SALR: Secure Adaptive Load-Balancing Routing in Service Oriented Wireless Sensor Networks

Lata B T
Department of CSE
UVCE, Bangalore University,
Bangalore 560 001.
Email: lata_bt@yahoo.co.in

Sumukha T V
Department of CSE
UVCE, Bangalore University,
Bangalore 560 001.

Suhas H
Department of CSE
UVCE, Bangalore University,
Bangalore 560 001.

Tejaswi V
Department of CSE,
NITK, Surathkal.

Shaila K
Department of CSE
UVCE, Bangalore University,
Bangalore 560 001.

Venugopal K R
Department of CSE
UVCE, Bangalore University,
Bangalore 560 001.

Dinesh Anvekar
Department of CSE,
Nitte Meenakshi Institute of
Technology (NMIT), Bangalore.

L M Patnaik
Indian Institute of Science,
Bangalore, India.

***Abstract*—Congestion control and secure data transfer are the major factors that enhance the efficiency of *Service Oriented Wireless Sensor Networks*. It is desirable to modify the routing and security schemes adaptively in order to respond effectively to the rapidly changing Network State. Adding more complexities to the routing and security schemes increases the end-to-end delay which is not acceptable in *Service Oriented WSNs* which are mostly in real time. We propose an algorithm *Secure Adaptive Load-Balancing Routing (SALR)* protocol, in which the routing decision is taken at every hop considering the unforeseen changes in the network. Multipath selection based on Node Strength is done at every hop to decide the most secure and least congested route. The system predicts the best route rather than running the congestion detection and security schemes repeatedly. Simulation results show that security and latency performance is better than reported protocols.**

***Keywords*—Wireless Sensor Networks, secure adaptive routing, load-balancing, network security, multipath, machine learning, hop-by-hop routing**

I. Introduction

Wireless Sensor Network is a network of sensors that are autonomous and are spatially distributed for capturing data. Sensors have restricted computational and communication power with little memory and limited battery power. The data collected by the individual sensors is then passed on to the base station or the sink. The sink processes the accumulated data for the specific application. Sensor networks have been widely used in military applications, environment monitoring, health-care applications and surveillance.

In a class of WSNs, known as the *service oriented WSNs*, Sensors have a specific task, and may not be communicating all the time. They trigger communication only when they come across a state change. It is necessary to have a robust routing technique that is adaptive to every change in the network along the path of the packet.

Motivation: The resources of a sensor node such as computational power and battery life is limited. Most protocols remain static and do not adapt to the rapidly changing state of the network. Both these classes of protocols do not facilitate the efficient functioning of a service oriented WSNs.

Contribution: In the proposed scheme, every sensor node monitors the load and the strength of each of its neighbours to determine malicious data. It transmits data only to those nodes that are least congested and highly secure. Since the analysis of the two parameters at every hop introduces an overhead in the network we have a feedback system that enables the network to learn from every earlier decision.

Organization:The rest of this paper is organized as follows: Section II provides a brief review of related works. The background of the paper is discussed in Section III. Section IV defines Problem Definition. The Mathematical model is explained in Section V and its Implementation is shown in Section VI and VII. The simulation and performance evaluation is contained in Section VIII. Conclusions are presented in Section IX.

II. Related Work

Uluagac et al., [1] designed a scheme called SOBAS (Secure SOurce-BAsed Loose Synchronization). It securely synchronizes events in the network without transmission of explicit synchronization messages. High clock precision has not been achieved. Ameer et al., [2] presented a Least-Disruptive topology Repair (LeDiR) algorithm. It restores the connectivity without extending the length of the shortest path among nodes compared to the prefailure topology. LeDiR is resilient to single node failure at a time. This work cannot handle simultaneous node failures.

Tao et al., [3] introduced mechanisms considering single domain that generate randomized multipath routes. Routes taken by the *shares* of different packages change over time. Besides randomness, the generated routes are also highly dispersive and energy efficient, making them quite capable of overcoming black holes. Guoxing et al., [4] explore Trust Aware Routing Framework (TARF) that avoids replay attack. The TARF algorithm keeps track of trustworthiness of its

978-1-4799-1823-2/15/$31.00 ©2015 IEEE

neighbors and selects the path based on the trust values. TARF is scalable to medium scale test beds. The protocol needs to be further evaluated with large-scale WSNs deployed in wild environments.

