Asian Journal of Information Technology 6 (3): 396-401, 2007
© Medwell Journals, 2007

Predictive Modeling for the Determination of Wireless Sensor Network Reliability

[1]M.C. Ndinechi, [1]J.I. Ejimanya and [2]C.M. Akujuobi
[1]Department of Electrical and Electronic Engineering, Federal University of Technology, Owerri, Nigeria
[2]Texas A and M University Prairie View, USA

Abstract: Energy conservation in wireless sensor nodes has posed a big problem in the applications and deployment of wireless sensor networks. This study proposes an algorithm to minimize energy consumption in the nodes so as to extend the life time of the sensors. This is achieved by generating a prediction model from the readings received from the sensor nodes in the sensed field for sometime. The readings are then used to generate a model at the base station and sent back to the sensor nodes. The nodes compare their readings with this model and transmit only when the difference is below or above a specified threshold. The software is developed using Visual Basic version 6.0. Experimental results show that the proposed algorithm considerably reduced the energy consumption at the sensor nodes. It achieves this by allowing the person monitoring the sensor field to set the sleep mode time for the sensor nodes. The test result shows the algorithm to increase the lifetime of the sensor nodes.

Key words: Predictive, conservation, wireless, sensor, consumption

INTRODUCTION

Of recent, increasing sophistication and developments in hardware and low level software have led to the deployment of small units of autonomous sensing nodes for performing various sensing tasks (Carle and Simplot, 2004; Goel and Imielinski, 2001). Where an aggregation of such nodes are thrown together to co-operate and perform a sensing task, communicate untethered in short distance as well as collaborate with one another using wireless communication in an asymmetric many-to-one data flow, they constitute what is known as Wireless Sensor Network (WSN) (Akyildiz, 2002). Co-operative sensing has very useful applications in military and even common day-to-day situations. Some of the applications include environmental monitoring, meteorological research, integrated patient monitoring in hospitals, Nuclear, Biological and Chemical (NBC) attack detection and reconnaissance and home automation etc. (Agre and Clare, 2000).

Co-operative sensing raises several interesting research issues. A majority of these are rooted in the fact that sensors are typically limited by their energy reserves, communication bandwidth, computational power and reliability of the readings. The most severe of these is the energy constraint (Estrin *et al*., 1999; Hill *et al*., 2000) because if the sensor runs out of battery, there can be neither communications nor computation. Another constraining factor is that the cost of managing the sensor nodes far exceeds the cost of the nodes themselves. For instance, the environment in which they are deployed could make human maintenance difficult. Also, it is expected that a reasonably sized network will comprise thousands of nodes. The large number of these nodes in a sensor network makes it difficult to manage. However, increasing hardware integration and economy of scale are driving the prices of sensor nodes down. All these make "use-and-throw" a very attractive option (Goel and Imielinski, 2001).

Given all these, the average lifetime of a sensor determines the cost of running a sensor network. Current technological trend shows that although the computational power in a sensor is expected to follow Moore's law, battery technology is only expected to improve by 2-3% per year (Srivastava, 2000; Rabaey, 2000). Therefore the only possible way one can increase the lifetime of a sensor is by making use of mechanisms that are highly energy-efficient. Every operation of a sensor node-receiving and transmitting data or computing consumes energy. The goal of being energy efficient therefore translates into the problem of optimizing the number of these operations that need to be performed (Heinzelman *et al*., 2000).

Corresponding Author: M.C. Ndinechi, Department of Electrical and Electronic Engineering, Federal University of Technology, Owerri, Nigeria

In this study, we propose mechanisms for performing monitoring in a wireless sensor network in an energy efficient way to increase reliability. In traditional sensing paradigm, a central server maintains a database of readings from all the sensors. Sensors update the server when their readings change. Monitoring operations is therefore supported by the server which maintains the current state of all the sensors involved in the operation. There are too many data transmitted in such a system making it very energy inefficient. Two key observations are necessary to significantly improve the energy efficiency of monitoring operation.

