Trust-based Backpressure Routing in Wireless Sensor Networks

Revathi Venkataraman*

SRM University, Chennai, India
Email: revathi.n@ktr.srmuniv.ac.in

Scott Moeller

University of Southern California, Los Angeles, CA, United States
Email: electronjoe@gmail.com

Bhaskar Krishnamachari

University of Southern California, Los Angeles, CA, Unites States
Email: bkrishna@usc.edu

T. Rama Rao

SRM University, Chennai, India
Email: ramarao.t@ktr.srmuniv.ac.in
*Corresponding author

Abstract: In this paper, we apply a Vector AutoRegression (VAR) based trust model over the Backpressure Collection Protocol (BCP), a collection mechanism based on dynamic backpressure routing in Wireless Sensor Networks (WSN) and show that the VAR trust model is suited for resource constraint networks. The backpressure scheduling is known for being throughput-optimal. However, it is usually assumed that nodes cooperate with each other to forward the network traffic. In the presence of malicious nodes, the throughput optimality no longer holds and this affects the network performance in collection tree applications of sensor networks. We apply an autoregression based scheme to embed trust into the link weights, making it more likely for trusted links to be scheduled. The novelty in our approach is that the notion of trust can be easily incorporated in a new state of the art distributed and dynamic routing Backpressure Collection Protocol in sensor networks. We have evaluated our work in a real sensor network testbed and shown that by carefully setting the trust parameters, substantial benefit in terms of throughput can be obtained with minimal overheads. Our performance analysis of VAR in comparison with other existing trust models demonstrate that even when 50% of network nodes are malicious, VAR trust offers approximately 73% throughput and ensures reliable routing, with a small trade-off in the end-to-end packet delay and energy consumptions.

Keywords: Backpressure routing, Floating Queues, Trust metrics, Sensor trust

Reference to this paper should be made as follows: Venkataraman, R., Moeller, S., Krishnamachari, B., RamaRao, T., (2014) 'Trust-based Backpressure Routing in Wireless Sensor Networks', *Int. J. of Sensor Networks*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Revathi Venkataraman received her PhD in Computer Science and Engineering in SRM University in 2013. Her research interests include wireless networks and security, trust computing, wireless ad hoc and sensor network testbed developments which are ongoing research activities funded by Indian Government.

Scott Moeller completed his PhD in Electrical Engineering from University of Southern California in 2010. His research interests include stochastic network optimization, systems implementation and characterization of source utility optimization and dynamic routing algorithm implementations for Wireless Sensor Networks.

Bhaskar Krishnamachari received his B.E. in Electrical Engineering at The Cooper Union, New York, in 1998, and his M.S. and Ph.D. degrees from Cornell University in 1999 and 2002 respectively. He is currently an Associate Professor and a Ming Hsieh Faculty Fellow in the Department of Electrical Engineering at the University of Southern California's Viterbi School of Engineering. His primary research interest is in the design and analysis of algorithms and protocols for next-generation wireless networks.

T. Rama Rao is working as Professor and Head of Telecommunication Engineering department, SRM University, India. He received his PhD degree from Sri Venkateswara University, Tirupati in 2000. His research interests are, Radio Channel Measurements and Modeling, Broadband Wireless Communications / Networks and Wireless Information Networks.

1 Introduction

Wireless sensor networks are providing solutions to plenty of real world challenges at a very low cost and are enabling applications Glisic (2006); Edgar H. Callaway (2004); Chong and Kumar (2003); Culler et al. (2004) in a variety of fields. They include large-scale monitoring of the earth environment (marine, soil and atmosphere), civil structures Xu et al. (2004), and animal habitats Cerpa et al. (2001); Mainwaring et al. (2002); Szewczyk et al. (2004), industrial sensing and diagnostics Rajeev et al. (2006). They form the robust backbone for various data collection applications like gathering sensor information in inhospitable locations, biochemical hazard detection, health monitoring and situational awareness in a military environment. In essence, wireless sensor networks provide the end user with intelligence and a better understanding of the environment. As these devices emerge to become an important part of our lives, so is their rising security concern Araujo et al. (2012). Many of the security approaches suitable for web services cannot be directly applied to these wireless networks, due to their limited resources of energy, bandwidth, computation and memory.

Numerous cryptographic schemes and techniques Traynor et al. (2007); Zhu et al. (2003); Liu and Ning (2003) have been applied in the wireless communication systems, which provide data confidentiality, integrity, access control, authentication and non-repudiation. These techniques are categorized as hard security approaches. There is a class of threats termed as soft security threats, where the user will be a legitimate compromised node in the network. These wireless nodes may either behave selfishly by not forwarding the data packets so as to save their resources, or maliciously to launch a Denial of Service attack on the entire network. Being an authenticated node, they will not be detected by the conventional cryptographic schemes. Hybrid security measures involving both cryptographic and trust-based schemes are needed to ensure complete security in a system.

A very common pattern of communication in WSN is the collection mechanism, whereby information from various sensor nodes will be gathered in a single sink. One such work based on dynamic backpressure routing is the Backpressure Collection Protocol Moeller et al. (2010). Here, the route computation is done on a per-packet basis, which takes into consideration the local queue size and the link transmission parameters. A wide variety of applications can be envisaged in wireless sensor networks using BCP. Security services like authentication and confidentiality are critical to secure communications in wireless sensor networks deployed in hostile environments. Cryptographic techniques like SNEP, TESLA Perrig et al. (2001) and TinySec Karlof et al. (2004) can be used to provide authentication, confidentiality and data freshness. Even these hard security measures does not guarantee a complete security solution. In defence environments and other geographically hostile locations, the sensor devices remain unattended for a long time after deployment and are vulnerable to adversary attacks. Such compromised sensor devices will act as legitimate users in the network and cause maximum damage to network resources.

