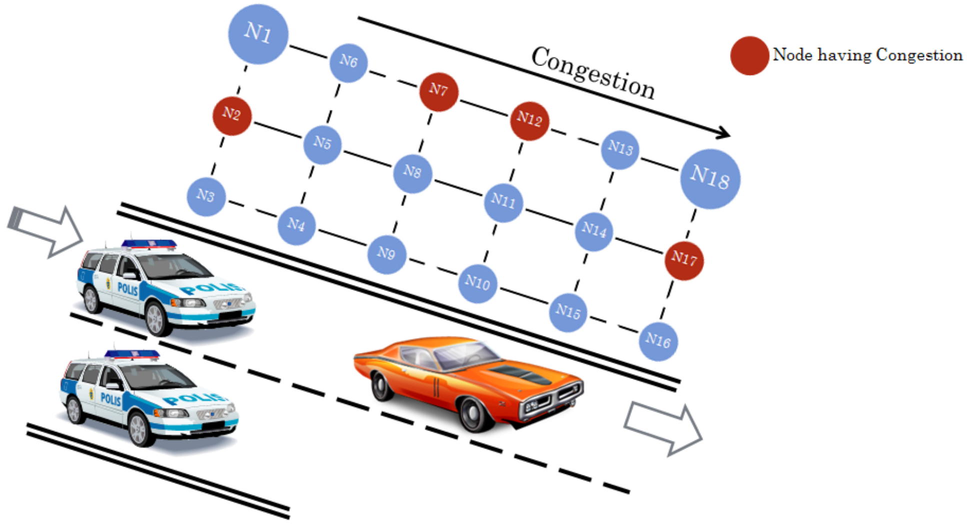



**Figure 3.1: Congestion traveling along with the target**

the notion of *Traveling Auctions* in which every node listens promiscuously to its neighboring nodes. This way every node running auction takes into account the status of auctions heard from the neighbors and has not only information locally available on the node but also the information about the auctions held on neighboring nodes. In such a configuration, the state of the auctions flows in the network along with the congestion which gives the effect of the auction traveling in the network.

## 3.2 Methodology

Lets discuss in more detail how the *Traveling Auction* mechanism works. Assuming TDMA protocol for channel access, every node will have one transmission slot to transmit its data. Hence if we have $n$ slots, each node will use one transmission slot over the period of $t_c$. So on average, transmission slot time $t_{slot}$ is $t_c/n$. Moreover, we are assuming DSDV as routing protocol, so each node knows its distance in hops from its final destination. Knowing all the aforementioned information, packet at each node can calculate its predicted arrival time at destination with a myopic assumption that when it leaves the current node there will be no congestion on way to destination. The time it takes a packet to reach its destination $t_{up}$ and it can easily be calculated using (3.1):

$$t_{up} = (Num.\ of\ hops\ to\ sink)(\frac{t_c}{2} + \frac{t_c}{n}) \tag{3.1}$$

Now suppose a node receives target report packets for multiple missions. Whenever the transmission slot comes, the node will deliver one of the packets waiting in the queue. Every packet in the queue will bid its loss of VoI using Eq.3.2 as a form of currency in auction run by the node for the transmission slot. Each mission packet will calculate its loss of VoI in case it wins auction and is transmitted and in case it loses auction and is delayed until the next auction.

$$bid_i = (tmi - tmpi)p_i v_i t_c \tag{3.2}$$

$$\Delta u_i^l = (t_{slot} + current\_time + t_{up} - t_{mpi} + \frac{t_c}{2})p_i v_i t_c \tag{3.3}$$

$$\Delta u_i^w = (t_{slot} + current\_time + t_{up} - t_{mi} + \frac{t_c}{2})p_i v_i t_c \tag{3.4}$$

When a packet wins the transmission slot, it adds VoI loss show by (3.4) to its total VoI loss while in case it loses the transmission slot, it adds VoI loss shown by (3.3) to its total VoI loss. ($t_{slot}$ + $current\_time$) represents the end time of currently available transmission slot. The $t_{up}$ is the predicted upstream delay given by (3.1). $p_i$ is the priority of the associated mission and $v_i$ is the velocity of the object being tracked by the mission. We denote the position measurement time at which the target object of mission $i$ was tracked by $t_{mi}$ whereas $t_{mpi}$ represents the previous position measurement time known to the sink of mission $i$ before the current one, which is $t_{mi}$. The node experiencing congestion acts as an auctioneer for the transmission slot; it tries to minimize the total utility loss of all the missions. Whenever a mission wins an auction, the $t_{mpi}$ pertaining to this mission changes and is updated to the $t_{mi}$ of the packet that is sent through this node. In static auctions, this $t_{mpi}$ remains on the local node due to which other nodes use their own value of $t_{mpi}$ for the mission $m_i$. As the difference between $t_{mi}$ and $t_{mpi}$ is critical to define the bid for a mission in traveling auctions, we share the $t_{mpi}$ for missions among the neighboring nodes in the new Traveling Auctions mechanism. This way if a mission wins auction on one node, the neighbors know the latest $t_{mpi}$ and if the auction moves to the neighboring node, the mission that won the latest mission will not get extra advantage due to its high priority. This way the state of auction is not static on nodes but it travels from node to node and when auction starts on a node, the node already knows the state of auctions on its neighbors and thus makes more accurate decisions. The sharing of this information is done via each node overhearing its neighbors.

What normally happens in routing level is when a node forwards a packet to another node, all the neighbors of the sender node listen to it and read the packet's header to check the destination address of the packet against their own addresses; nodes or a node to which the packet is forwarded receive it (as the destination address of the packet and node's address will match) and the rest of neighbors discards the packet. So when one node forwards the packet of certain mission $m_i$ to another node, all the neighboring nodes listen to it and they can read from the header of the packet the mission id for which the packet was forwarded and the *timestamp* which is used by every node to update the $t_{mpi}$ value for the mission held locally. For this purpose we have extended the header of the packet in *NS2*. The IP header also carries the *timestamp* when the packet was generated so the neighbors also know the $t_{mi}$ of the mission for which packet was forwarded. Since we have our customized traffic generator, we can assume that the time at which the packet was generated is the same as the time when the position of the target object was measured. We use the IP header *timestamp* to avoid extra bytes for the timestamps in our specialized header, otherwise the measurement time can also be included in specialized header or can be stored in data part of the packet. This way the $t_{mpi}$ propagates along with congestion and every node not only has local information but also the information that propagates to it from the neighbors all the way from the point where the congestion first occurred.

If the source generates packets at a high rate, we do not have packet drops in our approach because if there comes a new packet for a mission for which we already have a packet in the queue waiting to be delivered, we replace the old packet by the new packet if the object measurement time carried by the new packet is greater than the one already in the queue. We replace the old packet with the fresh one because the more current information sink has about the object the lesser will be VoI loss. The figure (3.2) illustrates the VoI losses that we save when we replace the old packet by new one, in other words when we deliver the latest available packet.

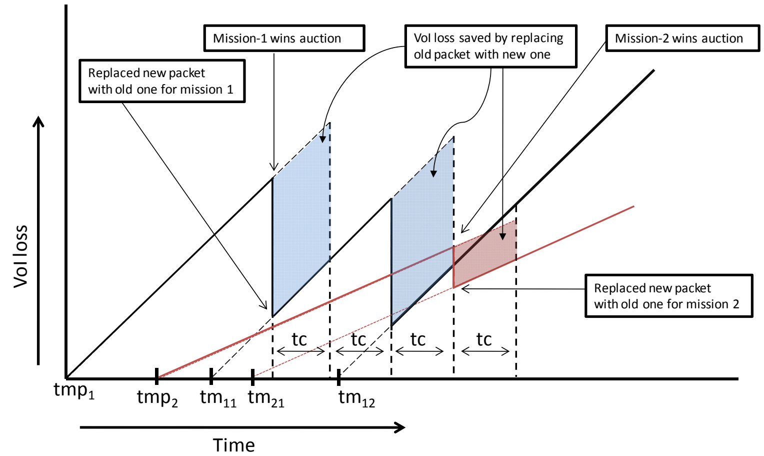



**Figure 3.2: Replacing old packet by fresh packet; $tmp_1$ and $tmp_2$ are the previous measurements or zeros whereas $tm_{11}$ and $tm_{21}$ are the first measurements of mission-1 and mission-2 respectively.**

As we have seen in above paragraphs, we share through overhearing the state of the auctions among the nodes; this way when congestion moves from one node to another the auction also moves from one node to the other node because the state of the auction moves along to the other node influencing that node's winner election decision. With such a mobile congestion and auction mechanism we can efficiently share the network bandwidth in a highly bursty event driven scenarios. We show in chapter-5 the results achieved under *Traveling Auctions* mechanism.