- Sensors in close proximity are likely to have correlated readings and in a majority of these cases, one can predict the reading at a sensor given the knowledge of readings of sensors around it and their past history. A cluster head (base station) may exploit this observation and predict the set of readings that a sensor is going to see in the near future. These predictions are represented concisely as a prediction-model and sent to the sensor. The sensor now needs to transmit its sensed data to the cluster head only when the data differs from the reading given by the prediction model by more than a certain pre-specified threshold.
- The second key observation is that the reliability of readings at the cluster head can be maintained indifferent to the existence of environmental noise. Then, sensors can remain in idle mode until request to transmit its reading by the cluster head is made.

In general, the prediction paradigm prevents a sensor from unnecessarily transmitting all the readings that can be successfully predicted at the cluster head thereby saving energy.

CLASSES OF WIRELESS SENSOR NETWORK

Many classes of wireless sensor networks abound, each with its own characteristics (Subramanium and Katz, 2000). Mechanisms meant for one class may not be appropriate for the other classes. In this study, it is assumed that the number of nodes in the sensor network is not too large and has non-deterministic topology i.e., sensors are thrown randomly in the target field. Other assumptions made in this study include:

- That sensors and base stations have limited energy and communicate sensed data over wireless link.
- That the base station or cluster head and the sensors use Time Division Multiple Access (TDMA) scheme at the link layer to communicate (Heinzelman *et al.*, 2000; Subramanium and Katz, 2000; Pottie and Kaiser, 2000). The TDMA is decided by the base station
- That the base station knows the location of the sensors relative to one another.

Determination of the location of a sensor dynamically in a wireless sensor network is a research area on its own and is orthogonal to the problem being addressed in this paper. Several approaches have been proposed to solve this problem (Niculescue and Nath, 2001; Savvides *et al.*, 2001; Doherty *et al.*, 2001) and their existence is assumed in this research.

PREDICTION-BASED MONITORING

The simplest approach to monitoring in sensor networks would be to have all the nodes send their readings to the base station whenever it changes. This is referred to as the "up-date" mode. The base station collects these updates and sends them to an access point as shown in Fig. 1. Thus at any point in time, a base station maintains the database of current readings of all the sensors under its control.

However, it is an axiom that a group of sensors is more effective in performing a sensing task than one powerful sensor (Pottie and Kaiser, 2000). In such a sensor network, a group of specially proximate sensors are very likely to have correlated readings. The correlation may be

- Spatial: The reading of a sensor being a function of the readings at nearby sensors.
- Temporal: The reading at a sensor being a function of its reading in the past.
- Absolute: The reading that a sensor is going to sense in the near future.
- Spatio-temporal: That reading in sensor X in time slot t is the same as the reading of sensor Y in the previous time slot.

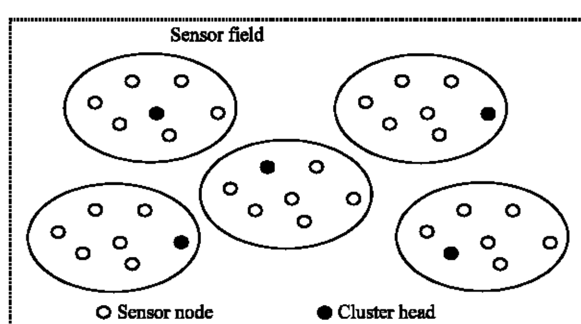

Fig. 1: Sensor network model

The basic procedure for the prediction in this research is as follows: The base station monitors the reading of sensors for some time and generates a prediction model i.e., a concise representation of the readings that a sensor is expected to sense in the near future. It sends this to the appropriate sensors. On receiving a prediction model, sensors change their behaviour and instead of sending an update whenever their reading changes, they now send an update only when their readings differ from the one predicted by the prediction model.