Trust based soft security schemes bridge the gap by addressing these challenges. They detect these attacks by monitoring neighbors for behavioral anomalies and quantifying these results into direct trust metrics. Through this mechanism, the misbehaving neighbors are easily identified and alternate routing mechanisms are employed to achieve reliable communication in the network with minimum loss of resources. Although numerous cryptographic and statistical schemes are presented in the literature for wireless networks, there is no implemented work on practical systems related to trust models with dynamic backpressure routing in WSN. This may be due to constraints like computational overheads, large delays and packet losses in these networks. Addressing these concerns, we present the first implemented work on Vector AutoRegressive (VAR) trust model for WSN that works with dynamic backpressure routing.

VAR models are generally used for analysis of multivariate time series. They describe the dynamic behavior of economic and financial time series for forecasting. Feature classification Anderson et al. (1998), mobility tracking Zaidi and Mark (2011) and semantic web applications Qiu and Chen (2008) are some of the other scenarios in which the multivariate autoregressive models are used. Our earlier work on VAR trust model Venkataraman et al. (2012b,a) over wireless ad hoc routing protocols showed interesting results and tradeoffs where the proposed trust model is compared with SRAC Yu et al. (2009), SLSP Papadimitratos and Haas (2003) and SMT Papadimitratos and Haas (2006). SRAC is suited to reactive ad hoc routing protocols where routing decisions are based on neighbor's trustworthiness and performance. SRAC focuses on internal attacks caused by authenticated routers in the network. Similarly, SLSP secures the links and when combined with SMT offers a secure data communication via redundant paths in an ad hoc network. In this work, we propose to implement the VAR trust model over a dynamic and distributed routing protocol, BCP in sensor networks and show that the VAR trust model is very much suited for resource constraint sensor networks. We also present the performance analysis and results of the VAR model in comparison with SRAC and SLSP / SMT over an IEEE 802.15.4 network and show that the VAR trust model outperforms these existing trust models that are available in literature.

The collection tree applications of WSN, especially with backpressure routing are prone to many security attacks. First, the compromised node receives the data packets from its neighbors and may indulge in Denial-of-Service attacks by either dropping all the data packets or selectively forwarding some of the data packets.

Second, the malicious nodes may either change the packet header information or modify the packet contents leading to loss of data integrity in the communication. Third, the adversaries may advertise false queue sizes in backpressure routing. If a neighboring node does not wish to cooperate in data forwarding, it may advertise a maximum data queue size. Since the backpressure routing is based on queue gradients developed from source to sink, this non cooperating neighbor will not receive any data packet for forwarding. On the other hand, a node may also advertise a low queue size, thereby attracting data traffic from its neighbors and later on indulge in Denial-of-Service attacks. Either way, the network performance is severely degraded in the presence of these compromised nodes. Finally, the malicious neighbors may not be fair in choosing its best neighbor for forwarding of data packets. It may either misroute the packets to random neighbors or to an already overloaded neighbor, leading to delay and loss of data packets.

The main contributions made in this work are as follows. We integrate the VAR trust model into the state of the art, low overhead, dynamic backpressure routing protocol for sensor networks, BCP. In order to do this, we develop custom VAR trust metrics for BCP. This is the first time such a distributed trust mechanism has been implemented for routing in sensor networks. We have more-over implemented this mechanism in TinyOS and experimentally verified its performance over a 25-node sensor network testbed, demonstrating that it provides good network performance in the presence of maliciously compromised nodes, while inducing minimal computation overhead.

The rest of this paper is organized as follows. The related works on trust in wireless sensor networks and the concept of BCP are explained in Section 2 . The modified routing for BCP with trust is explained in Section 3. The experimental results and analysis of the VAR trust model over BCP which are carried out in a sensor network testbed is presented in Section 4. The performance of the VAR trust model in comparison with other existing trust models in IEEE 802.15.4 network scenarios are discussed in Section 5. And in Section 6, we conclude by pointing to future extensions of this work.

2 Related Works

Trust-based routing in networks are generally used to mitigate the soft security threats, which are lacking in the traditional cryptographic schemes. Some of the hard security measures, tailor-made for sensor networks include SNEP, μTESLA Perrig et al. (2001) and TinySec Karlof et al. (2004). SNEP was designed to provide data confidentiality, authentication and data freshness, while μTESLA provides broadcast authentication in wireless sensor networks. TinySec ensures authenticity, integrity and confidentiality of messages. However, these schemes do not address the legitimate, authenticated nodes, which misbehave due to compromise. Our work is the first implemented soft security measure on the state of the art dynamic BCP in sensor networks.

Hybrid key distribution mechanisms, LION and TIGER Traynor et al. (2007) for sensor networks were proposed by Traynor et al. LION is a key distribution scheme suited for distributed sensor networks, while TIGER is a KDC based distribution mechanism for sensor networks similar to the collection tree topology. Few other key distribution schemes Zhu et al. (2003); Liu and Ning (2003), ensure cryptographic primitives, but they do not address the specific threats involved after the sensor nodes are deployed in an unattended environment. Another key distribution scheme Dai and Xu (2010) suited for WSN ensures resilience in the event of node capture. These schemes do not address the soft security issues in sensor networks.