# CHAPTER 4

# Winner Election Strategies for Auctions

## 4.1 Introduction

We have discussed in general the use of auctions in WSNs. Researchers have proposed Market Based Approaches for congestion resolution in WSNs [7]. Auctions are being used to efficiently allocate network resources to demanding applications. The applications may vary in number and priorities. The work we have seen in Chapter2 suggests winner selection in auctions based on greedy strategy [6, 9], so whenever there is a transmission slot available to transmit a packet, a winner is elected out of the contending applications that has the highest VoI loss in the current auction and packet pertaining to this application is routed through the node since sending the packet of such application saves the most of VoI loss.

However, we have found that selecting winners in auctions greedily is not always optimal. As the applications may very in number and priorities, there exist sets of priorities and number of missions for which greedy strategy is sub-optimal. In this chapter we will evaluate and compared auction winner selection strategies that are Greedy, Greedy Lookahead in general and in particular Lookahead-1 (LA-1) and Lookahead-2 (LA-2). We have developed two kinds of simulators and collected experimental data for aforementioned strategies for winner selection in auctions. We have evaluated the strategies with diverse set of parameters including number of bidders and diverse set of priorities in order to analyze the behavior of different strategies under different set of parameters. We first briefly describe the competitive strategies as follows.

1. **Greedy Strategy:** As mentioned earlier, in greedy strategy, application that has the highest VoI loss in current auction is selected as winner and packet pertaining to this application is routed through the node since sending the packet of such application saves the most of VoI loss. So the greedy strategy has myopic view of the VoI loss in current auction.

2. **Greedy Lookahead Strategy:** In contrast to the greedy strategy, lookahead strategy

also considers results of the future auctions and decides the winner for the current auction based on the total VoI loss as result of all possible sequences of winners in current auction and the $m$ future auctions. Suppose we have $n$ missions or applications and lookahead of $m$, then if $m = i$ $lookahead - i$ strategy will consider the highest VoI loss $i \leq n$ missions as possible winners in the current and next $i$ auctions, assuming other auction, will succeed the current auction. The first mission of the sequence of winners that saves the maximum VoI loss is selected as the winner for the current auction. We will show in 'Efficient Winner Election' section how to calculate the winners efficiently and how we tackle the indigenous computational complexity of lookahead methodology.

In the following sections we describe our implementation approaches and in Chapter5, we present the results achieved under Greedy, Lookahead-1 and Lookahead-2 approaches.

## 4.2 Evaluation Methodologies

We evaluated the strategies under two conditions. First, we consider ideal conditions where packets for all the applications are received at regular intervals and second, we consider real life scenarios where packets received at certain node depend on the objects being tracked in the sensor network.

### 4.2.1 Evaluation under ideal conditions

#### 4.2.1.1 Methodology Description

In this evaluation method we assume that the node where congestion occurs receives packets at regular intervals for all the applications. The packets received contain information about the tracked object associated to a particular application. Therefore we inherently assume the packets received at nodes are independent of mobility patterns adopted by the tracked objects. Moreover, we also assume that the packets come every $t$ $ms$ and transmission slot is sufficiently large to transmit a packet. The simulator for this strategy is implemented in JAVA. Each mission bids its loss of VoI as form of currency in the auction.

**Calculating Bids:** The bid value of mission $m_i$ in future auction depends if it was winner in the previous auction or not. So in any future auction every mission has two different bids.

Considering mission $m_i$ wins $(k-1)th$ auction or has lost no auctions {Win Bid}

$$b_{i,win}^{k} = b_i = p_i \tag{4.1}$$

Considering mission $m_i$ loses $(k-1)th$ auction {Lose Bid}

$$b_{i,lose}^{k} = (1 + l_i) * p_i \tag{4.2}$$

where $l_i$ is the number of auctions mission $m_i$ has lost. Please note that mission $m_i$ can lose any number of auctions from $l_i = 1...k-1$.

### 4.2.2 Evaluation under real-life conditions

The second approach we followed to evaluate the aforementioned strategies was to implement simulator in *NS2* with mobile nodes that move according to some mobility pattern. This approach is closer to the real life situation where a moving object is being traced via sensor network and the information is sent to a sink. In contrast to the earlier approach, there is no assumption of ideal situations. We use TDMA protocol in which the transmission slots come every $t_c$. The packets arrive at node whenever an object is traced by a sensor node and packet for a specific mission is sent to the sink.

#### 4.2.2.1 Methodology Description

The simulator is implemented in NS2 by implementing lookahead mechanism and greedy mechanisms as a separate queuing mechanisms for the nodes. There are 44 nodes in total and depending upon the number of missions specified, there are moving objects. The sensor nodes are stationary and have fixed positions whereas the mobile nodes are moving with certain mobility pattern to mimic real life object moving in a sensor network while being tracked by the sensor nodes. We use NS2 native DSDV as the routing protocol.

**Calculating Bids:** Let $p_i$ be the priority of the mission $m_i$ *and* $v_i$ be the velocity of the tracked object and assume that the transmission slot comes every $t_c$ time. Moreover, *tmi* is measurement time when a target object was tracked by a sensor node and *tmpi* is the measurement time for the latest mission specific packet received by the sink. Every mission has its own values for *tmi* and *tmpi* as the packets for each mission are sent to sink independently. The bid $b_i$ for the mission $m_i$ in current auction and in the future auctions is calculated as follows. Bid in current auction,

$$b_i = (tmi - tmpi) * p_i * v_i * t_c \qquad (tmi = tmi_t) \tag{4.3}$$

The bid value of mission $m_i$ in future auction depends if it was winner in previous auction or not. So in any future auction every mission has two different bids,

Bid in future auction if mission $m_i$ won the last auction,

$$b^k_{i,win} = (tmi_{t+k} - tmi_{t+k-1}) * p_i * v_i * t_c \tag{4.4}$$

Bid in future auction if mission $m_i$ lost the last auction,

$$b^k_{i,lose} = (tmi_{t+k} - tmpi) * p_i * v_i * t_c \tag{4.5}$$

where the superscript $'k'$ signifies the $kth$ future bid. So, if we have $lookahead - 2$ then we need to calculate two future bids that are $b^1_i$ and $b^2_i$, similarly if we have $lookahead - m$ we need to calculate $m$ future bids in addition to the current bid. Moreover *tmpi* for mission $m_i$ has to be updated before calculating the future auction bids if the mission $m_i$ was considered a winner in previous auction.

The packets received by node have time-stamp *tmi* which is the time when the information therein the packet was measured, it can be different from the time at which the packet was received at certain node. Moreover the packets can arrive in any oder at certain node so the packet with fresh information may precede a packet with older information due to the routing delays. It is important to note that calculating $tmi_{k+1}$ is critical. The $tmi_{k+1}$ needs to be predicted as it is not possible to calculate exactly because the object is moving and depending upon when the object was tracked the $tmi$ changes and a node cannot know when it will receive next packet for a mission and what measurement time that packet will carry. In order to predict the future $tmi$, we tried two techniques: last history based and packets per slot based techniques.

Future $tmi$ for mission $m_i$ using packets/slot on node,

$$tmi_{t+k} = current\_tmi_{mi} + k * \frac{\#Packets\ received\ for\ m_i\ at\ the\ node}{\#Transmission\ slots\ arrived\ at\ the\ node} \tag{4.6}$$

Future $tmi$ for mission $m_i$ using history on node,

$$tmi_{t+k} = current\_tmi_{mi} + k * (tmi - tmi_{t-1}) \tag{4.7}$$

Where $tmi_{t-1}$ is the target measurement time for mission $m_i$ just before the measurement with time $tmi$ was received at the node. We have run various experiments and found out that history based approach gives better results. The results presented in table in the *NS2 Results* section are based on the history based approach. We found the history based prediction approach for future $tmi$ is simple yet elegant.

**Finding Winner:** As mentioned earlier we use Greedy strategy and Lookahead strategy to find the winner of the auction. While using greedy strategy, we do pure greedy selection whereas to improve the results in lookahead mechanism, we use mixed strategy. We have noticed that during simulation it happens that lookahead strategy ends up with more than one optimal winner sequences starting with different missions. This means we can select as winner one mission or the other; in such a case we use greedy strategy by selecting a mission with highest VoI loss in the current auction. With such a mixed strategy, we save more VoI losses in case there is no auction in the future or there are not enough auctions in the future for the lookahead strategy to recover from the broken winners cycle. We will have more discussion on this in chapter-5.

## 4.3 Efficient Winner Election

### 4.3.1 Description

As we mentioned in the description of the lookahead mechanism in *Introduction* section, in order to decide the winner mission of the current transmission slot we not only take into account the bids of missions in current auction but we also calculate the bids in next coming auctions by considering every mission as winner in the current auction and thus up to '$m$' future auctions. The winner will be the mission that incurs the least loss in terms of '$m$' future auctions. We call this strategy '$m - lookahead$' where $m = 1, 2....$

We store future bids for each auction in a separate vector. Calculating auction winner can be combinatorially expensive operation when we have '$m$' vectors of future auctions bids for '$m - lookahead$'. Following the strategy, we will calculate bids not only for the current auction but also for $m - lookahead$ auctions. So if we have $m = 2$ we will calculate bids for two future auctions $k = 1, 2$ and the current auction which can also be called as $k = 0$. The bids for the current auction are stored in vector $B$ where bids are $b_1, b_2...b_n$ and the bids for future auctions are stored in vectors $V_1, V_2...V_m$ accordingly, where subscript denotes the lookahead auction number.