Shuai et al., [5] proposed a data collection scheme Maximum Amount Shortest Path (MASP) for Wireless sensor Networks with path constrained mobile sinks. This work does not address various movement trajectories of mobile sinks, subsink selection problem and security. Bing and Kwan [6] developed a framework for feedback-based scheduling algorithms. It achieves best delay and throughput under various traffic conditions. Security is not addressed in their work. Maciej and Sarangapani [7] presented a novel, decentralized, predictive congestion control (DPCC) scheme for wireless sensor networks (WSN). It guarantees weighted fairness during congestion by updating weights associated with each packet. This work can be extended for real world applications. Fenye et al., [8] developed a probability model, where multidimensional trust attributes such as subjective trust and objective trust are considered. Subjective trust is generated as a result of protocol execution at runtime, while objective trust is obtained from actual node status.

Prajakta et al., [9] designed congestion management algorithm called Congestion Avoidance and Route Allocation using Virtual Agent Negotiation (CARAVAN). The virtual nature of these deals requires no physical communication and, thereby, reduces communication requirements. It requires more travel time in comparison with shortest path algorithm. Shancang et al., [10] developed SM-AODV that adopts an adaptive congestion control scheme, which is effective even in the case of node or link failure.

III. Background Study

Shancang et al., [10] proposed SM-AODV (Secure Multipath AODV) protocol which has an evaluation metric, path vacant ratio, to evaluate and find a set of node-disjoint paths from all available paths in service oriented wireless sensor networks. SM-AODV includes three phases.

Phase one: Packet Delivery Scheme

In this phase, data is split into multiple data segments by using threshold secret sharing scheme. Data can be recovered from T received packets from a N split packets, then the scheme is called (T,N) threshold secret sharing scheme.

Phase Two: Multipath Evaluation and Scheduling

It involves five steps. (i) *Multipath Discovery*: where all node disjoint paths from source to destination are obtained, (ii) *Multipath Load Balancing Evaluation*: where Vacant rate of each path is evaluated, (iii) using *Threshold Secret Sharing Scheme*: when load is split on multiple paths, (iv) *Path Vacant Ratio*, where load is split on multiple paths, (v) *Congest events* are re-monitored by congest event module. If congest event occurs, then the congest control mechanism is invoked and load is forwarded according to the congest-level.

Phase Three: Congestion Control

Three parts are included in this phase. They are congestion detection, Congestion control and notification, and congestion cancellation and load adjusting. SM-AODV is effective in the case of node or link failure. This work needs to be further enhanced in terms of security by including intermediate trustworthiness which is implemented in our paper.

IV. Problem Definition

Most applications of Service Oriented Wireless Sensor Networks are meant to be real time. But achieving real time capability over a network is indeed a challenge. Unaddressed or even inefficiently addressed congestion issues in such service oriented networks can increase the end-to-end delay exponentially. Such a situation is never acceptable in a highly responsive real time application. At the same time, it is equally important to make sure the network is highly secure. The security procedures definitely add to the latency further delaying the transmission. *The objective is*

1) to provide high security with minimum overhead.

V. Mathematical Model

A. Congestion Detection

In this section, we 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,

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

The node is stable only when the following condition is met,

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

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

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

From (1) we can infer that,

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

When the number of packets in the buffer is $B_s - N$ we can interpret that the node is tending towards congestion.

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 obtained as,

$$NS \propto \sum K_t \tag{5}$$

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

$$NS \propto \frac{1}{\sum K_f} \tag{6}$$

Combining the above two equations Node Strength can be expressed as,

$$NS = k.\frac{K_t}{K_f} \tag{7}$$

where k is the constant of proportionality.

C. Trust Factor

Trust Factor of a path p_i is a value that signifies the quality of the route. It is not just the delay that defines the quality of the route. Here, we consider both the number of packets successfully transmitted in that route and the average delay along the route to define the quality of the path.

$$T_f \propto Number\ of\ packets\ transmitted \tag{8}$$

$$T_f \propto \frac{1}{delay} \tag{9}$$

$$P_r = \frac{Packets\ transmitted\ along\ path\ p_i}{Total\ number\ of\ packets\ transmitted} \tag{10}$$

$$D_r = \frac{Average\ delay\ along\ path\ p_i}{\sum_{j=0}^{n} Average\ delay\ of\ pathp_j} \tag{11}$$

$$T_f = \frac{P_r}{D_r} \tag{12}$$

The above equation suggests that as the number of packets being sent along the path increases, the average delay along the path decreases and the trust factor for the path increases.

