Implementation of Smart Elevator System based on Wireless Multi-hop AdHoc Sensor Networks

Hamza Ijaz Abbasi, Abdul Jabbar Siddiqui
Computer Engineering Department
King Fahd University of Petroleum & Minerals
{ hamza90ksa , dynamic.aj } @gmail.com

Abstract— **In modern day world, time has become a precious resource. Therefore, different strategies and techniques are constantly being employed in all fields of life to save every bit of time. Increasingly many of such applications involve wireless sensor networks. A highly potential system that can be made significantly more efficient using WSNs is an elevator system. There have been numerous attempts to improve the serving efficiency of the elevator system over the course of time. This paper proposes to utilize the elevator system in a more productive manner so that more number of people can be served in a lesser time. Hence, people will be able to spend this valuable time on other important and crucial matters rather than waiting for the elevator and wasting their time in vain. To achieve this goal, we implement the elevator system based on a wireless adhoc network of intelligent floors (equipped with sensors) which can communicate with each other in a multi-hop fashion. In this way, every floor is aware of the traffic conditions, i.e., the number of upward/downward requesting-passengers, waiting at every other floor and the elevator positions in real-time and hence can make efficient decisions of where to direct/stop the elevator.**

Keywords- wireless adhoc networks; mobile computing; smart systems; elevator system; wireless sensor networks

I. INTRODUCTION

The implementations of many smart systems are increasingly depending upon wireless adhoc sensor networks. Lack of need for centralized and dedicated infrastructure in wireless adhoc sensor networks and comparatively lower power consumption of the wireless sensor nodes make it more ample to be applied in many situations [1]. In this paper, an implementation of a smart elevator system based on a wireless multi-hop adhoc sensor network is demonstrated. For experimental simplicity, a single elevator in a five-storey building is considered. Each floor is equipped with a sensor and the traffic conditions of every floor are communicated to all other floors in a direct or multi-hop fashion. In the proposed system, the elevator serves the floor with greater traffic and hence achieving the objective of serving more people within a given time. The most striking feature of our proposed system is that of avoiding unnecessary stops at floors by sensing the departure of the requesters before the elevator reached there.

Wireless adhoc sensor networks are those that comprise of entities (sensor nodes) that can connect, communicate and coordinate with each other with neither a centralized server nor any pre-existing infrastructure [1][6]. Such Wireless sensor networks have been successfully deployed and have numerous applications in several fields such as medicine [7], military [8], environment monitoring [9] etc. Another application of wireless sensor networks is our elevator system. As our proposed elevator system communicates in a multi-hop manner, this means that every node (floor) can only communicate with its adjacent nodes (floors) only. All nodes are capable of being informed of each other's traffic through this technique of wireless multi-hop communication [2]. We propose this multi-hop nature of interacting between sensors keeping in mind the inter-floor distance and the floor thickness that could affect communication if each floor was to be connected to every other node. This is explained in greater detail in Section–2.

To integrate our WSN in controlling the elevator, a study on traditional elevator systems was made. The two basic algorithms that make the decision to stop the elevator car at a particular floor, called the dispatching algorithms, are (a) Based on the current direction of the elevator, and (b) Based on the time of request from each floor [3]. These revolve around the objectives of either minimizing the consumption of power or minimizing the average waiting time for passengers [4]. In the first approach, if the elevator is currently moving in a certain direction, it will stop at floors in its way that have requests in that direction only and will change direction once it serves them and if there are requests in the opposite direction. On the other hand, in the time-based approach, the requests are stored in a queue and ordered according to their time of arrival [3]. The elevator then would serve the floors according to this queue.

Though there are quite a few elevator dispatching algorithms being commonly implemented and used, like those above, there are several problems concerning the passengers and requesters. Firstly, considering the direction-based approach, if the elevator is moving in a particular direction (up/down), and the number of people requesting to go in that direction is less than those requesting to go the opposite way, a

978-1-4673-0498-6/11/$26.00 ©2011 IEEE

majority will be kept waiting. Secondly, in the time-based approach, if the queue of requests is haphazard, the elevator would waste a lot of power and trips. The elevator would be moving up and down the building and passing other requesting floors without serving them. Eventually it will return to them at some later point in time. So, in such cases, the elevator would be consuming more power and time than it would have to if it served the intermediate floors when it passed across them previously. In addition, the dispatching algorithms like those above and many more improvised ones fail to consider the cases when the passengers push the request button and leave the elevator-area due to some reason. In such cases where the elevator will have no mechanism to be aware of this update, it will have wasted trips to such floors where the requesters have left. We present a solution to such problems through the use of real-time wireless sensor networks connected together in a multi-hop ad-hoc manner.