For any $kth$ future auction ($k = 1...m$), we calculate bids in $kth$ auction by considering that every mission $m_i$ for $i = 1...n$ can be loser in one or more auctions until $kth$ auction, that will change the bid of mission $m_i$ in $kth$ auction accordingly.

For every mission $m_i$ we store bids in $kth$ vector considering $m_i$ mission as winner in $(k-1)th$ auction and also loser in previous auctions; in the *Evaluation Methodologies*, we have described how to calculate the future bids.

After having all the vectors populated, we sort all the $m$ vectors and the vector $B$ in descending order ensuring that we know which bid pertains to which mission. Please note that the win bid denoted by $b^k_{i,win}$ in earlier section, for winner of the auction $k-1$ will go down in the sorted vector $V_k$ even if mission $m_i$ has high priority. We know $V_k^{iW} = b_i$

$$b_i < (1 + l_i) * p_i \tag{4.8}$$

$$\therefore V_k^{i_W} < V_k^{i_L} \tag{4.9}$$

Similarly,

$$V_k^{i_W} = (tmi_{t+k} - tmi_{t+k-1}) * p_i * v_i * t_c \tag{4.10}$$

$$V_k^{i_L} = (tmi_{t+k} - tmpi) * p_i * v_i * t_c \tag{4.11}$$

$$(tmi_{t+k} - tmi_{t+k-1}) < (tmi_{t+k} - tmpi) \qquad \because \; tmi_{t+k-1} > tmpi \tag{4.12}$$

$$\therefore V_k^{i_W} < V_k^{i_L} \tag{4.13}$$

because $V_k$ is sorted in descending order, the win bid for mission $m_i$ will always be lower in the vector than the lose bid.

### 4.3.2 Avoiding combinatorial comparisons

After calculating all the bids and sorting all the vectors, the winner of the current auction can be elected. The winning mission will be allocated the current time slot and its packet will be transmitted by the node. Winner will be the mission $m_i$ such that adding $b_i$ to any possible combination of winners in future $m$ missions causes the highest VoI loss. We will select such mission $m_i$ as sending this particular mission will save us the most VoI loss in view of future auctions. Obviously computing and comparing all possible combinations is very expensive.

It turns out that we do not need to check all possible combinations to find out the optimal sum and thus winning bid for the current auction. Knowing that all the vectors are sorted, we can find out the winner as follows,

- Compare the first element of $B$ vector with the first element of $V_1$; if $V_1[0]$ is not related to $B[0]$ means that $V_1[0]$ is not losing a bid in the previous round regardless if mission $m_i$ (mission having bid at $B[0]$) loses it. If that is the case, then $B[0], V_1[0]$ is optimal combination for $B$ and $V_1$. Repeating the same procedure for $m - lookahead$ vectors when top entries of any two consecutive vectors $V_k$ and $V_{k-1}$ are not related, then $V_K[0]$ is selected winner for $kth$ auction and the sum of top entries of all the vectors is the optimal sequence of winners thus the winner for the current auction is the mission with bid at position $B[0]$. Please note that this is the case where greedy is the solution. In each auction, we are selecting winner with the highest bid.

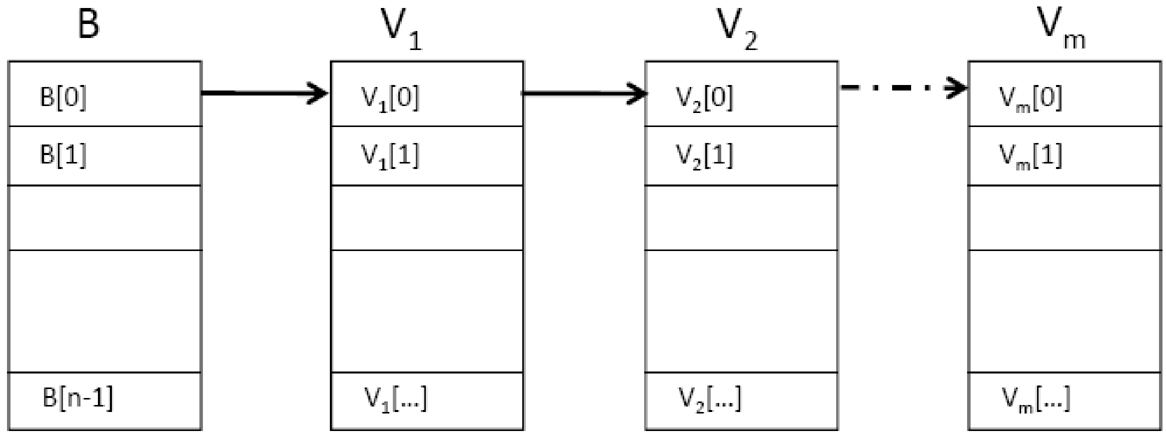



**Figure 4.1: Top elements of two consecutive vectors are not related**

- If bid $B[0]$ is related to $V_1[0]$ means that we cannot select $V_1[0]$ as the winner. Similarly, if for any two consecutive vectors $V_{k-1}[0]$ and $V_k[0]$ bids are related, we will select the either combination 1 or 2 below, whichever gives the largest sum.

  1. Compare the $B[0]$ with next two top entries in $V_1$ that are not related to $B[0]$. Similarly, for these two selected entries we further select two entries in the next consecutive vector. So for each selected entry in $V_{k-1}$, we further select two entries in $V_k$ such that entry in $V_{k-1}$ is not related to entry in $V_k$. Calculate sum for each combination and select the combination that gives the largest sum. So we end up with winner sequence starting at $B[0]$.

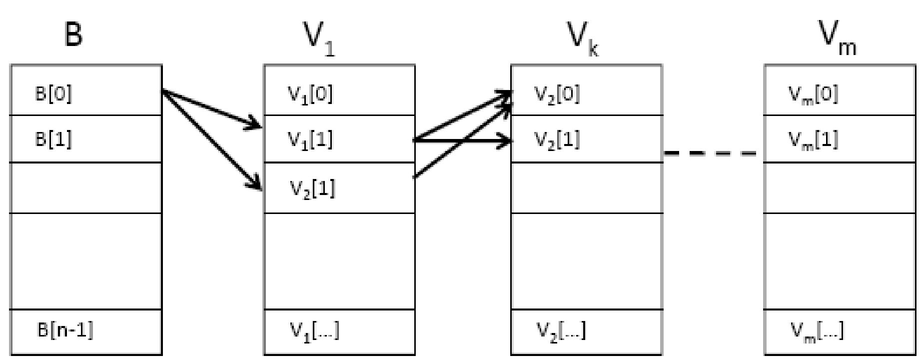



**Figure 4.2: Top elements of two consecutive vectors are related**

2. Select $B[1]$, $V_1[0]$ and another entry in $V_1$ that is not related to $B[1]$. For these two entries selected from $V_1$, select two top entries in vector next to $V_1$ that are not related to them. Similarly, for these two selected entries we further select two entries in the next consecutive vector. So for each selected entry in $V_{k-1}$ we further select two entries in $V_k$ such that entry in $V_{k-1}$ is not related to entry in $V_k$. Calculate sum for each combination and select the combination that gives the largest sum. So we end up with winner sequence starting from $B[1], V_1[0]$ *or* $B_1[1]$ and so on.

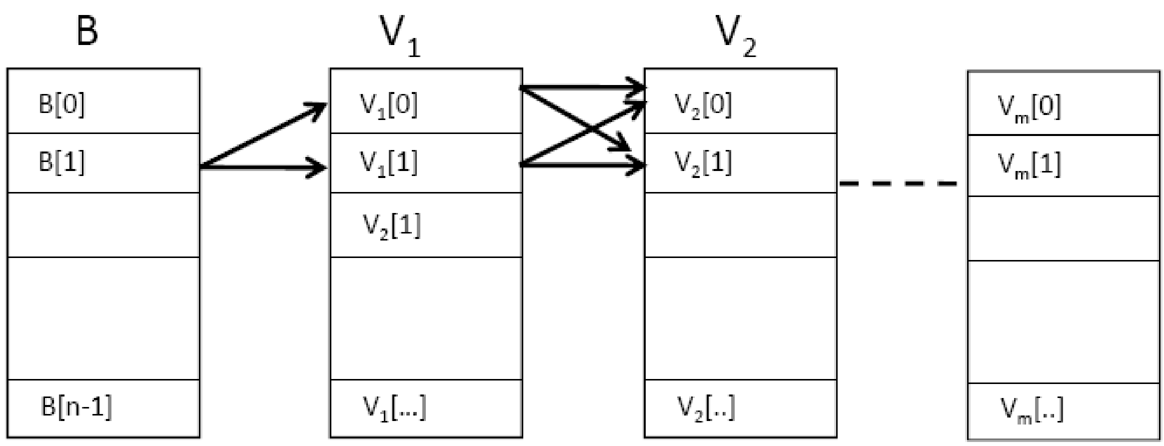



**Figure 4.3: Top elements of two consecutive vectors are related**

We only check combinations that originate from the first two entries of $B$ vector because of the following lemma.