EXPERIMENTAL SET-UP

The experimental setup consists of a Pentium V computer as the base station, an 8-bit random binary generator design in the laboratory to simulate the binary bits from sensor nodes. The binary generator is interfaced to the computer via a parallel port. The binary generator circuitry consists of LEDs to confirm the binary input to the base station as well as the output from the base station. The 8-bit random binary generator simulates the digital bits that are transmitted from the transceiver end of the actual sensor node. The 5 LSB of the 8-bits depict the actual reading while the 3 MSB is used for the naming attribute to locate the sensor node when the prediction model is to be transmitted to the sensing node. The setup is as shown in the block diagram of Fig. 2.

The binary generator is designed to transmit every five seconds using the 555 timer chip. The base station allocates the time to transmit to the sensor nodes on time division basis such that each node transmits twelve times in one minute. The software then generates a database and records the readings of each sensing node as represented by the binary digits from the random generator. After consecutive readings for two minutes i.e., twenty-four readings from each node, a prediction-model is generated and transmitted to the sensing node and this model which is also in form of binary digits is displayed in the LEDs. At this point, the binary generator will cease to transmit until the system is cleared.

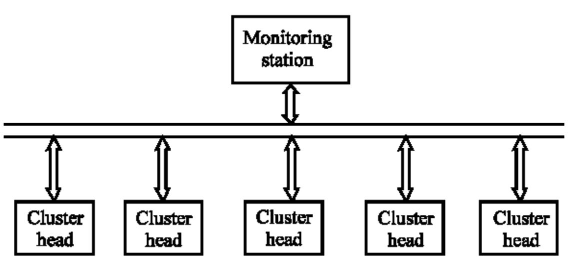

Fig. 2: Set-up block diagram

PROGRAM DEVELOPMENT

The program is developed using Visual basic version 6.0 with temperature as the environmental variable under investigation. The program allows readings from each sensor node every 5 sec and holds them in a database, each time comparing the reading with the previous one. The average of these readings is recorded as the "set-point". The base station then broadcasts this set-point to the nodes in the cluster and sets a threshold of value below and above this set-point which the node most read before sending its reading to the base station. The nodes are then put to "sleep" by the base station for a length of time specified by the base station. The program queries the nodes at interval that is set by the human interface. If the reading in any sensor node is the same with the set-point or within a prescribed threshold, then the base station broadcasts another sleep time for the nodes referred to as the "short sleep" time. The procedure is repeated after an interval that is again set by the human interface. If the readings are still within the set point or its threshold, then a longer sleep time is broadcast to the nodes referred to as "long sleep" time. Else, the nodes are queried more often to determine the trend whether upwards or downwards. This is referred to as "exceptional" time in the program.

If the readings get to an exceptionally high or low value called the "critical point" (also set by the human interface) then a conclusion can be drawn.

RESULTS

The program was tested with the set point put at 30 degree centigrade. The "short sleep" time was set at 1 min and "long sleep" time at 5 min. The "exceptional time" was put at thirty seconds while the "critical point" was put at 60 degree centigrade. The duration of the test was for 1h. During the test run, it was discovered that the nodes were in the sleep mode for more than half the time. Altering the variables i.e., sleep time can further increase the duration of the time in which the nodes are in sleep mode thereby extending the lifetime of the node.

However, the constraints on the quality of the result required by the human interface govern the amount of energy savings that may be achieved. The quality is controlled by two parameters (Goel and Imielinski, 2001):

Delay tolerance: This controls the interval from the time a sensors reading changes to the time it is received at the destination. The more the delay tolerance, the more time is available to generate prediction-models. This increases the potential for generating close-to-perfect prediction models at the base station, which in turn increases the potential for saving energy.

Error tolerance: This controls the amount by which the received data may deviate from its actual reading. The more the error tolerance, the less precise the prediction models need to be and hence more potential for saving energy. In the extreme case when the error tolerance is infinite, the sensors do not have to transmit anything.