This paper first gives a brief description of the proposed elevator system (Section-2), including the network design of the system and the communication protocols. Following this, the information awareness mechanism (Section-3) and inter-communication procedures (Section-4) of the elevator system have been presented. Then our main elevator dispatching algorithm (Section-5) has been explained along with some test cases and their analysis in Section-6. After this, the performance evaluation of the system is discussed in Section-7. Future recommendations to improve the system have also been laid forward in Section-8. Towards the end, the paper is concluded in Section-9 followed by acknowledgements and references.

II. DESCRIPTION OF THE PROPOSED SYSTEM

The proposed smart elevator system utilizes a wireless multi-hop ad-hoc sensor network and will be simulated and prototyped depicting a building comprising of five floors. We emulate each floor using a single laptop to ease simulation and implement a graphical view of the system at work. Therefore, five laptops will be connected together in an ad hoc multi-hop manner such that the nth laptop (emulating the nth floor) will be connected only to the (n-1)th laptop and (n+1)th laptop with the exception of 1st and 5th laptops respectively where 1st laptop will only be connected to 2nd laptop whereas 5th laptop will be only connected to 4th laptop.

The proposed idea is to have two separate areas (boxes) at every floor's elevator lobby/area: (1) for the people wanting to go up, and (2) for the people wanting to go down – see Figure-1. Each laptop could be equipped with some external sensor/camera connected to them which could then feed a people counting algorithm (for example, based on Image Processing) and know the number of people waiting in that floor to go upwards and downwards. The laptop will then communicate and pass this information to the next connected laptops (multi-hop). In this way, the information will be shared among all the laptops. However, the final decision will only be made by that laptop (i.e., floor) where the elevator is currently present. This decision will only be made by this laptop after running some algorithm as describer in Section-5.

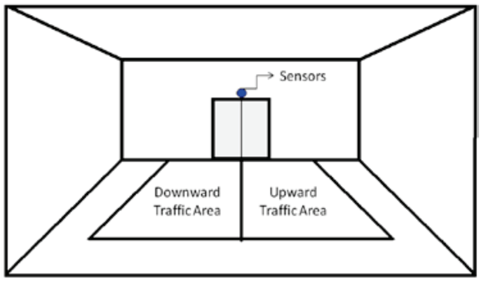


Figure 1. View of the elevator-lobby with upward/downward traffic areas

A. The Network Design

In our implementation of the elevator system, each floor is represented and simulated by a laptop with the Java based application to run the simulations. The floors (or laptops) are connected with each other in a multi-hop adhoc fashion such that each can communicate only with its immediate neighbouring laptop (the floor above, and the floor below). For example, Floor3 can communicate with Floor1 and Floor2 only. Each floor (laptop) in our system is assigned a unique IP Address of the form "192.168.1.x" where 'x' corresponds to the respective floor-number. Eg: Floor#1 has the IP Address "192.168.1.1".

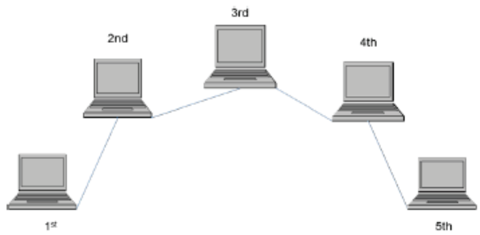


Figure 2. Five laptops (floors) connected in multi-hop fashion

B. Communication Protocols

In order to setup a wireless ad-hoc network amongst the laptops (floors), we use the standard IEEE 802.11 b/g wireless Ethernet cards on each laptop (which is simulating a floor).