***Lemma:*** In $m - lookahead$, strategy for any value of $m$, the combination of bids with largest sum always starts from one of the first two entries of the bids $B$ vector provided all the bid vectors are sorted in descending order.

**Proof:** Suppose for contradiction that the third entry in the $B$ vector ends up with combination having higher sum than the first two. Following all the possible combinations originating from first three elements for $B$ vector.

$$X \;\; = B[0] + V_1[0] + V_2[0] + ...V_m[t]$$

$$Y \;\; = B[0] + V_1[1] + ...V_m[t]$$

$$Z \;\; = B[1] + V_1[0] + ...V_m[t]$$

$$O1 = B[2] + V_1[0] + ...V_m[t]$$

$$O2 = B[2] + V_1[1] + ...V_m[t]$$

In case of $X$, where top entries of two consecutive vectors e.g. $B, V_1$ and $V_2...V_m$ are not related, X will always beat all the other combinations as all the vectors are sorted and adding the largest elements will always give the largest value.

If this is not the case, lets compare Y, Z and O1. We know that

$$Z > O1 \ and \ Z > O2 \ \because \ B[1] > B[2]$$

$$Y > O2 \ because \ B[0] > B[2]$$

We cannot say anything about $Y$ and $O1$, but no matter what, we have already got $Z$ that gives us larger sum than $O1$. The only situation in which $O1$ or $O2$ can be selected is when either $O1$ or $O2$ gives us such a combination having higher sum with future bids which cannot be made with $Y \ or \ Z$. But this is not possible, any combination that $O1$ can make with entries in succeeding lookahead vectors can be made by $Z$ as they both end up at the same entry in $V_1$. Similarly, we can get combination, that $O2$ can make with succeeding lookahead vector, using $Y$ which would be even larger in sum than $O2$ as $Y > O2$. Therefore in no circumstances the third entry of $B$ vector can end up in combination with larger sum than the first two entries. Similarly, considering any fourth, fifth or any $nth$ mission, we conclude that it will not produce larger sum than the first two entries, the sum will rather decrease as we go down the list in vector $B$. So our supposition was wrong and considering only the first two missions or first two entries of $B$ are sufficient to find the combination with the largest sum.

# CHAPTER 5

# Evaluation Framework and Results

In this chapter, we present results achieved by evaluating the strategies proposed in this thesis.

## 5.1 JAVA Simulator

This simulator is aimed to test the efficiency of the lookahead approach under ideal conditions. We have discussed the assumptions and evaluation methodology for this simulator in chapter-4. We conducted various experiments with diverse set of mission priorities and with 2, 3 and 4 missions. The *Rounds* in the following table denote the number of auctions run at the node. The missions are numbered in increasing order starting from 1 to the total number of missions in order in which they are specified in the *Priorities* column below. The *Loss/Auction* column shows the ratio of the total loss to the total number of auctions held during certain simulation run.

### 5.1.1 Results

**Table 5.1: Loss under competitive election strategies in Auctions**

| **Strategy** | **Priorities** | **VoI Loss** | **Loss/Auction** | **Rounds** | **LA-1 opt.** | **LA-2 opt.** |
|---|---|---|---|---|---|---|
| Lookahead-1 | 4,3,2,1, | 22480 | 14.98667 | 1500 | | |
| Lookahead-2 | 4,3,2,1, | 22480 | 14.98667 | 1500 | 1 | 0 |
| Greedy | 4,3,2,1, | 23834 | 15.88933 | 1500 | | |
| Lookahead-1 | 9,8,7,1, | 53246 | 35.49733 | 1500 | | |
| Lookahead-2 | 9,8,7,1, | 55011 | 36.674 | 1500 | 1 | 0 |
| Greedy | 9,8,7,1, | 55011 | 36.674 | 1500 | | |
| Lookahead-1 | 9,8,2,1, | 39686 | 26.45733 | 1500 | | |
| Lookahead-2 | 9,8,2,1, | 42789 | 28.526 | 1500 | 1 | 0 |

Continued on next page

**Table 5.1 – continued from previous page**

| Strategy | Priorities | VoI Loss | Loss/Auction | Rounds | LA-1 opt. | LA-2 opt. |
|---|---|---|---|---|---|---|
| Greedy | 9,8,2,1, | 44985 | 29.99 | 1500 | | |
| Lookahead-1 | 10,3,2,1, | 30797 | 20.53133 | 1500 | | |
| Lookahead-2 | 10,3,2,1, | 31326 | 20.884 | 1500 | 1 | 0 |
| Greedy | 10,3,2,1, | 34026 | 22.684 | 1500 | | |
| Lookahead-1 | 10,9,8,1, | 59898 | 39.932 | 1500 | | |
| Lookahead-2 | 10,9,8,1, | 61703 | 41.13533 | 1500 | 1 | 0 |
| Greedy | 10,9,8,1, | 61703 | 41.13533 | 1500 | | |
| Lookahead-1 | 15,10,3,2, | 62186 | 41.45733 | 1500 | | |
| Lookahead-2 | 15,10,3,2, | 62001 | 41.334 | 1500 | 0 | 1 |
| Greedy | 15,10,3,2, | 68160 | 45.44 | 1500 | | |
| Lookahead-1 | 12,5,4,3, | 50374 | 33.58267 | 1500 | | |
| Lookahead-2 | 12,5,4,3, | 50386 | 33.59067 | 1500 | 1 | 0 |
| Greedy | 12,5,4,3, | 53486 | 35.65733 | 1500 | | |
| Lookahead-1 | 24,12,6,3, | 93672 | 62.448 | 1500 | | |
| Lookahead-2 | 24,12,6,3, | 99168 | 66.112 | 1500 | 1 | 0 |
| Greedy | 24,12,6,3, | 102708 | 68.472 | 1500 | | |
| Lookahead-1 | 47,23,11,5, | 172608 | 115.072 | 1500 | | |
| Lookahead-2 | 47,23,11,5, | 187450 | 124.9667 | 1500 | 1 | 0 |
| Greedy | 47,23,11,5, | 187450 | 124.9667 | 1500 | | |
| Lookahead-1 | 30,16,9,4, | 123326 | 82.21733 | 1500 | | |
| Lookahead-2 | 30,16,9,4, | 124430 | 82.95333 | 1500 | 1 | 0 |
| Greedy | 30,16,9,4, | 124430 | 82.95333 | 1500 | | |
| Lookahead-1 | 65,64,27,8, | 335614 | 223.7427 | 1500 | | |
| Lookahead-2 | 65,64,27,8, | 347765 | 231.8433 | 1500 | 1 | 0 |
| Greedy | 65,64,27,8, | 358442 | 238.9613 | 1500 | | |
| Lookahead-1 | 17,9,8,2, | 76862 | 51.24133 | 1500 | | |