Related work: Studies in wireless sensor networks have looked at various ways of saving energy of sensor nodes. Particularly (Hill *et al*., 2000; Rabaey, 2000; Savvudes *et al*., 2001) have proposed exploiting redundancy in sensed readings of sensors that are in closed proximity. However, the approach described in (Pottie and Kaiser, 2000) relies on sensor nodes to figure out among themselves the redundant part in their sensed information and suppress it from transmission. The problem here is that it will take M transmissions for a group of sensor nodes in a cluster to figure out all the redundant information which is costly and may outweigh the advantage of removing redundancy. (Where M is the number of nodes that must be passed through to the destination).

The authors in (Rabaey, 2000) dealt with the problem of how to self-organize sensor nodes that are thrown together so that a meaningful sensing task can be accomplished. The work prescribed compressing information at the nodes acting as cluster heads before sending to a central station. However, the work does not specify any mechanism for performing this compression operation.

The emphasis in (Savvides *et al*., 2001) is on being able to route sensed data to where the interest is. It does not deal with scaling down the information being relayed.

In the research reported here, emphasis is on placing the burden of transmission and computation on the base station which can be served with adequate battery thereby reducing the amount of operations taking place at the nodes thereby saving energy.

Further work: Further work will explore the following future directions:

- Performing simulations to analyze the performance of this mechanism (prediction model) over large sensor networks.
- Performing simulations to analyze the performance of the proposed mechanism over the monitoring of other variables like motion sensors
- Determine the actual hours this mechanism can add to the lifetime of the sensor nodes.

CONCLUSION

In this study, an energy-efficient mechanism for performing monitoring operations in a wireless sensor network to monitor temperature has been presented. Five nodes are used for the demonstration and a computer as the cluster head of the sensor field. The research shows that the average lifetime of a sensor node could be doubled by applying this prediction model. Retaining the reliability of the network is also guaranteed depending on the variable that is being monitored since the lifetime of the sensor node is a function of the energy available to it for computation and transmission of its sensed readings. However, the prediction model presented here may not be suitable for high frequency events. Because of the software nature of the work presented here, there is the possibility of even achieving more than double of the lifetime of the node.

DECLARATION FOR GLOBAL VARIABLES AND SUBROUTINES

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal
value As Integer)
  Public intExceptionallyLow, intToEnableShortSleep,
intToEnableLongSleep As Integer
Public intCase, intSetPoint, intCriticalPoint As Integer
Public intPrevious, intShortTime, intLongTime,
intExceptionTime, intShortSleepTime, intLongSleepTime
As Integer
Public Sub processForLongSleep()
    intToEnableLongSleep = intToEnableLongSleep + 1
    intToEnableShortSleep = 0
    intExceptionallyLow = 0
    If intToEnableLongSleep Mod intLongTime = 0 Then
intCase = 1
End Sub
```

MAIN PROGRAM

```
Private Sub Command1_Click()
Out 888, Val(Text1.Text)
End Sub
Private Sub Form_Load()
'initialise port with test input value
Out 888, Val(Form1.Text1.Text)
```