For the inter-laptop (inter-floor) communication, the datagram communication protocol we have used in our system is known as UDP (User Datagram Protocol) which is a connectionless protocol, as opposed to the TCP (Transfer Control Protocol) which is a connection-oriented protocol. [5]

UDP serves our purpose best because of its simplicity and owing to the following reasons. Firstly, our system does not require connections between floors to be continuously established as the communication between floors is required only in few circumstances like change in traffic or elevator position. Secondly, the information to be transferred is simply packets of a maximum of 4 bytes and not streams of data. Hence, UDP is more efficient for our system than TCP as it emphasizes on greater speeds and less delays. [5]

Moreover, the sensor nodes only need to send/receive information intermittently (i.e., only whenever traffic conditions fluctuate) and will stay idle when there's no need to

978-1-4673-0498-6/11/$26.00 ©2011 IEEE

communicate. So, maintaining a connection between them would be a waste of power. Hence we exploit the connection-less nature of UDP [5]. On the other hand, using TCP will maintain the connection (i.e. keep the sensor active) for a relatively longer period of time, thus causing power wastage as opposed to the UDP.

III. INFORMATION AWARENESS MECHANISM

In our system, every floor is intelligently aware of the traffic conditions and elevator positions at all other floors apart from the capability to transferring (transporting) the elevator towards the most efficient direction (see Section-5). To implement this feature, we have devised the following components:

A. *Floor Information Packets (FIPs)*

These are UDP datagram packets having information such as the floor-number which it belongs to, the amount of upward-traffic, the amount of downward traffic, and an indicator of the elevator presence at that floor (yes/no). The structure of the packet is shown in Figure-3 below.

Byte0	Byte1	Byte2	Byte3
Floor ID	Up Traffic	Down Traffic	Elevator Presence

Figure 3. Structure of the Floor Information Packet (FIP)

- *FloorID:* This corresponds to the number of the respective floor to which the packet information belongs. Eg: 1,2,…5.
- *Up-Traffic:* The number of people at the corresponding floor who wish to go upwards
- *Down-Traffic:* The number of people at the corresponding floor who wish to go downwards.
- *Elevator-Presence:* Containing a simple ‘y’ or ‘n’ saying whether or not the corresponding floor has the elevator at it currently.

B. *Elevator Information Packet (EIP)*

The EIP is a UDP packet simulating the elevator car. Its structure and comprising fields are shown below:

Byte0	Byte1	Byte2	Byte3
Packet-ID	SRC	DST	Direction

Figure 4. Structure of the Elevator Information Packet (FIP)

- *Packet-ID:* To distinguish between FIPs and EIP. While the FIPs will have their Packet-IDs representing their corresponding floor-numbers, EIP will have this field value as ‘0’ to make it convenient for the running codes to identify it as an elevator packet.
- *SRC:* The floor number which has just sent this elevator packet
- *DST:* The floor numer which this elevator is supposed to serve, based on the decision made by the SRCth floor.
- *Direction:* Indicating the direction in which the elevator packet is supposed to go based on the decision-process of the SRCth floor. It is either ‘u’ (for upwards) or ‘d’ (for downwards).

IV. THE INTER-COMMUNICATION PROCEDURE

We have developed and implemented algorithms to send/receive the FIPs and the EIP according to the procedure described below.

A. *Updating other Floors*

Depending on the input by the sensor/camera (or from the user in our simulations) about the traffic at a certain floor, its traffic/no-traffic info is sent as a packet (FIP) to the floor above it and the floor below it (i.e., its immediate neighbors only). A sending thread is created whenever there is change (increase/decrease) in traffic

B. *Listening to other Floors*

To be aware of the traffic conditions of other floors, two threads of codes run simultaneously on each floor (laptop/sensor): (1) To receive FIPs/EIP from the floor(s) above, and (2) To receive FIPs/EIP from the floor(s) below. Separate threads will take care of avoiding possible collisions of simultaneously arriving FIPs or EIP.

C. *Conveying floors' traffic information*

In our design, as soon as a floor receives the FIP(s) from its immediate neighbors, it passes it on immediately upwards (if FIP from below) or downwards (if FIP from above). See Fig-5.

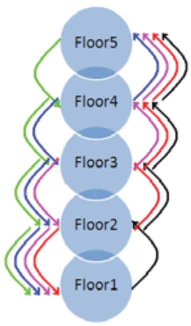


Figure 5. Multi-hop FIPs transfer

D. *Assignment of Sockets*

In order to have convenience in experimenting and testing, we have designed a convention to be used when assigning sockets to respective floor's (laptop's/sensor's) Java threads. The socket numbers used belong to the free-to-use ranges.