Continued on next page

**Table 5.1 – continued from previous page**

| Strategy | Priorities | VoI Loss | Loss/Auction | Rounds | LA-1 opt. | LA-2 opt. |
|---|---|---|---|---|---|---|
| Lookahead-2 | 17,9,8,2, | 80900 | 53.93333 | 1500 | 1 | 0 |
| Greedy | 17,9,8,2, | 80970 | 53.98 | 1500 | | |
| Lookahead-1 | 125,64,27,8, | 426414 | 284.276 | 1500 | | |
| Lookahead-2 | 125,64,27,8, | 449392 | 299.5947 | 1500 | 1 | 0 |
| Greedy | 125,64,27,8, | 449650 | 299.7667 | 1500 | | |
| Lookahead-1 | 40,30,20,10, | 224800 | 149.8667 | 1500 | | |
| Lookahead-2 | 40,30,20,10, | 224800 | 149.8667 | 1500 | 1 | 0 |
| Greedy | 40,30,20,10, | 238340 | 158.8933 | 1500 | | |
| Lookahead-1 | 17,13,11,7, | 107656 | 71.77067 | 1500 | | |
| Lookahead-2 | 17,13,11,7, | 110519 | 73.67933 | 1500 | 1 | 0 |
| Greedy | 17,13,11,7, | 110519 | 73.67933 | 1500 | | |
| Lookahead-1 | 30,15,10,5, | 128830 | 85.88667 | 1500 | | |
| Lookahead-2 | 30,15,10,5, | 134825 | 89.88333 | 1500 | 1 | 0 |
| Greedy | 30,15,10,5, | 133480 | 88.98667 | 1500 | | |
| Lookahead-1 | 36,25,6,5, | 147245 | 98.16333 | 1500 | | |
| Lookahead-2 | 36,25,6,5, | 159050 | 106.0333 | 1500 | 1 | 0 |
| Greedy | 36,25,6,5, | 159050 | 106.0333 | 1500 | | |
| Lookahead-1 | 64,27,4,3, | 171424 | 114.2827 | 1500 | | |
| Lookahead-2 | 64,27,4,3, | 190650 | 127.1 | 1500 | 1 | 0 |
| Greedy | 64,27,4,3, | 190650 | 127.1 | 1500 | | |
| Lookahead-1 | 61,47,23,11, | 300416 | 200.2773 | 1500 | | |
| Lookahead-2 | 61,47,23,11, | 307851 | 205.234 | 1500 | 1 | 0 |
| Greedy | 61,47,23,11, | 307851 | 205.234 | 1500 | | |
| Lookahead-1 | 44,33,22,11, | 247280 | 164.8533 | 1500 | | |
| Lookahead-2 | 44,33,22,11, | 247280 | 164.8533 | 1500 | 1 | 0 |
| Greedy | 44,33,22,11, | 262174 | 174.7827 | 1500 | | |

Continued on next page

**Table 5.1 – continued from previous page**

| Strategy | Priorities | VoI Loss | Loss/Auction | Rounds | LA-1 opt. | LA-2 opt. |
|---|---|---|---|---|---|---|
| Lookahead-1 | 20,9,4,1, | 63267 | 42.178 | 1500 | | |
| Lookahead-2 | 20,9,4,1, | 66889 | 44.59267 | 1500 | 1 | 0 |
| Greedy | 20,9,4,1, | 66889 | 44.59267 | 1500 | | |
| Lookahead-1 | 20,14,7,2, | 87062 | 58.04133 | 1500 | | |
| Lookahead-2 | 20,14,7,2, | 90053 | 60.03533 | 1500 | 1 | 0 |
| Greedy | 20,14,7,2, | 90053 | 60.03533 | 1500 | | |
| Lookahead-1 | 20,16,7,2, | 92638 | 61.75867 | 1500 | | |
| Lookahead-2 | 20,16,7,2, | 98367 | 65.578 | 1500 | 1 | 0 |
| Greedy | 20,16,7,2, | 98367 | 65.578 | 1500 | | |
| Lookahead-1 | 9,8,6,5, | 62971 | 41.98067 | 1500 | | |
| Lookahead-2 | 9,8,6,5, | 62976 | 41.984 | 1500 | 1 | 0 |
| Greedy | 9,8,6,5, | 63569 | 42.37933 | 1500 | | |
| Lookahead-1 | 31,30,15,14, | 200131 | 133.4207 | 1500 | | |
| Lookahead-2 | 31,30,15,14, | 200152 | 133.4347 | 1500 | 1 | 0 |
| Greedy | 31,30,15,14, | 200167 | 133.4447 | 1500 | | |
| Lookahead-1 | 99,63,40,7, | 436733 | 291.1553 | 1500 | | |
| Lookahead-2 | 99,63,40,7, | 467730 | 311.82 | 1500 | 1 | 0 |
| Greedy | 99,63,40,7, | 456010 | 304.0067 | 1500 | | |
| Lookahead-1 | 28,21,14,7, | 157360 | 104.9067 | 1500 | | |
| Lookahead-2 | 28,21,14,7, | 157360 | 104.9067 | 1500 | 1 | 0 |
| Greedy | 28,21,14,7, | 166838 | 111.2253 | 1500 | | |
| Lookahead-1 | 88,66,22,3, | 346990 | 231.3267 | 1500 | | |
| Lookahead-2 | 88,66,22,3, | 382190 | 254.7933 | 1500 | 1 | 0 |
| Greedy | 88,66,22,3, | 374720 | 249.8133 | 1500 | | |
| Lookahead-1 | 70,66,22,10, | 335312 | 223.5413 | 1500 | | |
| Lookahead-2 | 70,66,22,10, | 365018 | 243.3453 | 1500 | 1 | 0 |

Continued on next page

**Table 5.1 – continued from previous page**

| Strategy | Priorities | VoI Loss | Loss/Auction | Rounds | LA-1 opt. | LA-2 opt. |
|---|---|---|---|---|---|---|
| Greedy | 70,66,22,10, | 375568 | 250.3787 | 1500 | | |
| Lookahead-1 | 30,15,5,1, | 97190 | 64.79333 | 1500 | | |
| Lookahead-2 | 30,15,5,1, | 97210 | 64.80667 | 1500 | 1 | 0 |
| Greedy | 30,15,5,1, | 99381 | 66.254 | 1500 | | |
| Lookahead-1 | 16,15,5,1, | 73848 | 49.232 | 1500 | | |
| Lookahead-2 | 16,15,5,1, | 82011 | 54.674 | 1500 | 1 | 0 |
| Greedy | 16,15,5,1, | 82175 | 54.78333 | 1500 | | |
| Lookahead-1 | 16,15,5,3, | 81715 | 54.47667 | 1500 | | |
| Lookahead-2 | 16,15,5,3, | 87002 | 58.00133 | 1500 | 1 | 0 |
| Greedy | 16,15,5,3, | 89788 | 59.85867 | 1500 | | |
| Lookahead-1 | 88,81,9,3, | 349236 | 232.824 | 1500 | | |
| Lookahead-2 | 88,81,9,3, | 402730 | 268.4867 | 1500 | 1 | 0 |
| Greedy | 88,81,9,3, | 402730 | 268.4867 | 1500 | | |
| Lookahead-1 | 61,60,59,58, | 535262 | 356.8413 | 1500 | | |
| Lookahead-2 | 61,60,59,58, | 535262 | 356.8413 | 1500 | 1 | 0 |
| Greedy | 61,60,59,58, | 535262 | 356.8413 | 1500 | | |
| Lookahead-1 | 8,6,4,2, | 44960 | 29.97333 | 1500 | | |
| Lookahead-2 | 8,6,4,2, | 44960 | 29.97333 | 1500 | 1 | 0 |
| Greedy | 8,6,4,2, | 47668 | 31.77867 | 1500 | | |
| Lookahead-1 | 11,8,5,2, | 55131 | 36.754 | 1500 | | |
| Lookahead-2 | 11,8,5,2, | 57918 | 38.612 | 1500 | 1 | 0 |
| Greedy | 11,8,5,2, | 57918 | 38.612 | 1500 | | |
| Lookahead-1 | 8,7,5,2, | 49449 | 32.966 | 1500 | | |
| Lookahead-2 | 8,7,5,2, | 49130 | 32.75333 | 1500 | 0 | FALSE |
| Greedy | 8,7,5,2, | 49130 | 32.75333 | 1500 | | |
| Lookahead-1 | 12,7,3,2, | 51137 | 34.09133 | 1500 | | |

Continued on next page

**Table 5.1 – continued from previous page**

| Strategy | Priorities | VoI Loss | Loss/Auction | Rounds | LA-1 opt. | LA-2 opt. |
|---|---|---|---|---|---|---|
| Lookahead-2 | 12,7,3,2, | 50966 | 33.97733 | 1500 | 0 | FALSE |
| Greedy | 12,7,3,2, | 50966 | 33.97733 | 1500 | | |
| Lookahead-1 | 31,15,6,2, | 105169 | 70.11267 | 1500 | | |
| Lookahead-2 | 31,15,6,2, | 111761 | 74.50733 | 1500 | 1 | 0 |
| Greedy | 31,15,6,2, | 111761 | 74.50733 | 1500 | | |

## 5.2 Simulator in NS2

### 5.2.1 Framework

We have developed simulator using Network Simulator *NS2* in order to evaluate performance of the lookahead strategy and the Traveling Auction mechanism. The simulator simulates a 3000mx2000m flat grid. There are 44 nodes in total where 2, 3 or 4 nodes act as objects being tracked by the sensor nodes and one node acts as a sink that calculates the loss of VoI as the packets arrive. We have written a traffic generator that is attached to the sensing nodes and generates customized UDP packets. These packets have an extra header, containing mission specific information such as the total number of missions, the associated mission id, *tmpi* of the missions and the mission priority. Whenever the target enters the sensing range of a node, the node generates target measurement packets for the mission associated with the target.

The simulator is configured with *TwoRayGround*. We are using TDMA as a medium access protocol and DSDV as routing protocol since all of our nodes are stationary except for the nodes acting as target objects. Such a configuration allows us to use simple routing algorithm such as DSDV. Moreover both the TDMA and DSDV protocols are already implemented in the standard distribution of *NS2* so we use them right away from the distribution.