VI. Secure Adaptive Load-Balancing Routing Protocol (SALR)

The *Secure Adaptive Load-Balancing Routing Protocol* is divide into three parts. (i)Adaptive load balancing, (ii) Security based on Node Strength and (iii) Route prediction based on learning from previously chosen routes.

A. Congestion Detection

Our approach to solving the congestion problem is different from the conventional one in that, we are trying to prevent congestion from taking place rather than redistributing the load after a congestion has taken place. Our load balancing scheme is such that each and every node in the network is continuously monitored for congestion. When the system feels that a particular node is going to be congested in the near future, then a dynamic load balancing scheme is incorporated to prevent that node from entering into a congestion state (Algorithm SALR).

The congestion detection algorithm basically classifies a node as *Tending Towards Congestion [TTC]* or *Available*. A node is classified as TTC if its buffer can at-most accommodate only one packet sent by each of its neighbour. Every node maintains information about its buffer capacity and the current number of packets in its buffer. Once it realizes that it is tending towards congestion, it immediately sends out 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.

TABLE I: Table of notations

Symbol	Definition
A_r	packet arrival rate
S_r	packet service rate
T	traffic
B_s	size of the buffer
R_v	required value of packet loss
NS	Node Strength
K_t	number of true key matches
K_f	number of false key matches
PNL	Potential Neighbor List
PN	Potential Neighbor
SNL	Secure Neighbor List
SN	Secure Neighbor
$path_i$	a path i from source to destination
LT	Learning Table
n_i	no. of packets sent along $path_i$
n_t	total no. of packets sent
$delay_i$	delay along $path_i$
$delay_{rec}$	delay received in the ack for $path_i$
W_i	Weight of $path_i$

B. Node Strength

In this phase, the nodes that clear the congestion detection test are checked for node strength. The node strength of a node is 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. A node with the highest node strength among other nodes is chosen to route the packet.

C. Packet Routing

Once the node with the highest node strength is selected, it is certain that the node is least congested as well. At the next node, the entire dynamic secure route selection procedure is executed to determine the next hop for the packet. The path may not remain uncongested or secure for ever. Therefore we cannot rely on the same path throughout the transmission.

VII. Learning and Prediction

The computational overhead leveraged on the network because of dynamic load balancing and secure route selection based on node strength is significant. In order to make sure that this does not impact the end-to-end delay of the transmission of data, the system continuously learns from previous routing decision. This significantly reduces the time required for determining the route based on the two schemes discussed earlier. In this section, we discuss an efficient way to learn and predict routes (See Algorithm SALR).

A. Route Statistics Collection

Proper prediction requires a good amount of training data to support it. The collection of the training data to make a reliable prediction in future happens in this phase.

Algorithm 1: SALR: Secure Adaptive Load-Balancing Routing

Phase 1: Congestion Detection
Input: $availableMultipaths$
Output: PNL
begin
  if $byPass == FALSE$ **then**
    $PNL \leftarrow availableMultipaths$
    if $CONGEST_{node} == TRUE$ **then**
      $PNL \leftarrow PNL - node$
    else if $AVAILABLE_{node} == TRUE$ **then**
      $PNL \leftarrow PNL + node$
  else
    goto $Phase\ 3$
end

Phase 2: Node Strength
Input: PNL
Output: SN
begin
  if $byPass == FALSE$ **then**
    for $PN\ in\ PNL$ **do**
      $NS \leftarrow getNS(PN)$
      $SNL \leftarrow append(\{PN, NS\})$
    $SN \leftarrow getMaximumNS(SNL)$
  else
    Transmit the packet
end

Phase 3: Constructing the LT
Input: $path_i$
begin
  if $path_i\ in\ LT$ **then**
    $n_i \leftarrow n_i + 1$
    $delay_i \leftarrow delay_i + delay_{rec}$
    $update_{LT}(path_i, delay_i)$
  else
    $n_i \leftarrow 1$
    $delay_i \leftarrow delay_{rec}$
    $insert_{LT}(path_i, delay_i)$
end

Phase 4: Weight Adjustment
Input: $n_i, delay_i$
begin
  $AvgDelay_i \leftarrow \frac{delay_i}{n_i}$
  $r_n \leftarrow \frac{n_i}{n_t}$
  $r_{AvgDelay} \leftarrow \frac{AvgDelay_i}{AvgDelay_t}$
  $W_i \leftarrow \frac{r_n}{r_{AvgDelay}}$
  $update_{LT}(path_i, W_i)$
end