The statistical model proposed for Dynamic Source Routing (DSR) Pirzada et al. (2006) captures neighbor's misbehavior in DSR and presents mitigation techniques for dependable routing. Another scheme suited for secure routing over reactive protocols Yu et al. (2009) proposes a key distribution scheme and a statistical trust model for secure routing. Similarly, a scalable model Velloso et al. (2010) based on relationship maturity and recommendation trust is proposed for evaluating trust. But, these schemes are not suitable for resource constraint sensor networks. The trust metrics in the proposed trust model identify different types of misbehaviors by the compromised neighbors in BCP and ensures optimal throughput in the presence of adversaries. Since routing is done on per packet basis in BCP, the recommendation trust cannot be used as an evaluation metric for judging the trustworthiness of a node.

In all the above mentioned schemes, the evidence is mostly based on successful and failed interactions with the neighbors. Using VAR trust model in BCP, misbehaving neighbors that launch multiple types of security attacks are easily identified and trustworthy alternate routes are taken to forward the packets.

2.1 Overview of BCP

Collection protocols in sensor networks are used to gather data from multiple sources to a single sink or multiple sinks. BCP is based on dynamic backpressure routing and it is implemented and tested on IEEE 802.15.4 Tmote Sky motes. The backpressure routing algorithm has its roots in Utility Optimal Lyapunov Networking Neely (2009). It comes under the class of utility-optimal algorithms. It was shown that Backpressure algorithm when combined with LIFO queuing discipline is able to achieve near-optimal utility-delay tradeoff Huang et al. (2011). Other variants like backpressure with adaptive redundancy (BWAR) Alresaini et al. (2012), to reduce the delay under low network load conditions were developed for delay tolerant networks.

In backpressure routing, a queue gradient is developed with the generation of packets from source, which decreases from source to the sink. The collection cost, measured in transmissions to the sink, and network stability are encoded in the queue backlogs. Using this information, routing decisions are made on a per packet basis.

One of the key features of BCP is the per-packet collection cost, which is captured by using the ETX Couto et al. (2003) penalty function. Next, the queuing discipline of packets in each node is Last-In-First-Out (LIFO), which decreases the end-to-end packet delay, traditionally large in backpressure algorithms. There is 98% delay reduction for moderate source rates.

Next, the concept of floating queues ensures that BCP can scale well to large networks. The nodes which are furthest from the sink do not suffer from queue saturation, due to the presence of virtual queues. The BCP floating queue lies on top of this virtual queue, which does not store any useful data. The protocol adapts well to the situations when the floating queue is full/empty, by increasing/decreasing the size of the virtual queue.

Finally, the queue backlog information of neighbors is collected from the BCP packet headers by snooping. To reduce the processor load on snooping, a five-packet snoop queue is attached to the snoop interface. The BCP packets get dropped in this queue thereby reducing the processor overload.

The routing and forwarding decisions are made by each node on a per-packet basis. Every node i computes the backpressure weight, $w_{i,j}$ of all j neighbors by Equation (1).

$$w_{ij} = (\Delta Q_{ij} - V\overline{ETX}_{i\rightarrow j}).\overline{R}_{i\rightarrow j} \quad (1)$$

where ΔQ_{ij} refers to the difference in transmission queue size between the node i and its neighbor j, V is a constant to weigh between backpressure and ETX minimization, $\overline{ETX}_{i\rightarrow j}$ is the link usage penalty, $\overline{R}_{i\rightarrow j}$ refers to the estimated link rate. The node picks the neighbor with the highest $w_{i,j}$ as the next hop for routing the packet.

3 Modifications in BCP to incorporate Trust

3.1 Motivation behind the VAR trust Model

In the context of security, trust is defined as the staunch belief over the competence of another entity, to perform a set of actions reliably and sincerely. The VAR trust model is a node-centric trust model similar to the trust models in Pirzada et al. (2006); Theodorakopoulos and Baras (2006) but without any recommendation trust. Every node individually evaluates its neighbor and makes a judgment on the trust value, which is more realistic and similar to the trust in social networks. The novelty in our model lies in capturing the different behavior of the neighbor nodes, like willingness to participate in routing, data forwarding, sincerity of the neighbor nodes in forwarding the data without modification, etc. These behaviors are termed as *trust metrics* in our model and are stored as a *trust vector* for each neighboring node.

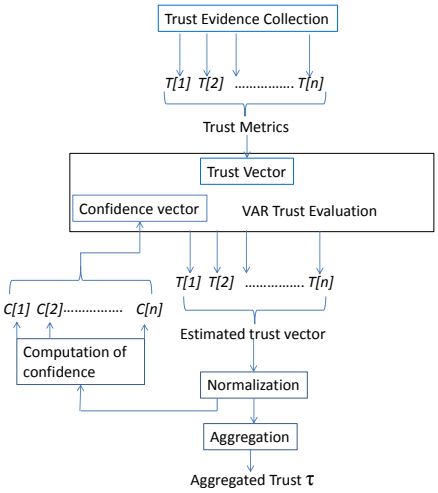


Figure 1: Trust Computation Procedure

In general, a VAR is a n-equation, n-variable linear model in which each variable depends on its own lagged values, plus current and past values of the remaining *n-1* variables. Since they exhibit strong correlation between them and their past time-lagged values, they are called as *endogenous* variables. A simple two variable VAR model with a time lag $p = 1$ is given in Equation (2).

$$\begin{aligned} T_t[1] &= R_{11}T_{t-1}[1] + R_{12}T_{t-1}[2] + \epsilon_{1,t} \\ T_t[2] &= R_{21}T_{t-1}[1] + R_{22}T_{t-1}[2] + \epsilon_{2,t} \end{aligned} \quad (2)$$

where $R_{i,j}$ are the regression coefficients and ϵs are the error terms. The following are the assumptions made while applying the VAR model for prediction of neighboring trust. The trust metrics at different instants of time are dependent on each other and their past time lagged values. Hence, they are represented as multivariate time series. This is due to the fact that multiple attacks can be launched by a compromised neighbor at different time intervals. The time series is assumed to be stationary. The error terms are not autocorrelated.