We have implemented Fifo, Priority Queue, static auction with greedy election strategy, static auction with LA-1 election strategy, static auction with LA-2 election strategy and traveling auction as queue management mechanisms and we compared the performance of these mechanisms in terms of VoI loss incurred by missions as described in the upcoming sections.

In order to generate congestion, we change the bandwidth and packet generation rate according to the number of missions we have. This way we can produce congestion on nodes which is essential part for testing our approaches.

### 5.2.2 NS2 Simulator Results for static Auctions

We have simulated three different mobility models in which the target objects or the mobile objects move using "Random Waypoint model", "Pursue Model" [10] and "Manhattan Grid Model" [11]. We generated the mobility traces using BonnMotion mobility scenario generation tool [12]. We have experimented with these mobility models over the diverse set of priorities and with number of missions varying from 2 to 4. The sensor nodes send packets to the sink according to the object they have tracked in their vicinity. The packet generation rate at the sensor node is set such that congestion occur at the relay nodes. Relay nodes are the nodes which are on the shortest path to the sink from the more than one sensing node. These congested nodes then run auction to find the winner of the arriving transmission slot. The winner is selected in three different ways: Greedy, Lookahead-1 (LA-1) and Lookahead-1 (LA-2). The simulation is run for certain duration and the VoI loss is recorded as sum of VoI losses for all missions at the sink.

The following table presents results obtained by conducting various experiments.

**Table 5.2: Loss under competitive winner election strategies in Auctions**

| **Strategy** | **Priorities** | **VoI Loss** | **Duration(sec)** | **Mobility Model** |
|---|---|---|---|---|
| LA | 9,36,81 | 2563.721198 | 750 | Pursue |
| LA2 | 9,36,81 | 2602.389831 | 750 | Pursue |
| greedy | 9,36,81 | 2585.852908 | 750 | Pursue |
| LA | 1,4,9 | 284.857911 | 750 | Pursue |
| LA2 | 1,4,9 | 289.154426 | 750 | Pursue |

Continued on next page

**Table 5.2 – continued from previous page**

| Strategy | Priorities | VoI Loss | Duration(sec) | Mobility Model |
|---|---|---|---|---|
| greedy | 1,4,9 | 287.31699 | 750 | Pursue |
| LA | 1,5,10 | 634.182774 | 750 | Pursue |
| LA2 | 1,5,10 | 670.051451 | 750 | Pursue |
| greedy | 1,5,10 | 635.427721 | 750 | Pursue |
| LA | 3,6,9 | 380.509665 | 750 | Pursue |
| LA2 | 3,6,9 | 402.03 | 750 | Pursue |
| greedy | 3,6,9 | 381.256633 | 750 | Pursue |
| LA | 4,16,36 | 1139.431643 | 750 | Pursue |
| LA2 | 4,16,36 | 1156.617703 | 750 | Pursue |
| greedy | 4,16,36 | 1149.267959 | 750 | Pursue |
| LA | 4,16,36 | 9136.266792 | 750 | Waypoint |
| LA2 | 4,16,36 | 9145.828479 | 750 | Waypoint |
| greedy | 4,16,36 | 9155.041398 | 750 | Waypoint |
| LA | 1,5,10 | 3971.892849 | 750 | Waypoint |
| LA2 | 1,5,10 | 3979.656846 | 750 | Waypoint |
| greedy | 1,5,10 | 3971.111164 | 750 | Waypoint |
| LA | 4,16,36 | 9873.159463 | 1200 | Waypoint |
| LA2 | 4,16,36 | 9886.621457 | 1200 | Waypoint |
| greedy | 4,16,36 | 9893.122308 | 1200 | Waypoint |
| LA | 9,36,81 | 20556.60028 | 750 | Waypoint |
| LA2 | 9,36,81 | 20578.11408 | 750 | Waypoint |
| greedy | 9,36,81 | 20598.84315 | 750 | Waypoint |
| LA | 1,4,9 | 2284.066698 | 750 | Waypoint |
| LA2 | 1,4,9 | 2286.45712 | 750 | Waypoint |
| greedy | 1,4,9 | 2288.76035 | 750 | Waypoint |
| LA | 9,36,81 | 11365.67186 | 750 | Manhattan Grid |

Continued on next page

**Table 5.2 – continued from previous page**

| Strategy | Priorities | VoI Loss | Duration(sec) | Mobility Model |
|---|---|---|---|---|
| LA2 | 9,36,81 | 11374.2962 | 750 | Manhattan Grid |
| greedy | 9,36,81 | 11358.8166 | 750 | Manhattan Grid |
| LA | 1,4,9 | 1262.852429 | 750 | Manhattan Grid |
| LA2 | 1,4,9 | 1264.159297 | 750 | Manhattan Grid |
| greedy | 1,4,9 | 1262.090734 | 750 | Manhattan Grid |
| LA | 4,16,36 | 5051.409717 | 750 | Manhattan Grid |
| LA2 | 4,16,36 | 5056.637189 | 750 | Manhattan Grid |
| greedy | 4,16,36 | 5048.362934 | 750 | Manhattan Grid |
| LA | 1,5,10 | 3264.071491 | 750 | Manhattan Grid |
| LA2 | 1,5,10 | 3265.300126 | 750 | Manhattan Grid |
| greedy | 1,5,10 | 3272.627063 | 750 | Manhattan Grid |
| LA | 4,8 | 224.991711 | 750 | Pursue |
| LA2 | 4,8 | 224.726048 | 750 | Pursue |
| greedy | 4,8 | 217.946721 | 750 | Pursue |
| LA | 1,8 | 166.241522 | 750 | Pursue |
| LA2 | 1,8 | 165.423075 | 750 | Pursue |
| greedy | 1,8 | 158.796589 | 750 | Pursue |
| LA | 9,36 | 836.283134 | 750 | Pursue |
| LA2 | 9,36 | 817.292664 | 750 | Pursue |
| greedy | 9,36 | 805.827144 | 750 | Pursue |
| LA | 4,16 | 371.681393 | 750 | Pursue |
| LA2 | 4,16 | 363.241184 | 750 | Pursue |
| greedy | 4,16 | 358.145397 | 750 | Pursue |
| LA | 4,8 | 260.060211 | 750 | Waypoint |
| LA2 | 4,8 | 258.707115 | 750 | Waypoint |
| greedy | 4,8 | 258.555183 | 750 | Waypoint |

Continued on next page

**Table 5.2 – continued from previous page**

| Strategy | Priorities | VoI Loss | Duration(sec) | Mobility Model |
|---|---|---|---|---|
| LA | 1,8 | 198.357459 | 750 | Waypoint |
| LA2 | 1,8 | 194.332671 | 750 | Waypoint |
| greedy | 1,8 | 192.84002 | 750 | Waypoint |
| LA | 9,36 | 981.078641 | 750 | Waypoint |
| LA2 | 9,36 | 960.100798 | 750 | Waypoint |
| greedy | 9,36 | 977.269474 | 750 | Waypoint |
| LA | 4,16 | 436.034951 | 750 | Waypoint |
| LA2 | 4,16 | 426.711466 | 750 | Waypoint |
| greedy | 4,16 | 434.341988 | 750 | Waypoint |
| LA | 4,8 | 2348.20411 | 750 | Manhattan Grid |
| LA2 | 4,8 | 2342.33621 | 750 | Manhattan Grid |
| greedy | 4,8 | 2329.436843 | 750 | Manhattan Grid |
| LA | 1,8 | 2028.499734 | 750 | Manhattan Grid |
| LA2 | 1,8 | 2048.093324 | 750 | Manhattan Grid |
| greedy | 1,8 | 2039.173524 | 750 | Manhattan Grid |
| LA | 9,36 | 9572.388829 | 750 | Manhattan Grid |
| LA2 | 9,36 | 9628.134329 | 750 | Manhattan Grid |
| greedy | 9,36 | 9618.635904 | 750 | Manhattan Grid |
| LA | 4,16 | 4254.395035 | 750 | Manhattan Grid |
| LA2 | 4,16 | 4279.170813 | 750 | Manhattan Grid |
| greedy | 4,16 | 4274.949291 | 750 | Manhattan Grid |
| LA | 4,8,12,16 | 197.130644 | 220 | Pursue |
| LA2 | 4,8,12,16 | 195.498415 | 220 | Pursue |
| greedy | 4,8,12,16 | 198.337633 | 220 | Pursue |
| LA | 5,10,15,20 | 203.692481 | 200 | Pursue |
| LA2 | 5,10,15,20 | 202.710434 | 200 | Pursue |