Phase 5: Prediction
Input: LT
begin
  $p_j \leftarrow getRouteWithMaxWeight(weight_{p_i})$
  $byPass \leftarrow TRUE$
  $packet \leftarrow append(p_j, timestamp)$
  $neighbour \leftarrow getNextNode(p_j)$
  $send(packet, neighbour)$
  if $node_i == dest$ **then**
    $ack \leftarrow append(p_i, delay)$
    $send(ack, source)$
  goto $Phase\ 1$
end

TABLE II: Simulation Parameters

Simulator	**NS-2.35**
Duration	100s
Sample Rate	1s
Area	1000 m^2
Radio Range	300 m
Thres Dis	175 m
Thres temp	75 ^0C
Thres pres	675 mmHg
Thres smoke	40 mgL^{-1}

B. Weight Assignment

Once the threshold number of packets have been transmitted, i.e., once sufficient training data is collected, each of the routes is analyzed and weights are assigned to them. The weight of a route is the trust factor of that route. The weight of each route is compared to determine the best route for a prediction. A route is trust worthy if it has lower delay and a good number of packets have been sent along that route. (See equation 12)

C. Prediction and Feedback

This is the final phase in which the system predicts an appropriate route for transmission of the packet. The weights of each route reflects the trust factor of that route. The route with highest weight is the one that has lower delay and has transmitted a good number of packets compared to other routes. Such a path which is trustworthy is then chosen to route the next packet.

VIII. Simulation and Performance Analysis

In this section, we evaluate the efficiency of our scheme based on the data loss ratio, the packet delivery ratio, the avg. delay, and compare these results with SM-AODV, a similar multipath dynamic routing scheme.

A. Simulation Setup

Our algorithm is implemented using the discrete event network simulator NS-2.35. The area of node deployment is 1000m x 1000m with the base station positioned close to the origin at (100, 100). The remaining nodes are deployed randomly. The base station is placed at the bottom left corner of the deployment area, so that it is outside the danger zone. Hence, in case any accidents occur in the deployment site the base station is not affected. The simulation parameters are given in Table III. The number of sensor nodes are varied from 100 to 150 and simulation runs are carried out for a duration 100 seconds. It is assumed that the network topology is known and multipaths can be found in each source destination pair which is at least three hops.

B. Simulation Results

Data loss ratio is a metric which can illustrate the dynamic adaptability of the congestion control scheme of SALR. *Figure* 1 shows the data loss ratio with different paths for SM-AODV and SALR, where a fixed data stream is generated with Constant Bit Rate (CBR). SALR protocol shows an improvement in data loss ratio of up to 62.5% when compared

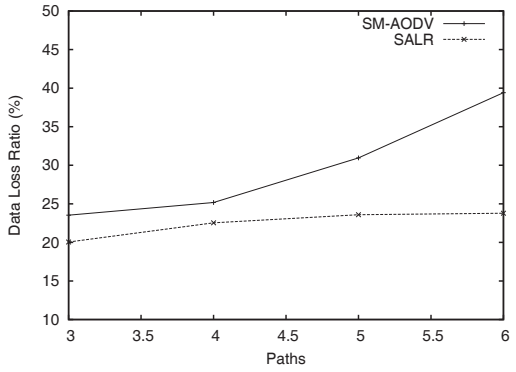

Fig. 1: Data loss ratio against different number of paths

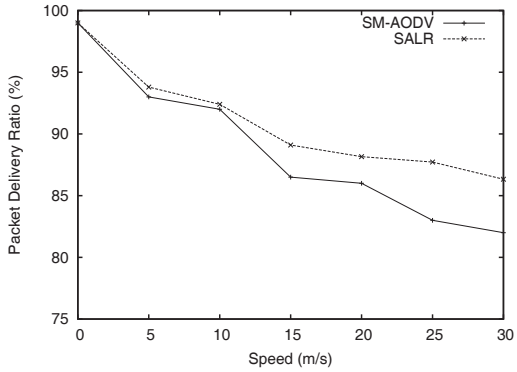

Fig. 2: Packet Delivery Ratio against mobility of nodes

with SM-AODV. When the number of paths is less, the data loss ratios of both SALR and SM-AODV are quite close. However, an increase in the number of paths has an adverse effect on the performance of SM-AODV while the SALR protocol is much more stable. This is mainly because of our dynamic load balancing scheme which determines if a node is *Tending Towards Congestion [TTC]*.

