Learning Based Dynamic Secure Load Balanacing In Service Oriented Wireless Sensor Networks

Lata BT, Sumukha TV, *Student Member, IEEE,* Suhas H, *Student Member, IEEE,* Shaila K, Venugopal KR, *Life Member, IEEE,* LM Patnaik

***Abstract*—The abstract goes here.**

***Index Terms*—Wireless Sensor Networks, secure dynamic routing, load-balancing, machine learning**

I. Introduction

WIRELESS Sensor Networks is a network of sensors that are autonomous. These sensors are spatially distributed. They have limited computational and communication power. They do not posses large memory as well.

In WSN, we have a family of sensors called the *service oriented WSNs*. These WSNs are different from the usual ones, in that they have a specific task. They may not be communicating all the time. They will trigger communication only when they come across a state change. For example, consider a WSN monitoring the enemy infiltration. State changes in suce situations do not occur every now and then. But when they do, it must be reported immediately.

In these kind of real time systems, there will be little or room for delay. In such situation comgestion detection becomes critical. Effecient load balancing in order to overcome congestion becomes highly important. A packet should never enter the congested part of the network. Further considering the criticality of the application, it becomes even more important that the data moves along a secure route only. But, since the time and resources are major constraints there is a need for dynamic and adaptive load balancing and security schemes.

Dynamic decision making capability at every node will enhance load balancing and security. Handling unforeseen changes in the network will prove beneficial in real time WSNs. Static routing may fail in such cases and lead to packet loss. Whereas handling such situations intelligently at every hop will not only reduce congestion but also prevent packet loss.

Lata BT is with the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore, 560001 India e-mail: lata_bt@yahoo.com.

Sumukha TV and Suhas H are with UVCE.

Manuscript received April 19, 2005; revised January 11, 2007.

II. Mathematical Model

A. Congestion Detection

In this section, we will discuss a congestion detection model for Wireless Sensor Networks.
Consider a node N with packet arrival rate A_r and packet service rate S_r The traffic at node N is given by,

$$T = \frac{A_r}{S_r} \tag{1}$$

There will be stability at the node only when the following condition is met,

$$T < 1 \tag{2}$$

Let the B_s be the size of the buffer at node N and n be the total number of neighbors of the node N. The probobility that the node is idle at a given instance of time is,

$$\begin{aligned} P_{idle} &= \frac{1}{1 + (\frac{A_r}{S_r}) + (\frac{A_r}{S_r})^2 + (\frac{A_r}{S_r})^3 ....(\frac{A_r}{S_r})^\infty} \\ &= \frac{S_r - A_r}{S_r} \\ P_{idle} &= 1 - (\frac{A_r}{S_r}) \end{aligned} \tag{3}$$

From (1) we can say that,

$$P_{idle} = 1 - T \tag{4}$$

When the number of packets in the buffer is $B_s - N$ we say that the node is tending towards congestion. The probability that the node is tending towards congestion ig given by,

$$\begin{aligned} P(B_s - bufferedPackets = N) &= P_{idle}\Big(\frac{A_r}{S_r}\Big)^N \\ &= \frac{S_r - A_r}{S_r} \times \frac{{A_r}^N}{{S_r}^N} \\ P(B_s - bufferedPackets = N) &= \frac{(S_r - A_r){A_r}^N}{{S_r}^{N+1}} \end{aligned} \tag{5}$$

Probability that there will be a buffer overflow is given by,

$$P_{overflow} = \sum_{i=B_s}^{\infty} \Big(1 - \frac{A_r}{S_r}\Big)\Big(\frac{A_r}{S_r}\Big)^i = \Big(\frac{A_r}{S_r}\Big)^{B_s} \tag{6}$$

In order to prevent packet loss to a certain required value $R_v$the following condition should be met

$$\Big(\frac{A_r}{S_r}\Big)^{B_s} \leq R_v \tag{7}$$

To obtain the buffer size B_s for the required packet loss value R_v,

$$\begin{aligned} \log\left(\frac{A_r}{S_r}\right)^{B_s} &\leq \log(R_v) \\ B_s \log\left(\frac{A_s}{S_r}\right) &\leq \log(R_v) \\ B_s &> \frac{\log(R)}{\log\left(\frac{A_r}{S_r}\right)} \\ B_s &> \log_{\frac{A_r}{S_r}} R \end{aligned} \tag{8}$$

Subsituting (1) in (8),

$$B_s > \log_T R \tag{9}$$

B. Node Strength

Node Strength is the measure of the ability of a node to detect malicious data.
Let K_t be the number of true keymatches at a node N, then the Node Strength NS of N is,

$$NS \propto \sum K_t \tag{10}$$

Further, Node Strength is dependent on the number of false keymatches K_f as,

$$NS \propto \frac{1}{\sum K_f} \tag{11}$$

Combining the above two equations Node Strength is given by,

$$NS = k.\frac{K_t}{K_f} \tag{12}$$

where k is the constant of proportionality.
Node Strength of a node N is a relative parameter, which means that it has meaning only when compared with Node Strengths of other nodes in the same network. As an absolute value, Node Strength has no meaning.
Since all the nodes have the same size of keyspace, we can ignore the constant of proportionality k,

$$NS = \frac{\sum K_t}{\sum K_f} \tag{13}$$

We know that, probability of a true keymatch is proportional to the total number of true keymatches,

$$P_{K_t} \propto \sum K_t \tag{14}$$

$$P_{K_t} = \frac{\sum K_t}{\sum totalKeymatches} \tag{15}$$

The propbability of a false keymatch is directly proportional to the total number of false keymatches,