Continued on next page

**Table 5.2 – continued from previous page**

| Strategy | Priorities | VoI Loss | Duration(sec) | Mobility Model |
|---|---|---|---|---|
| greedy | 5,10,15,20 | 204.805409 | 200 | Pursue |
| LA | 4,16,36,64 | 584.921859 | 220 | Pursue |
| LA2 | 4,16,36,64 | 536.72465 | 220 | Pursue |
| greedy | 4,16,36,64 | 567.126339 | 220 | Pursue |
| LA | 3,6,9,12 | 122.224654 | 200 | Pursue |
| LA2 | 3,6,9,12 | 121.618684 | 200 | Pursue |
| greedy | 3,6,9,12 | 122.858298 | 200 | Pursue |
| LA | 4,8,12,16 | 4116.26 | 750 | Waypoint |
| LA2 | 4,8,12,16 | 4138.789139 | 750 | Waypoint |
| greedy | 4,8,12,16 | 4107.244 | 750 | Waypoint |
| LA | 5,10,15,20 | 5145.328583 | 750 | Waypoint |
| LA2 | 5,10,15,20 | 5166.748176 | 750 | Waypoint |
| greedy | 5,10,15,20 | 5134.055916 | 750 | Waypoint |
| LA | 4,16,36,64 | 8450.463831 | 850 | Waypoint |
| LA2 | 4,16,36,64 | 8407.033123 | 850 | Waypoint |
| greedy | 4,16,36,64 | 8432.020861 | 850 | Waypoint |
| LA | 3,6,9,12 | 3186.406013 | 850 | Waypoint |
| LA2 | 3,6,9,12 | 3198.993247 | 850 | Waypoint |
| greedy | 3,6,9,12 | 3172.572364 | 850 | Waypoint |
| LA | 4,8,12,16 | 13636.48771 | 750 | Manhattan Grid |
| LA2 | 4,8,12,16 | 13686.58795 | 750 | Manhattan Grid |
| greedy | 4,8,12,16 | 13654.6154 | 750 | Manhattan Grid |
| LA | 5,10,15,20 | 17045.60963 | 750 | Manhattan Grid |
| LA2 | 5,10,15,20 | 17082.95408 | 750 | Manhattan Grid |
| greedy | 5,10,15,20 | 17068.26925 | 750 | Manhattan Grid |
| LA | 4,16,36,64 | 42370.4252 | 750 | Manhattan Grid |

Continued on next page

**Table 5.2 – continued from previous page**

| Strategy | Priorities | VoI Loss | Duration(sec) | Mobility Model |
|---|---|---|---|---|
| LA2 | 4,16,36,64 | 42359.74865 | 750 | Manhattan Grid |
| greedy | 4,16,36,64 | 42339.53773 | 750 | Manhattan Grid |
| LA | 3,6,9,12 | 10227.61966 | 750 | Manhattan Grid |
| LA2 | 3,6,9,12 | 10248.74668 | 750 | Manhattan Grid |
| greedy | 3,6,9,12 | 10240.96155 | 750 | Manhattan Grid |

**Table 5.3: Average VoI Loss Comparison**

| #Missions | LA-1 | LA-2 | Greedy |
|---|---|---|---|
| 2 | 1806.518 | 1809.022 | 1955.192863 |
| 3 | 5126.335 | 5139.808 | 5134.849851 |
| 4 | 8773.881 | 8778.846 | 8770.200395 |

Table 5.3 presents results obtained by conducting various experiments and averaging the losses over all the mobility models for 2, 3 and 4 missions. Table 5.4 presents the average VoI loss for mission under different mobility models followed by target objects.

**Table 5.4: Average VoI loss for mission under different mobility models**

| #Missions | LA-1 | LA-2 | Greedy | Mobility |
|---|---|---|---|---|
| 2 | 399.7994 | 392.6707 | 385.1789628 | Pursue |
| 2 | 468.8828 | 459.963 | 465.7516663 | Waypoint |
| 2 | 4550.872 | 4574.434 | 4565.548891 | Manhattan Grid |
| 3 | 1000.541 | 1112.548 | 1007.824442 | Pursue |
| 3 | 9164.397 | 9175.336 | 9181.375673 | Waypoint |
| 3 | 5236.001 | 5240.098 | 5235.474333 | Manhattan Grid |
| 4 | 276.9924 | 264.138 | 273.2819198 | Pursue |
| 4 | 5224.615 | 5227.891 | 5211.473285 | Waypoint |
| 4 | 20820.04 | 20844.51 | 20825.84598 | Manhattan Grid |

### 5.2.3 NS2 Simulator Results for Traveling Auctions

We compared static auctions with lookahead strategy and Traveling auctions with greedy strategy; the results are as follows.

**Table 5.5: Traveling Auctions vs Static Lookahead Auctions**

| Strategy | Priorities | VoI Loss | Duration(sec) |
|---|---|---|---|
| LA | 9,36,81 | 1254.07096 | 250 |
| LA2 | 9,36,81 | 394.160484 | 250 |
| greedy | 9,36,81 | 1261.802039 | 250 |
| Trav.LA | 9,36,81 | 418.945531 | 250 |
| Trav.LA2 | 9,36,81 | 373.926376 | 250 |
| Trav. Auct.greedy | 9,36,81 | 372.238146 | 250 |
| LA | 1,5,10 | 127144.6025 | 400 |
| Trav. Auct.greedy | 1,5,10 | 98905.49988 | 400 |
| LA | 4,8 | 53991.92132 | 400 |
| Trav. Auct.greedy | 4,8 | 41161.54354 | 400 |
| LA | 1,8 | 34210.66864 | 400 |
| Trav. Auct.greedy | 1,8 | 29434.34227 | 400 |
| LA | 9,36 | 183619.8879 | 400 |
| Trav. Auct.greedy | 9,36 | 150045.3421 | 400 |
| LA | 4,16 | 81608.83906 | 400 |
| Trav. Auct.greedy | 4,16 | 66686.81873 | 400 |
| LA | 1,4,9 | 139350.6098 | 700 |
| Trav. Auct.greedy | 1,4,9 | 124523.8613 | 700 |
| LA | 4,16,36 | 224075.5136 | 400 |
| Trav.greedy | 4,16,36 | 182903.3009 | 400 |
| LA | 4,8,12,16 | 421.479132 | 400 |
| Trav.greedy | 4,8,12,16 | 232.985302 | 400 |
| LA | 5,10,15,20 | 526.848916 | 400 |
| Trav.greedy | 5,10,15,20 | 291.231628 | 400 |
| LA | 4,16,36,64 | 1355.131237 | 400 |

Continued on next page

**Table 5.5 – continued from previous page**

| Strategy | Priorities | VoI Loss | Duration(sec) |
|---|---|---|---|
| Trav.greedy | 4,16,36,64 | 753.177826 | 400 |
| LA | 3,6,9,12 | 315.981045 | 400 |
| Trav.greedy | 3,6,9,12 | 174.738977 | 400 |

We have evaluated the *Traveling Auctions* mechanism with "Pursue" mobility model and compared it with static lookahead-1 auction mechanism under the same mobility model. The results are show in table 5.6. We compare it with LA-1 model because LA-1 on average performs better than Greedy and LA-2, so if Traveling Auctions do better than LA-1 then it is better than other strategies too. We also conducted experiments using lookahead as winner selection strategy for Traveling auctions but we have seen that greedy strategy gives best results when used along with Traveling Auctions mechanism.

**Table 5.6: Average VoI Loss Comparison**

| #Missions | LA-1 Static Auctions | Traveling Auctions |
|---:|---:|---:|
| 2 | 88357.82923 | 71832.01 |
| 3 | 122956.1992 | 101676.2 |
| 4 | 654.8600825 | 363.0334 |

## 5.3 Discussion

We can see from the results presented that it is not obvious to say which strategy is the best in all situations. In fact none of the strategy is best for all possible sets of parameters. The results we obtained seem to be dependent on the chosen parameters that are mobility pattern, distribution of the priorities and the duration of the simulation.

The experimental results show (see java Simulation Results) that all the strategies tend to complete a winners cycle and this cycle repeats over the course of simulation. Greedy selection is good because it decides using currently available information and does not assume future events. Hence, greedy strategy is independent of the sequence of winners over the course of simulation. Therefore, no matter when the simulation is stopped, greedy decision is not affected. In contrast the $lookahead - m$ strategy takes decision based on the assumption that there will be at least $m$ auctions in the future. Due to this assumption, lookahead strategy selects such a winner for current auction which may be suboptimal in the current auction but be better than greedy in case there are at least $m$ future auctions. This eventually makes lookahead dependent on sequence of winners during the simulation. If the simulation stops at a point where the cycle is completed, lookahead is optimal, but if the simulation stops before the cycle is completed, than lookahead may do worse because it was assuming that there will be future auctions and the broken cycle costs more than in case of greedy. Therefore, if there are enough auctions and the simulation runs for long enough time, the lookahead will cost less than the greedy selection because in longer runs the lookahead completes enough winners cycles that they cover the cost incurred by the last broken cycle.