```
'Monitor for short sleep for nine seconds
intShortTime = 9
'Monitor for Long sleep for ten seconds
intLongTime = 10
'Monitor for Exception for six seconds
intExceptionTime = 6
'Short sleep time is 20 seconds
intShortSleepTime = 20
'Long sleep time is 30 seconds
intLongSleepTime = 30
'set-point is 35
intSetPoint = 35
'critical Point is 70
intCriticalPoint = 70
End Sub
Private Sub Timer1_Timer()
Dim portData As Integer
Static intLongSleepIndex, intShortSleepIndex As Integer
Select Case intCase
Case 0
Form1.Label3.Caption = "Awake"
portData = Inp(888)
If portData > (intSetPoint + 0.15 * intSetPoint) Then
If portData > intCriticalPoint Then
'raise Alarm here
Form1.Label1.Caption = "Dangerous Condition is Likely"
ElseIf portData > (intPrevious + 0.15 * intPrevious) Or
portData < (intPrevious - 0.15 * intPrevious) Then
intToEnableLongSleep = 0
intExceptionallyLow = 0
intPrevious = portData
intToEnableShortSleep = intToEnableShortSleep + 1
If intToEnableShortSleep Mod intShortTime = 0 Then
intCase = 2
Else
intPrevious = portData
Call processForLongSleep
End If
ElseIf portData < (intSetPoint - 0.15 * intSetPoint) Then
intToEnableShortSleep = 0
intToEnableLongSleep = 0
intExceptionallyLow = intExceptionallyLow + 1
If intExceptionallyLow Mod intExceptionTime = 0 Then
Form1.Label1.Caption = "Exceptionally Low Data
Condition": intExceptionallyLow = 0
Else
' process for long sleep
Call processForLongSleep
End If
Case 1
Form1.Label1.Caption = "Normal"
Form1.Label3.Caption = "Normal Sleep"
intLongSleepIndex = intLongSleepIndex + 1
If intLongSleepIndex Mod intLongSleepTime = 0 Then
intCase = 0: intLongSleepIndex = 0
Case 2
Form1.Label1.Caption = "Normal"
Form1.Label3.Caption = "Short Sleep"
intShortSleepIndex = intShortSleepIndex + 1
If intShortSleepIndex Mod intShortSleepTime = 0 Then
intCase = 0: intShortSleepIndex = 0
End Select
End Sub
```

REFERENCES

Agre, J. and L. Clare, 2000. An integrated architecture for co-operative sensing networks. IEEE. Computer Mag., pp: 106-108.

Akyildiz, I.F., W. Su, Y. Sankarasubramanian and E. Cayicri. Wireless sensor networks: A Survey. Computer Networks, 38: 393-422.

Carle, J. and D. Simplot-Ryl, 2004. Energy efficient area monitoring for sensor networks. IEEE. Computer Mag., 37: 40-46.

Doherty, L., L.E. Ghoui and K.S.J. Pister, 2001. Convex position estimation in wireless sensor networks. In: Proceedings of IEEE. INFOCOM, Alaska.

Estrin, D., R. Govindan, J. Heidemann and S. Kumar, 1999. Next century challenges: Scalable co-ordination in sensor networks. In Proceeding of ACM Mobicom, Washington, USA, pp: 263-270.

Goel, S. and T. Imielinski, 2001. Prediction-Based monitoring in sensor networks: Taking lessons from MPEG. In Proceedings of ACM Computer Communication Review, Vol: 31.

Heinzelman, W., A. Chandrasekaran and H. Barakrishnan, 2000. Energy-efficient communication protocol for wireless micro sensor network. In Proceeding of the 33rd Hawaii International Conference on System Sciences (HICSS).

Hill, J., R. Szewczyk, A. Woo, D. Culler, S. Hollar and K. Pister, 2000. System architecture directions for networked sensors. In: Proceedings of the ACM SIGPLAN Notices, 35: 93-104.

Niculescu, D., and B. Nath, 2001. Ad hoc positioning system. Technical report DCS-TR-435, Rutgers University. In: Proceeding of IEEE. Globecom.

Pottie, G.J. and W.J. Kaiser, 2000. Wireless integrated network sensors. In: Proceeding of Communications of the ACMN., 43: 551-558.

Rabaey, J., 2000. Silicon platforms for the next generation wireless systems-what roledoes reconfigurable hardware play? In: Proceedings FPL.

Savvides, A., C.C. Han and M. Srivastava, 2001. Dynamic fine-grained localization in ad hoc networks of sensors. In: Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom) L.A.

Srivastava, M., 2000. Energy efficient wireless systems. In: Proceeding of DIMACS summer school on foundations of wireless networks and applications.

Subramanium, L. and R. Katz, 2000. An architecture for building self-configurable systems. In: Proceeding of IEEE/ACM workshop on mobile ad hoc Networking and Computing (MobiHoc) Boston.