- *For FIPs:* We have designed our system such that every floor receives FIP(s) on socket-port 9878 (from floors above) and on socket-port 9877 (from floors below). Exceptions would be the topmost and the bottom-most floors which would only have socket-port 9877 (receives only from below) and 9878 (receives only from above), respectively.

978-1-4673-0498-6/11/$26.00 ©2011 IEEE

- *For EIP:* A dedicated socket-port (9780) is created at each floor to send and receive the elevator packet (EIP)

In this manner, we succeeded in designing and implementing the feature of having each floor's information reach all other floors in a multi-hop fashion.

V. ELEVATOR DISPATCHING ALGORITHM

Although our basic idea of integrating a WSN to an elevator system can be coupled with any elevator dispatching techniques, we implemented the elevator system based on developing an approach to serving the denser areas first, i.e. serving the floors with greater traffic intelligently sensed, which is a quite unique idea and very different from traditional elevator systems. Thus, we had to come up with our own design of the system and redefine the basic elevator decision algorithm. The following points summarize and highlight the most basic and fundamental components of our elevator dispatching algorithm.

Initially (During the startup of the elevator system program), elevator is always present at Floor-1. The floors intermittently communicate with each other in order to keep each other updated about the latest traffic information as well as the current location of the elevator. Each floor (sensor) that receives the Elevator (EIP) is entitled to take the decision about where to send it next based on the following criteria:

- Let n be the floor which currently has the elevator. If the sum of the traffic (number of people) requesting from above floors and the up traffic (people requesting to go up) in the n^{th} floor is greater than the sum of the traffic (number of people) requesting from down floors and the down traffic (people requesting to go up) in the n^{th} floor, the result will be that the elevator will be sent to the above floor i.e. to Floor (n+1).
- However, If the sum of the traffic (number of people) requesting from above floors and the up traffic (people requesting to go up) in the n^{th} floor is lesser than the sum of the traffic (number of people) requesting from down floors and the down traffic (people requesting to go up) in the n^{th} floor, then the elevator will be sent downwards to Floor(n-1).
- In the same way, the new floor which receives the elevator will then make the decision.
- If the traffic above equals the traffic below, the elevator will continue to move in the same direction that it was previously travelling in, according to pre-defined default elevator dispatching algorithm(s).
- If the total traffic in all floors is zero (i.e. there are no requests), the elevator will eventually return back to Floor 1 (the default stop).

After the elevator serves any floor, the traffic in that floor is updated to '0' and communicated across other floors. The stopping case for the elevator is Floor-1. In other words, after serving all the requests, the elevator will always return back to Floor-1.

If there are any new requests or changes in traffic conditions at any floor, the system would be capable of accommodating these changes dynamically and would responding to them as per the algorithm. For example: (1) If there is increase/decrease in number of people waiting at a floor, then change the priority of serving based on the new traffic conditions; (2) If the people at a floor requested the elevator but left before it arrived, cancel stopping at that floor so as to save a considerable amount of time of other passengers.

The Figure-6 explains the concept with greater detail, considering the example of a five floor building with a single elevator. As depicted in the figure below, since traffic above Floor_3 + up-traffic at Floor_3 (i.e., 0+6=6) is greater than traffic below Floor_3 + down-traffic at Floor_3 (i.e., 3+2=5), the elevator is sent upwards to Floor_4 (to be sent to Floor_5).

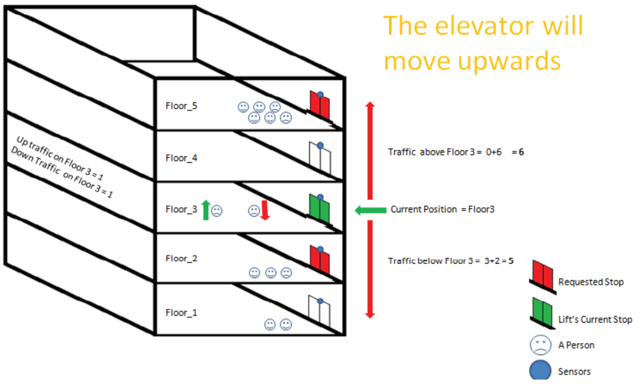


Figure 6. Proposed Elevator Dispatching Algorithm

VI. TESTING AND ANALYSIS

To demonstrate the working of our proposed wireless multi-hop ad-hoc sensor network based elevator system, a Java based Graphical User Interface (GUI) was designed and implemented on the laptops representing the floors/sensors. Based on these, several combinations of traffic conditions and changes were tested, of which we present the test-case as described in Figure-8.