Our notion of trust relies on direct observations of neighboring nodes. Hence, there is no propagation of trust and no recommendations of neighbor trust. This assumption is more suitable for implementing our trust in BCP as the routing decision is on per packet basis. The process of VAR trust computation is shown in Figure 1 and details of trust computation in BCP are presented in the next section.

3.2 Trust Metrics for BCP

In wireless sensor networks with packet transmission using BCP, the malicious nodes are assumed to indulge

in four types of attacks as mentioned in Section 1. To capture the neighbor's behavior related to these attacks, five trust metrics are required in the VAR model.

The trust metrics to monitor the successful forwarding of data packets, without content modification are shown in Equations (3) and (4) respectively.

$$T[1] = \frac{N_F}{T_F} \tag{3}$$

$$T[2] = \frac{N_{CF}}{N_F} \tag{4}$$

where N_F is the number of packets forwarded to the neighbor, T_F is the total number of packets forwarded to the neighbor and N_{CF} is the number of packets forwarded by the neighbor without content modification.

A compromised node involved in no/selective forwarding attack and header modification attack, can be promiscuously detected by the neighbors and its behavior captured in the VAR trust metric T[1] of the evaluating neighbor. Similarly, misbehavior of the neighboring node with regard to content modification is captured in T[2].

If a compromised node indulges in false queue size advertisements, its behavior is captured in trust metrics T[3] and T[4] and they are given in Equations (5) and (6). The queue size advertised by a cooperating node should be in the range $[Q_{min},\ Q_{max}]$. The advertised queue values outside this range are considered as false advertisements and they are appropriately captured in these trust metrics.

$$T[3] = 1 - \frac{N(Q_{min})}{N(Q)} \tag{5}$$

$$T[4] = 1 - \frac{N(Q_{max})}{N(Q)} \tag{6}$$

where $N(Q_{min})$ refers to the number of occurrences wherein the advertised queue size by neighbor is $\leq Q_{min}$, $N(Q_{max})$ refers to the number of occurrences wherein the advertised queue size is $\geq Q_{max}$ and $N(Q)$ denotes the total number of queue advertisements. A malicious node's bad impact is restricted by the number of occurrences of false queue advertisements, after which it loses the trust of its neighbors and it may not be chosen as a good neighbor for data forwarding. Moreover, the regression coefficients detect those malicious nodes indulging in more than one type of attack in a specified time interval. For instance, a node may attract traffic by advertising low queue size and later on, indulge in blackhole attack. These malicious behaviors are easily captured as the correlation between the trust metrics at different time units are reflected in the regression coefficients of the VAR model.

To identify a neighbor indulging in malicious misrouting, the trust metric T[5] is used as shown in Equation (7).

$$T[5] = \frac{N_R}{T_R} \tag{7}$$

where N_R is the number of matched routing decisions made by the neighbor with the evaluating entity and T_R is the total number of routing decisions made for the neighbor.

3.3 VAR Equation for BCP

The selection of the VAR model order plays a critical role in evaluating the accuracy of the model estimation. The time lag coefficient p is the order of the VAR model and Akaike's Information Criterion (AIC) Priestley (1981) is used to determine the order that best fits the model. Sample observations of neighbor's trust metrics are collected and AIC values are computed for different orders of the VAR model as shown in Figure 2. The minimum value of the AIC indicates the order that best fits the model and it was found to be 2.

The VAR trust model of order 2 for BCP is represented in simple linear regression form as shown in Equation (8). The regression coefficient matrices are estimated by Ordinary Least Squares (OLS) technique Gujarati (2003) using MATLAB simulations with a sample data size of 250 observations.

$$\begin{aligned}
\hat{T}_{y(t)}[1] &= \alpha C_y[1] + \sum_{x=1}^{5} R'_{1x} T_{y(t-1)}[x] + \sum_{x=1}^{5} R''_{1x} T_{y(t-2)}[x] \\
&\quad + \epsilon[1] \\
\hat{T}_{y(t)}[2] &= \alpha C_y[2] + \sum_{x=1}^{5} R'_{2x} T_{y(t-1)}[x] + \sum_{x=1}^{5} R''_{2x} T_{y(t-2)}[x] \\
&\quad + \epsilon[2] \\
\hat{T}_{y(t)}[3] &= \alpha C_y[3] + \sum_{x=1}^{5} R'_{3x} T_{y(t-1)}[x] + \sum_{x=1}^{5} R''_{3x} T_{y(t-2)}[x] \\
&\quad + \epsilon[3] \\
\hat{T}_{y(t)}[4] &= \alpha C_y[4] + \sum_{x=1}^{5} R'_{4x} T_{y(t-1)}[x] + \sum_{x=1}^{5} R''_{4x} T_{y(t-2)}[x] \\
&\quad + \epsilon[4] \\
\hat{T}_{y(t)}[5] &= \alpha C_y[5] + \sum_{x=1}^{5} R'_{5x} T_{y(t-1)}[x] + \sum_{x=1}^{5} R''_{5x} T_{y(t-2)}[x] \\
&\quad + \epsilon[5]
\end{aligned} \tag{8}$$