Figure 2 shows an improvement in the packet delivery ratio of about 6% is achieved when compared to SM-AODV. Initially, at lower levels of node mobility not much difference is observed in the performance of SALR and SMAODV. But as the mobility in the network increases, SALR achieves a considerable degree of improvement in successfully delivering packets to the destination. This is a direct consequence of our *Node Strength* phase which determines the most secure node based on the node's ability to detect malicious data. This increases the overall reliability of the network, which helps in establishing trustworthiness of the sensor network which is extremely essential in real applications.

IX. Conclusion

Service oriented WSNs are a special kind of WSN in which real time reliable data delivery is a major requirement. Our proposed SALR protocol caters to such requirements by adopting a learning based dynamic load balancing model with advanced security.

The algorithm employs a hop-by-hop mechanism in which each intermediate node determines the least congested neighbor which has the highest node strength before forwarding a packet to it. Performing both congestion detection and security analysis at every hop can result in increased delay and energy consumption. To overcome this a feedback mechanism is employed in which the source keeps track of all the available multipaths and their corresponding delays. We achieve considerable improvement in average delay, packet delivery ratio and data loss ratio when compared to SMAODV by incurring a little memory overhead while collecting training data. Further, the feedback is continued even after the source chooses a path in order to adapt to any future changes in the network characteristics. We have developed a mathematical model to detect congestion and to measure node strength and trust factor. Future work would involve determining an exact threshold point for commencing the prediction phase which would provide a balanced trade-off between delay and security.

References

[1] A. Uluagac, R. Beyah and J. Copeland, "Secure SOurce-BAsed Loose Synchronization(SOBAS) for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 24, no. 4, pp. 803-813, April 2013.

[2] Ameer A. Abbasi, Mohammed F. Y. and Uthman A., "Recovering From a Node Failure in Wireless Sensor-Actor Networks with Minimal Topology Changes," *IEEE Transactions on Vehicular Technology,* vol. 62, no. 1 , pp. 256-271, January 2013.

[3] T. Shu, M. Krunz and S. Liu, "Secure Data Collection in Wireless Sensor Networks Using Randomized Dispersive Routes," *In IEEE Transactions on Mobile Computing,* vol. 9, no. 7, pp. 941-954, July 2010.

[4] Guoxing Zhan, Weisong Shi and Julia Deng "Design and Implementation of TARF:A Trust-Aware Routing Framework for WSNs," *IEEE Transactions on Dependable and Secure Computing,* vol. 9, no. 2, pp. 184-197, 2012.

[5] Shuai Gao, Hongke Zhang and Sajal K. Das, "Efficient Data Collection in Wireless Sensor Networks with Path-Constrained Mobile sinks," *IEEE Transactions on Mobile Computing,* vol. 10, no. 5, pp. 592-608, April 2011.

[6] Bing Hu and Kwan L. Yeung, "Feedback-Based Scheduling for Load-Balanced Two-Stage Switches," *IEEE Transactions on Networking,* vol. 18, no. 4 , pp. 1077-1090, August 2010.

[7] Maciej Zawodniok and Sarangapani Jagannathan, "Predictive Congestion Control Protocol for Wireless Sensor Networks," *IEEE Transactions on Wireless Communications,* vol. 6, no. 11, pp. 3955-3963, November 2007.

[8] Fenye Bao, Ing-Ray Chen, MoonJeong Chang and Jin-Hee Cho, "Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management,* vol. 9, no. 2, pp. 169-183, June 2012.

[9] Prajakta Desai, Seng W. Loke, Aniruddha Desai, and Jugdutt Singh, "CARAVAN: Congestion Avoidance and Route Allocation Using Virtual Agent Negotiation," *IEEE Transactions on Intelligent Transportation Systems,* vol. 14, no. 3, pp. 1197-1207, Sept. 2013.

[10] Shancang Li, Shanshan Zhao, Xinheng Wang, Kewang Zhang and Ling Li, "Adaptive and Secure Load-Balancing Routing Protocol for Service-Oriented Wireless Sensor Networks," *IEEE Systems Journal,* vol. 8, no. 2, pp. 1-10, June 2013.