A. The Graphical User Interface (GUI)

The GUI has been designed in such a way so as to receive input from the user (or a camera/sensor) about the up/down traffic of a particular floor, and to display the current status/position of the elevator, the current traffic at each floor, etc. A log of all received FIPs is also displayed at each floor which helps trace back the steps of the elevator, status of each floor during these steps, etc. during the analysis. Also, the number of upward and downward passengers waiting at each floor is shown in blue and red, respectively . The following snapshot (Figure-7) shows and describes the GUI that we have implemented to simulate our elevator system:

978-1-4673-0498-6/11/$26.00 ©2011 IEEE

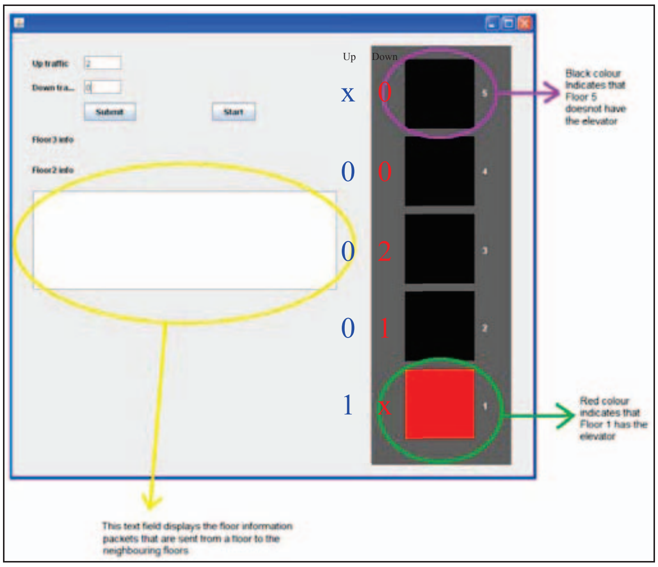
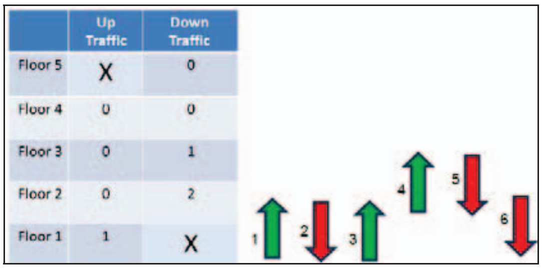


Figure 7. The GUI of our Smart Elevator System at one floor

B. A Sample Test-Case

We have tested our proposed smart elevator system considering a five floor building with a single elevator. With the traffic conditions depicted below in Figure-8, the proceedings points describe how the algorithm operates.



	Up Traffic	Down Traffic
Floor 5	X	0
Floor 4	0	0
Floor 3	0	1
Floor 2	0	2
Floor 1	1	X

Figure 8. A sample test-case for our smart elevator system

To begin with, as described in Section-V, the elevator is present at Floor-1 by default. The following steps describe how the different floors/people will be served and what exactly happens at each stage.

1) *At Floor-1:*
   Traffic of Above Floors= 3, Up-Traffic = 1, Traffic of Floors Below = 0 (no floor below), Down-Traffic = 0. Since (2+1 + 1) > (0 + 0), Elevator is sent upwards to Floor-2 and Floor 1 up-Traffic becomes 0 (Now served).

2) *At Floor-2:*
   Traffic of Above Floors= 1, Up-Traffic = 0, Traffic of Floors Below= 0, Down-Traffic =2. Since (1 + 0) < (0 + 2), Elevator is sent downwards to floor 1 and Floor 2 down-Traffic becomes 0 (Now served).

3) *At Floor-1:*
   Traffic of Above Floors= 1 (at Floor-3), Up-Traffic = 0, Traffic of Floors Below = 0, Down-Traffic =0, Since (1 + 0) > (0 + 0), Elevator is sent upwards to Floor-2 to be sent to Floor-3.