where α is a parameter to weigh between computed trust and confidence, $[\hat{T}_{y(t)}]_{5\text{x}1}$ represents the five estimated trust metrics for neighbor y, $[C_{y(t)}]_{5\text{x}1}$ denotes the confidence vector corresponding to these trust metrics, $[R']_{5\text{x}5}$ refers to the regression coefficient matrix for the first time lag, $[R'']_{5\text{x}5}$ denotes the regression coefficient matrix for the second time lag and $[\epsilon]_{5\text{x}1}$ refers to the error vector. The estimated trust is normalized using Equation (9) and categorized into trust ranges as shown in Table 1.

$$NT_y = \frac{2(T_{y(t)} - A_{min})}{A_{max} - A_{min}} - 1 \tag{9}$$

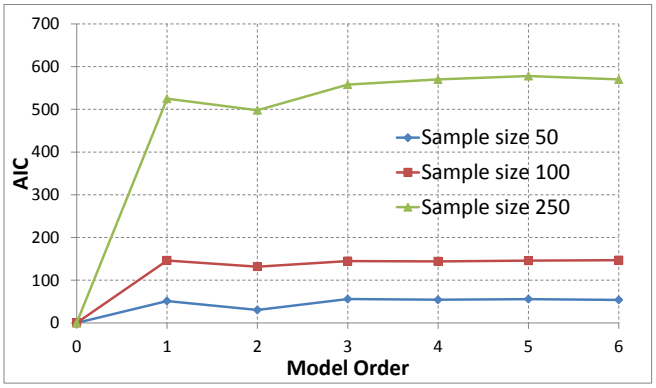


Figure 2: AIC values showing the best fit model order

Table 1 Trust Levels

Trust Ranges	Meaning	Weights for computing Confidence
$T < 0$	Distrust	-0.1
$T = 0$	Ignorance	0.2
$0 < T < 0.5$	Marginal	0.4
$0.5 < T < 1$	Complete	0.5

where NT is the normalized trust vector, A_{min} is the minimum possible trust and A_{max} is the maximum possible trust. The normalized trust value is used to compute the confidence value of the neighboring node in the next time slot.

The confidence value of the neighboring node y, for the n^{th} trust metric, at time t, is given by Equation (10).

$$C_{y(t)}[n] = \frac{\sum_{x=1}^{4} N_x w_x}{\mu} \tag{10}$$

where x is a trust level as shown in Table 1, N_x is the number of times the neighboring node has acquired the trust level x, w_x is the weight associated with this trust level and μ is the maximum length of association between the neighboring nodes. The weights are computed by numerous trial evaluations and the values shown in Table 1 are found to model the implications of trust in social networks. Higher trust levels gain more weightage than the lower ones. Distrust is assigned a negative weight.

The aggregated trust τ for a neighboring node is computed by weighted averaging of trust metrics as in Equation (11).

$$\tau = \sum_{j=1}^{n} w_j T(j) \tag{11}$$

where w_j are equal positive weights associated with each trust metric and $\sum_{j=1}^{n} w_j = 1$

3.4 Trusted Routing in BCP

Initially, the neighbor nodes will have unknown trust with each other. Hence, their default trust values against each other will be zero. As time progresses, the neighbors will start evaluating each other's behavior by listening to their transmissions promiscuously. The snoop queue attached to the snoop interface of BCP is increased to 8 and no additional snooping interfaces are required for the trust model.

In steady state, the trust values of the neighboring node will play a significant role in making routing decisions of the packets along with backpressure weights. By this way, good neighbors will earn higher trust and confidence. The malicious nodes will be easily detected by their neighbors. For detecting content modification attacks, we have used the SHA-1 source code from TinyECC Liu and Ning (2008) which computes the hash of header and data for a BCP packet. Before forwarding a packet, the sender node computes the hash, H(Origin,Origin-Sequence-No,Data) and stores it locally. Later, it snoops its neighbor's transmission and re-computes the hash. If the hash value matches, the node increases the corresponding trust metric of its neighbor.

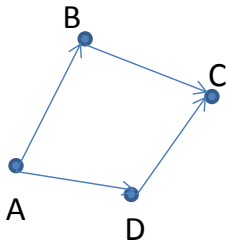
In order to integrate the trust value into backpressure routing, the weight computation procedure is modified to include the trust of the neighboring nodes. The modified weight calculation for BCP is given in Equation (12).

$$w_{ij} = (\Delta Q_{ij} - V\overline{ETX}_{i\to j} + \beta\tau_j).\overline{R}_{i\to j} \tag{12}$$

where τ_j is the aggregated trust of the neighboring node and β is the weighing factor between backpressure weight and trust weight. According to the backpressure routing policy, each node i computes the backpresssure weight, $w_{i,j}$ for all its neighbors and chooses that neighbor with the highest backpressure weight for forwarding the packets. The integration of VAR into backpressure routing is shown in Algorithm 1.