$$P_{K_f} \propto \sum K_f \tag{16}$$

$$P_{K_t} = \frac{\sum K_f}{\sum totalKeymatches} \tag{17}$$

Therefore we can say that,

$$\frac{P_{K_t}}{P_{K_f}} = \frac{\sum K_t}{\sum K_f} = NS \tag{18}$$

Further, we can say that,

$$P_{K_f} = 1 - P_{K_t} \tag{19}$$

Substituting for P_{K_f},

$$NS = \frac{P_{K_t}}{1 - P_{K_t}} \tag{20}$$

III. Secure Dynamic Load Balancing Sceheme

A. Dynamic Load Balancing

The congestion detection algorithm basically classifies a node as 'Tending Towards Congestion [TTC]' or 'Available'. A node will be classified as TTC if its buffer is so much filled that it can atmost accomodate only one packet sent by each of its neighbour. A node n with buffer size B_s and a total number of neighboring nodes N will tend towards congestion when the number of packets in its buffer is equal to $B_s - N$. Once the number of packets in the buffer is lesser than $B_s - N$, it will be classified as available.

Every node will maintain a *Neighbour Information Table [NIT]*. This table will contain information whether each of its neighbours is tending towards congestion or available. Once a node realizes that it is tending towards congestion, it will imediately send a message to all its neighbours updating them about its status. A node which sends such a message must also send an *available* message to all its neighbours as soon as it comes out of the TTC situation.

Once a node receives status information about its neighbours, it will update its *Neighbour Information Table*. When a packet has to be routed, the node will first obtain the multiple paths. Then, the first node on each of these paths is taken and checked for congestion. If the NIT says that the node is available, then the node is retained, else the node gets rejected.

B. Node Strength

In this phase, the nodes that clear the congestion detection test will be then tested for node strength. The node strength of a node will tell us the capability of that node to detect malicious data.

To determine the node strength of a particular node, we need to obtain the total number of true keymatches and the total number of false keymatches of that node. Once this data is obtained, the node strength of that node is derived by dividing the total number of true keymatches by the total number of false keymatches. Comparing the node strengths of multiple nodes, we can get a clear picture of the most secure node. The node with the highest node strength will be the most secure node. This means that the node has a good track record of determining legitimate and malicious data.

C. Packet Routing

Once we select the node with the higest node strength, we can be sure that the node is least congested as well. This is because, the node has been shortlisted for node strength analysis only after analysing its congestion status.

Now, the packet will be routed to that node. At the next node the entire secure dynamic route selection procedure is executed to determine the next hop for the packet. By doing so we can be sure that we will be adapting to any of the unforeseen changes in the network that may not be known to the source prior to the routing.

IV. Learning

A. Route Statistics Collection

This phase is a training data collection collection phase. It will run parallely from the begining of the data transmission uptil the prediction phase begins. The source node maintains a *Learning Table* that will facilitate the collection of route information. The *Learning Table* consists of a column for the route taken by a packet, one column for the number of packets that went along that route, one column for the sum total of the delay of every packet that went along that route. This is needed to obtain the average delay in future.

As soon as the source node receives an acknowledgement from the destination, it will look for the route information sent by the destination and the timestamp. If the route has not been entered in the *Learning Table* previously, it means that it is a new route taken by a packet. In such a case, a new entery is made to the *Learning Table*. If the route already already exists, the packet count for that entry will be increamented and the sum total of the delay will be updated by determing the packet delay using the timestamp sent by the destination and adding it to the existing value in the corresponding entry of the *Learning Table*.

B. Weight Assignment

In this phase, each route will be analysed to determine the best route for a prediction. Once the treshold number of packets have been transmitted, i.e once we have collected sufficient training data, we must analyse each of the routes and assign weights to them. The weight of a route is the trust factor of that route.

A route is trust worthy if it has lesser delay and a good number of packets have been sent along that route. A route with least delay alone or a route along which the maximum packets have been sent alone cannot be declared as the most trusted route. Trust factor is a more comprehensive value which evaluates both the delay of the route and the number of packets sent along the route.

The weight of a route is determined by dividing the quotient obtained by dividing the average delay along that route by the sum total of the average delay along all the route, by the quotient obtained by diving the number of packets sent along the route by the total number of packets sent by the source.

C. Prediction and Feedback

The weights of each route will give us an idea about the trust factor of each route. The route with least weight will mean that it has lesser delay and a good number of packets have been sent along that route compared to other routes. We then choose this path to route the next packet.

During the prediction phase, the path for a packet will be fixed. This means that at every node the congestion detection scheme and the security scheme need not be executed. This is because we have decided to route the packet along this path only after studying the routes performance until the threshold number of packets were transmitted. Since each node along this route has passed the congestion detection test and the node strength analysis, the performance of the route will reflect in the weight of that route which was considered for deciding to transmit the packet along this route.

In order to bypass the congestion detection and node strength schemes, a flag is set in the packet. Further the entire route will be mentioned in the packet before routing it. At every hop, when an intermediate node gets this packet, it checks for the bypass flag. If it is set, the node will blindly route it to the next hop mentioned in the packet. Once the packet reaches the destination, the destination will send the route details and the timestamp of packet arrival back to the source.

The source will then compute the delay obtained along the route and repeat the weight assignment procedure and then make the next prediction. This scheme is also dynamic because, if a wrong prediction is made, that means, if the delay along the route increased due to some reason unknown to the source, the feedback from the destination will enlighten the source about the situation. The weight will be accordingly adjusted which will then reduce the trust factor of such routes.

V. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.