4) *At Floor-2:*
   Traffic of Above Floors= 1, Up-Traffic = 0, Traffic of Floors Below = 0, Down-Traffic =0, Since (1 + 0) > (0 + 0), Elevator is sent upwards to Floor-3.

5) *At Floor-3:*
   Traffic of Above Floors= 0, Up-Traffic = 0, Traffic of Floors Below= 0, Down-Traffic =1, As (0 + 0) < (0 + 1), Elevator is sent downwards to Floor-2

6) *At Floor-2:*
   Traffic of Above Floors= 0, Up Traffic = 0, Traffic of Floors Below= 0, Down Traffic =0, Since (0 + 0) = (0 + 0), i.e., no traffic at any floor, therefore the Elevator is sent downwards to Floor-1

Finally, as the elevator reaches Floor-1 and there are no requests from any floor, the elevator remains at Floor-1 and continuously waits for any new requests or changes in traffic conditions.

VII. PERFORMANCE EVALUATION AND RESULTS

To compare the efficiency and performance of our proposed system with that of an existing traditional one, we chose the elevator of one of our campus buildings. This elevator in consideration followed the algorithm of serving elevator calls/requests in one direction first and then those in the opposite direction. Based on several test scenarios, the 'waiting time' for each category of passengers was measured.

From observation, the elevator's transition time (i.e., the time taken by it to move from one floor to the other, T_{trans}) was found to be around 5 seconds. To account for the time taken when an elevator stops at any floor (stopping time, $T_{stopping}$), we considered only the doors' opening and closing time which was measured to be around 6 seconds. We ignored the loading/unloading time of passengers to avoid statistical complexities which is beyond the focus of this paper. So, in total, the waiting time for passengers is calculated based on equation (1).

$$(T_{waiting})_i = (N_{transitions})_i *(T_{trans}) + (N_{stops})_i *(T_{stopping}) \quad (1)$$

$$T_{stopping} = T_{door\text{-}opening} + T_{door\text{-}closing} \quad (2)$$

Where

$(T_{waiting})_i$ –waiting time for (upward or downward) passengers of a given floor 'i';

$(N_{transitions})_i$ –number of transitions made by the elevator to reach a particular floor 'i' and load the corresponding upward or downward passengers;

T_{trans} –time taken by the elevator to move from one floor to the other;

$(N_{stops})_i$ –number of stops made by the elevator before serving corresponding upward or downward passengers of a particular floor 'i';

978-1-4673-0498-6/11/$26.00 ©2011 IEEE

$T_{stopping}$ (6 sec.) –the time taken to open ($T_{door-opening}$= 3 sec.) and then close doors ($T_{door-closing}$ = 3 sec.) while serving a particular floor.

The performance was measured and compared based on the above parameters and equations by calculating the waiting times in various scenarios in both the proposed system and the existing system of the elevator considered at campus. To summarize our results, we found that the proposed system proved to be advantageous over the traditional one in several cases of traffic quantities at different floors in reducing waiting times by 33.3%–76.9% (reduction of around 10 to 50 seconds). However, there were quite a few cases in which, owing to the limitations and challenges of simulation (as discussed in Section-8), the traditional system provided lesser waiting times.

Moreover, in cases where the requesters left the elevator-lobby before it arrived, our proposed system proved to be significantly more efficient owing to the fact that this information of requesters leaving the lobby at a given floor was communicated across other floors and hence the elevator was not made to stop there and hence reducing time wastage due to unnecessary stops ($T_{stopping}$ = 6 sec. saved per reduced stop).

VIII. FUTURE RECOMMENDATIONS

In buildings where there is a group of elevators and not just a single one as we consider in this paper, the system will have to be extended to have a group of sensors at each floor. Consequently, more complex inter-floor traffic information awareness mechanism and elevator dispatching algorithms would have to be developed to achieve the main goal of serving more people in lesser/same time.

Furthermore, the smart elevator system should be equipped with some mechanism to keep track of people inside it at any given time, and also of people going in/out as there are certain situations when the proposed algorithm could face bottlenecks when it has reached its passenger limit.