The key challenge lies in choosing an appropriate value for β, wherein too low a value will downplay the significance of trust, and too high a value will outweigh the advantages of backpressure routing, When β value is high, the trusted neighbors will have an overflowing floating queue, leading to many packet drops. The ETX penalty cost included in the backpressure weight estimation will become negligible, which will lead to more transmissions over lossy wireless links, leading to reduction in throughput and efficiency of the sensor network. Hence, the value of β should be just enough to aid in choosing the best neighbor, given their queue differentials and link estimation costs for backpressure routing. Our empirical results in Figure 3 (c) with 40 sensor motes in Tutornet testbed shows the optimum value of β at 2.

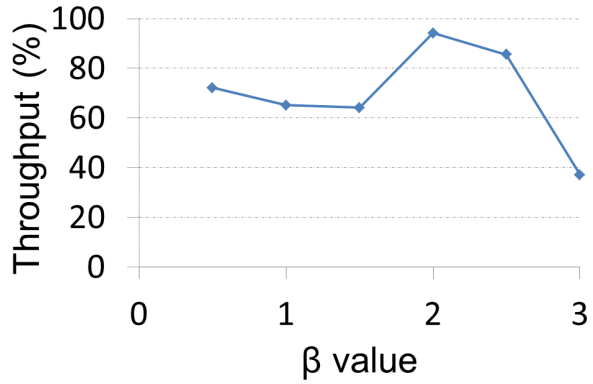
In Figures 3 (a) and (b), we illustrate the notion of trusted routing in BCP with VAR, using an intuitive example of a four node network. For simplicity, we have considered the trust metrics T[1] and T[2] alone for every neighbor. We assume that the backpressure gradient from $A \Rightarrow B$ and $A \Rightarrow C$ is formed so that node A transmits 2 packets every 2 secs, alternatively to node B and node D. Let the trust update interval Δ be 12 secs.



(a)

Trust Metrics	t=2	t=4	t=6	t=8	t=10	t=12
	B	D	B	D	B	D
T[1]	$\frac{0}{0}=0$	$\frac{2}{2}=1$	$\frac{2}{4}=0.5$	$\frac{3}{4}=0.75$	$\frac{4}{6}=0.66$	$\frac{5}{6}=0.83$
T[2]	$\frac{0}{0}=0$	$\frac{2}{2}=1$	$\frac{0}{2}=0$	$\frac{3}{3}=1.0$	$\frac{2}{4}=0.5$	$\frac{0}{2}=0.0$

(b)



(c)

Trust Variables	Node B	Node D
$T_y(t)$	$\begin{bmatrix}1.32\\1\end{bmatrix}$	$\begin{bmatrix}1.58\\2\end{bmatrix}$
NT	$\begin{bmatrix}0.32\\0\end{bmatrix}$	$\begin{bmatrix}0.58\\1\end{bmatrix}$
τ	0.15	0.79

(d)

Figure 3: (a) A four node scenario using BCP with VAR where node A alternatively transmits 2 packets, every two time units to B and D. (b) Trust metrics as computed by node A for its neighbors B and D. Neighbor B is a compromised node and drops both the packets at t= 2. At t=6, node B indulges in packet modification attack. At t=8, one packet is not received by node D due to link failure. (c) Empirical results showing maximum throughput at $\beta = 2$ (d) Trust evaluations done by node A for its neighbors

Initialization;
Initialize the confidence vector, error vector and regression coefficients;
if *trust update interval* = Δ **then**
  for *every neighbor* y **do**
    Compute the trust metrics $T_y[1]...T_y[5]$ using Equations 3 - 7;
    Estimate the trust, $\hat{T}_{y(t)}[1..5]$ for this neighbor using
      Equation 8 with confidence vector $C_{y(\Delta-1)}[1..5]$;
    Normalize the trust vector into the range [-1,1] using Equation 9;
    Compute the confidence vector $\hat{C}_{y(t)}[1..5]$ using Equation 10;
    Compute aggregated trust τ using Equation 11;
    Compute the backpressure weight $w_{i,y}$ for this neighbor link
      using Equation 12;
  end
end
Routing decision;
Choose the link with maximum positive backpressure weight $w_{i,j}$ as the best trustworthy neighbor for forwarding the packets;

Algorithm 1: VAR trust computation with backpressure routing

Assuming zero confidence and error vectors and unit regression coefficient matrices, the estimated trust of the nodes B and D are computed using Equation 8. These values are normalized in the range [-1,1] using Equation 9 with $A_{max} = 2$ and $A_{min} = 0$. Assuming equal weights for T[1] and T[2], the aggregated trust τ is computed using Equation 11 and their values are shown in Figure 3 (d). With all other backpressure routing parameters being equal for the weight computation process, node D will have higher backpressure weight when compared to node B. Hence, node D will be chosen as a trustworthy neighbor for forwarding the data packets in the next trust update interval.

4 Experimental setup and Performance Analysis over BCP

The VAR model is integrated with BCP and the proposed modifications in BCP are implemented in Tutornet, a IEEE 802.15.4 wireless sensor network testbed comprising of TmoteSky motes. Table 2 describes the VAR trust and experimental setup parameters. Figure 4 represents the topologies used for experimentations and their average node degrees are 3.36 and 5.76 respectively. They serve as a representative network to evaluate the memory and computational complexity of the VAR trust model over BCP. BCP-VAR refers to the modified BCP with the VAR trust model.