In our experiments we have explored lookahead up to two levels. We found that in ideal situations $lookahead - 1$ is enough and we achieve the best results with $lookahead - 1$ if we run long enough. The $lookahead - 2$ becomes unnecessary, as we can achieve the same results from $lookahead - 1$ by running longer. There is one case in which $lookahead - 2$ does slightly better than $lookahead - 1$ but that is the case where the last cycle is broken and the distribution of the priorities is such that previously completed cycles do not payoff for the broken cycle. It is important to note that as we increase the number of lookaheads we have more and more computations involved which may eventually slow down decision process. Conducting experiments under ideal circumstances was helpful as it gives simplified version of the strategies and we can see the winners cycle and behavior of the strategies under simplistic scenarios.

In our simulations with real-life situations, it is obvious from the results that the situation is more complicated. We have mobile objects moving with some mobility pattern and packets are not continuously generated but appears only when an object is tracked, so behavior is less predictable. We can see from the results that there were cases where one or the other strategy works the best. Overall $lookahead - 1$ gives better results most often, but in some cases $lookahead - 2$ gives better results by 0.5-1%. This means that by increasing number of lookaheads we can slightly improve the results. Yet, for the slight improvement of 0.5-1% we have to pay the computation cost for deeper lookahead. Due to the fact that more lookaheads are computationally expensive and improve results slightly, we can say that even in real-life scenarios $lookahead-1$ is overall a better strategy which is not computationally very expensive, yet gives better results. Moreover, the higher level of lookahead will have another drawback that it assumes that higher number of auctions will follow the current auction which may not be true. Also the more lookahead, more calculations will be based on predictions so less precise these predictions will be. Thus, theoretically and practically $lookahead - 1$ seems to be the most balanced approach.

## 5.4 Findings

Based on experimentation that we have done, we can deduce which strategy should be used given certain conditions. Based on the type of traffic expected in the network, mobility of the object and sparsity of the sensor network appropriate strategy should be used in order to efficiently allocate the network bandwidth and therefore reduce the loss of VoI.

In a network where sensor nodes either sense or receive information about all the mobile target in the network at regular intervals and the mobile objects are spending more time in the vicinity of the sensor network $lookahead - 1$ is the best choice. As we have shown by results that we achieved from the JAVA simulator with aforementioned assumptions, the $lookahead-1$ performs better than $lookahead-2$ and is at least as good as *greedy* approach. In such situation the uncertainty is low, which means we have a predictable future events, thus the predicted future bids have higher chance of being accurate at least to depth 1. Consequently, $lookahead-1$ is an efficient strategy to use in such scenario. Luštrek, M. and Bulitko, V have also found 1 as the best fixed lookahead depth for the real time path finding problems [13].

When the sensor network has traffic that is poorly predictable, then a mixed strategy of $lookahead - 1$ and *greedy* is a better choice. This is the case where sensor nodes may not receive or generate packets for all mobile nodes at regular intervals; the packet generation is rather dependent upon the mobility pattern of the target object. Moreover, if the object is not very aggressive in changing its speed and direction, then this strategy will also more yield better results. This model is also most suitable for events that are longer in duration and the burst length is longer, but still the packets received are dependent upon the object tracked. In this scenario, we have a medium level of uncertainty which is why sometimes greedy strategy is good and sometimes lookahead performs well. As we have discussed in the earlier section, there is trade off between the computational complexity of doing deeper lookahead and the gain we get from the deeper lookahead; we believe that $lookahead - 1$ is enough.

In a highly dynamic scenario where the mobile target is moving aggressively and the events are very short term generating bursts of large size but of short length, the uncertainty in the network increases. For example, an object passes in high speed in front of the camera or police car in hot pursuit of another car. In such scenario for optimal sharing of bandwidth, sensor nodes need to share information of auctions that are held locally among other nodes so that a better collective decision can be made. In such situations, the normal static auction discussed earlier is not efficient; a $TravelingAuctions$ strategy with $greedy$ winner election strategy must be used to minimize the loss of VoI. As we have shown in the result section, in such bursty traffic situations $TravelingAuctions$ on average do up to 18% better than the normal auctions, even if we use $lookahead-1$ or mix of greedy and lookahead as winner election strategy. Moreover, if the sensor network is sparse, then using $TravelingAuctions$ is beneficial, but if the network is dense, then the experiments show that it is better to use static auctions.

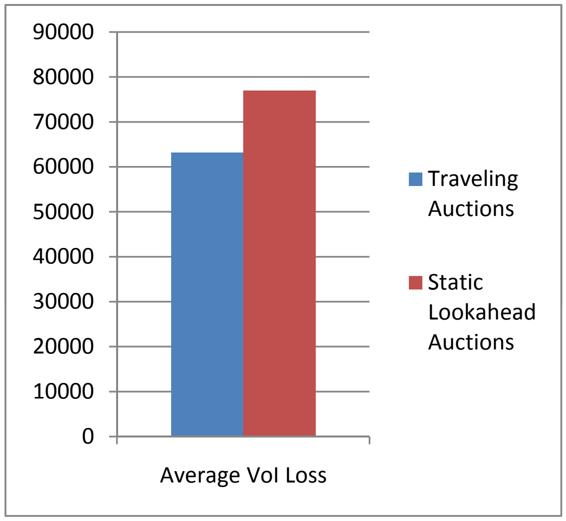



**Figure 5.1: Comparison of static Lookahead and Traveling Auctions**

The above chart shows that traveling auctions does up to 18% better than the static auctions in cases where we have sparse network and very spontaneous events. The values shown in the chart are averaged VoI losses over different priorities.

# CHAPTER 6

## Conclusion and Future Work

In this thesis we have investigated the use of Market Mechanisms in sensor networks. In particular, we have focused on use of *Auctions* for efficient resource sharing among multiple applications running on top of sensor networks. We have investigated the use of *lookahead* mechanism for winner selection in auctions and, thus, for congestion resolution in sensor networks. We have compared the efficiency of the traditional greedy winner selection mechanism with lookahead mechanisms. We have also proposed a new *Traveling Auctions* mechanism in which the state of the auction travels along with the congestion in the network in order to minimize the total VoI loss and thus maximize the social welfare for all the applications participating in the auctions. The results show that we can improve routing decisions using Traveling Auctions mechanism in sensor networks that are prone to high bursts of event driven data and intense congestion originating from such bursts of data.

There are multiple facets of the *Traveling Auctions*, *static auctions* and *lookahead* mechanism that can be investigated in more detail. One of them is introducing a preemptive mechanism for missions with lower bids. In such mechanism, mission packets with lower priority will be able to dynamically choose alternative path to the destination instead of competing with high priority missions on congestion nodes for shortest path. One can also investigate a dynamic lookahead mechanism for congestion resolution in sensor networks using auctions. In such a mechanism the node will dynamically calculate an optimal lookahead depth and use greedy or lookahead mechanism with calculated depth, in order to elect winners of the auctions.

# BIBLIOGRAPHY


[1] J.K. MacKie-Mason and H.R. Varian. Pricing congestible network resources. *Selected Areas in Communications, IEEE Journal on*, 13(7):1141–1149, 1995.

[2] J. Shu and P. Varaiya. Pricing network services. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1221–1230. IEEE, 2003.

[3] C.T. Ee and R. Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 148–161. ACM, 2004.

[4] K. Karenos, V. Kalogeraki, and S.V. Krishnamurthy. A rate control framework for supporting multiple classes of traffic in sensor networks. 2005.

[5] J. Ibbotson, C. Gibson, J. Wright, P. Waggett, P. Zerfos, B. Szymanski, and D.J. Thornley. Sensors as a service oriented architecture: Middleware for sensor networks. In *Intelligent Environments (IE), 2010 Sixth International Conference on*, pages 209–214. IEEE, 2010.

[6] L. Chen, B.K. Szymanski, and J.W. Branch. Auction-based congestion management for target tracking in wireless sensor networks. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–10. IEEE, 2009.

[7] 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*, 53(5):513, 2010.

[8] S.C. Geyik, B.K. Szymanski, P. Zerfos, and D. Verma. Dynamic composition of services in sensor networks. In *2010 IEEE International Conference on Services Computing*, pages 242–249. IEEE, 2010.

[9] B. Szymanski, S.Y. Shah, S.C. Geyik, S. Das, M. Chhabra, and P. Zerfos. Market mechanisms for value of information driven resource allocation in sensor networks. In *The Third International Workshop on Information Quality and Quality of Service for Pervasive Computing, IQ2S 2011*, pages 62–67. IEEE, 2011.

[10] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad-hoc network research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[11] A. Samukic. Umts universal mobile telecommunications system: Development of standards for the third generation. In *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, volume 4, pages 1976–1983. IEEE, 1998.

[12] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn. Bonnmotion: A mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.

[13] M. Luštrek and V. Bulitko. Lookahead pathology in real-time path-finding. In *Proceedings of the National Conference on Artificial Intelligence (AAAI), Workshop on Learning For Search*, pages 108–114, 2006.