Other issues such as those related to security of the wireless ad-hoc network of sensors/laptops, failure of nodes, information update delays in case of high-rise buildings, enabling emergency or manual interruptions and controls, etc. (which are beyond the scope of this paper) also need to be researched and analysed in order to come up with a completely robust and applicable solution of a smart elevator system.

Considering the results obtained by testing our proposed system and the above mentioned challenges, there is definitely a wide array of opportunities for research into the implementation of a smart elevator system based on wireless multi-hop ad-hoc sensor networks.

IX. CONCLUSION

A smart elevator system has been designed, implemented and tested utilizing wireless ad-hoc sensor nodes that are connected in a multi-hop fashion and sense passenger-traffic in real-time. Furthermore, an algorithm is developed to implement and prioritize the elevator system based on the traffic inflow such that all the floors communicate with each other and share their traffic information to optimize and control the movement/stopping of the elevator. To summarize, the elevator system is implemented in such a way that it serves those floors first which have the greater traffic. The experimental observations and comparisons of our proposed system against a traditional elevator proved the former to be more efficient in terms of reduced waiting times for passengers in various cases, especially those in which there's the problem of elevator-requesters at a floor leaving the elevator's lobby before it arrives. Our system is designed to eliminate such unnecessary stops and hence drastically saving others' time. Despite the positive results, the proposed system had some limitations in performance in several other cases, owing to reasons identified in Section-8. Tackling these issues and incorporating enhanced features would be in the next phase of the project.

ACKNOWLEDGMENT

The authors of this paper extend their sincere thanks and acknowledgement to Dr. Tarek R. Sheltami (Associate Prof., Computer Engineering Dept., KFUPM) for his dedicated support and guidance along with inspiring us throughout the research work and also for providing us the necessary resources to implement a testing prototype of the idea.

REFERENCES

[1] Ansari, I.S.; , "An Implementation of Traffic Light System Using Multi-hop Ad hoc Networks," Network-Based Information Systems, 2009. NBIS '09. International Conference on , vol., no., pp.177-181, 19-21 Aug. 2009.

[2] Chowdhury, M.U.; Perera, D.; Pham, T.; , "A performance comparison of three wireless multi hop ad-hoc network routing protocols when streaming MPEG4 traffic," Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International , vol., no., pp. 516- 521, 24-26 Dec. 2004 doi: 10.1109/INMIC.2004.1492933.

[3] Daniel M. Munoz; Carlos H. Llanos; Mauricio Ayala-Rincon; Rudi van Els; , "Implementation, Simulation and Validation of Dispatching Algorithms for Elevator Systems," Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference on , vol., no., pp.1-8, Sept. 2006.

[4] Jianchang Liu; Yiyang Liu; , "Ant Colony Algorithm and Fuzzy Neural Network-based Intelligent Dispatching Algorithm of An Elevator Group Control System," Control and Automation, 2007. ICCA 2007. IEEE International Conference on , vol., no., pp.2306-2310, May 30 2007-June 1 2007.

[5] Kurose, J. F.; Ross, K. W. (2010). Computer Networking: A Top-Down Approach (5th ed.). Boston, MA: Pearson Education.

[6] Vijaya, I.; Mishra, P.B.; Dash, A.R.; Rath, A.K.; , "Influence of Routing Protocols in Performance of Wireless Mobile Adhoc Network," Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on , vol., no., pp.340-344, 19-20 Feb. 2011.

[7] Furtado, H.; Trobec, R.; , "Applications of wireless sensors in medicine," MIPRO, 2011 Proceedings of the 34th International Convention , vol., no., pp.257-261, 23-27 May 2011.

[8] Zhi Sun, Pu Wang, Mehmet C. Vuran, Mznah A. Al-Rodhaan, Abdullah M. Al-Dhelaan, Ian F. Akyildiz, BorderSense: Border patrol through advanced wireless sensor networks, Ad Hoc Networks, Volume 9, Issue 3, May 2011, Pages 468-477, ISSN 1570-8705, 10.1016/j.adhoc.2010.09.008.

[9] Shibata, Y.; Sato, Y.; Ogasawara, N.; Chiba, G.; , "A Disaster Information System by Ballooned Wireless Adhoc Network," Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on , vol., no., pp.299-304, 16-19 March 2009.

978-1-4673-0498-6/11/$26.00 ©2011 IEEE

978-1-4673-0498-6/11/$26.00 ©2011 IEEE