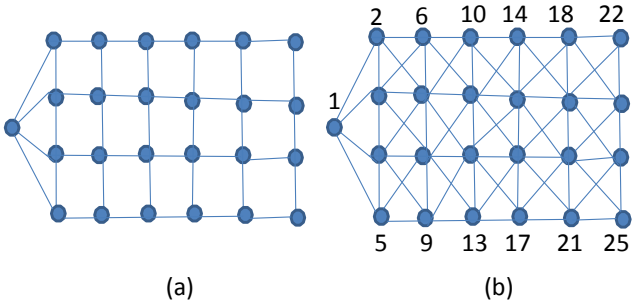


Figure 4: (a) Mesh topology with average node degree 3.36. (b) Mesh Topology with node id column ordered and average node degree 5.76

Table 2 Experimental Setup and VAR trust parameters

Parameters	Values
Number of motes	25
Transmit Power	-18 dBm
Packet Size	34 bytes
Inter packet arrival time	Exponential
Constant used in BCP weight computation, V	2
Number of trust metrics (n)	5
VAR time lag (p)	2
weight for T[1]	0.2
weight for T[2]	0.2
weight for T[3]	0.2
weight for T[4]	0.2
weight for T[5]	0.2
α	0.5
β	2
Q_{min}	1
Q_{max}	10
μ	30 minutes
Source Rate	variable - 0.25 to 2 pkts/sec (pps)
Trust update interval (Δ)	20 secs

The regression co-efficients of the VAR trust model for BCP are computed using MATLAB simulations for more than 1000 observations. Hence, the time complexity of VAR trust model with BCP depends on the number of trust metrics (n) and the number of neighbors (y) and it is found to be $O(ny)$. The code size for the BCP-VAR including the test application is found to be around 29KB, which is larger than BCP without trust by approximately 6KB.

4.1 Performance Analysis of BCP and BCP-VAR without any attacks

Firstly, the performance of these two protocols is studied under normal circumstances where no compromised nodes are present in the network. The efficiency of BCP and BCP-VAR is evaluated with three metrics:

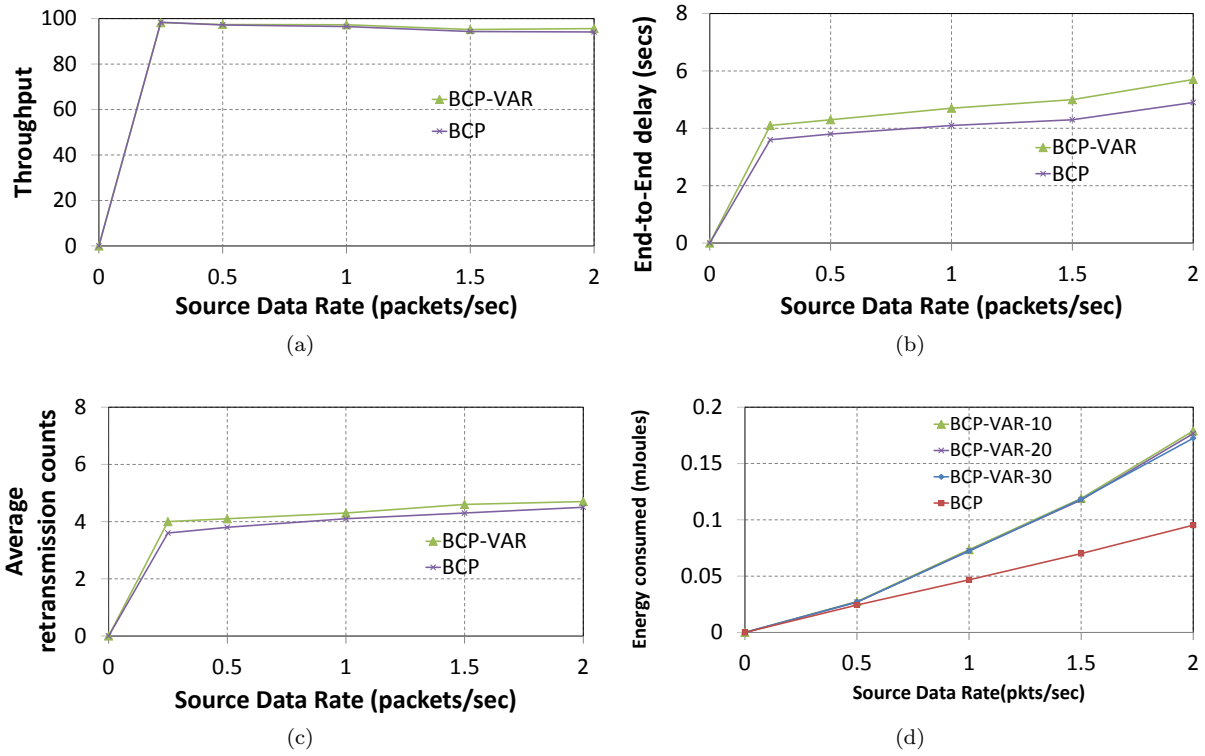


Figure 5: Performance Comparison of BCP and BCP-VAR without any compromised nodes in the network. (a) Throughput of the network (b) Average End-to-end delay experienced by the packets (c) Average retransmission counts experienced by the packets (b) Average Energy consumed by the sensor nodes at different trust update intervals(Δ) at 95% confidence intervals.

throughput, end to end delay experienced by packets and the average retransmission counts. The sensor nodes are arranged in topology as shown in Figure 4 (b).

Figure 5 (a) shows that BCP-VAR is able to offer the same throughput as BCP at different source data rates. But, the end-to-end delay experienced by the packets in BCP-VAR is 0.5 seconds higher than in BCP which is shown in Figure 5 (b). The increase in delay is attributed to the execution of trust computation logic in the individual sensors of the network. Next, the average retransmission counts in the network is analyzed for BCP and BCP-VAR. The retransmission count is a metric which signifies the number of attempts made by the node to forward a data packet to its designated neighbor before making a successful transmission. Hence, it is desirable to minimize the average retransmission counts which indicates the stability of links in the network. In Figure 5 (c), it was found that the average retransmission counts experienced by the packets in BCP-VAR is very close to that in BCP.

In Figure 5 (d), the average energy consumed by the sensor devices is analyzed in the network for a random topology at different trust update intervals. BCP-VAR shows a near linear increase in the average energy consumption when compared to the sub-linear increase in BCP against increments in source data rate by 0.5 pkts/sec (pps). This is predominantly due to the energy spent on neighbor packet snooping. It is also evident that the average energy consumption in BCP-VAR does not vary much with the variations in trust update intervals. Hence, the trust computation algorithm does not add much overhead to energy consumption.

4.2 Performance of BCP-VAR in the presence of no forwarding nodes

Next, we consider a scenario where the compromised nodes partially drop few data packets. The sensor motes are arranged in topology as shown in Figure 4 (a). Few sensor motes are chosen at random to behave as malicious nodes that drop data packets. Figure 6 shows the throughput of the network under BCP and BCP-VAR amidst no forwarding nodes at source data rates of 0.25 pps and 1.25 pps. In the presence of 40% no forwarding nodes, BCP-VAR offers around 75% packet delivery ratio. The average end-to-end delay experienced by the packets generated from all sources for BCP-VAR was analyzed at 95% confidence intervals for source data rates of 0.25 pps and 1.25 pps. It was found be greater by 0.5 seconds, amidst 40% compromised sensors, as shown in Figure 7. This is the overhead associated with the trust computation algorithm.

The average retransmission counts experienced by the packets is presented in Figure 8 (a). This shows that BCP-VAR, amidst 40% malicious nodes increases the average retransmission attempts by 2 at source data rates of 1.25 pps. This is due to the fact that the trustworthy neighbors are selected to be forwarders, in

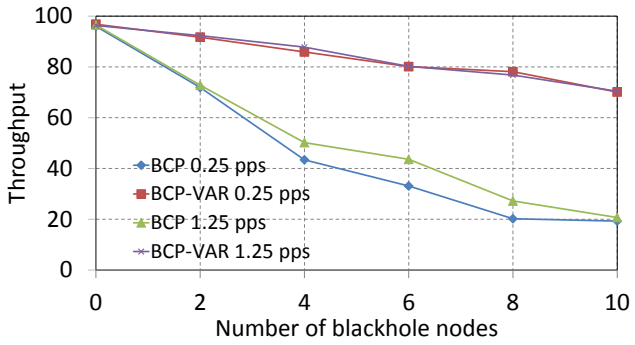


Figure 6: Throughput of the network for BCP and BCP-VAR in the presence of blackhole nodes

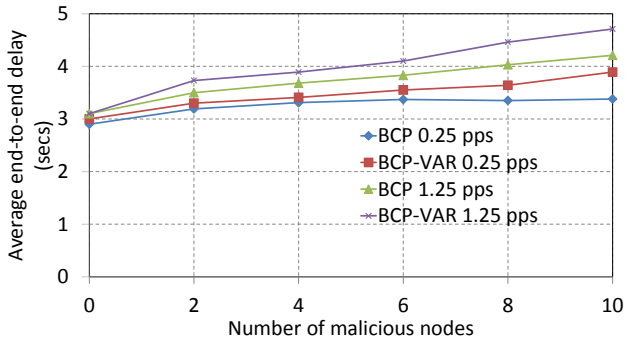


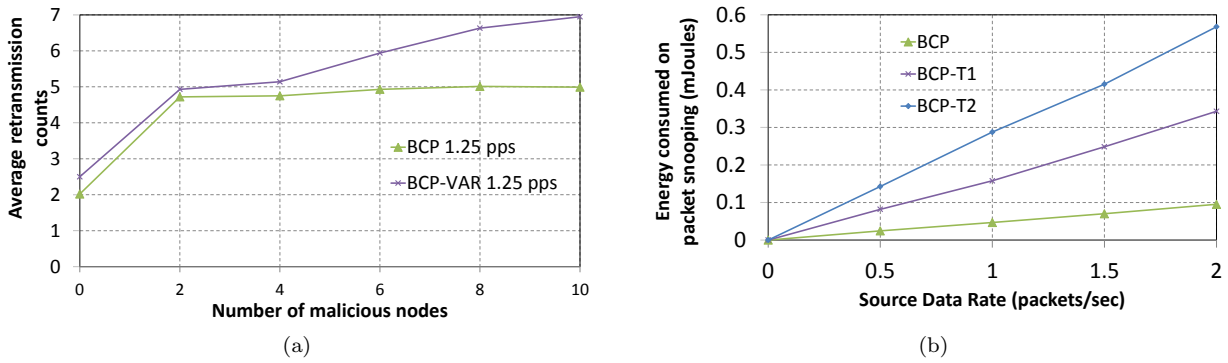
Figure 7: Average End-to-end delay experienced by packets at 95% confidence intervals

spite of high backpressure, leading to packet drops and retransmission.

The predominant source of energy consumption in wireless sensor networks is the radio when it is transmitting or receiving packets. In BCP-VAR, there are no additional trust related packet transmissions. With all other energy consumption sources being the same, BCP-VAR spends extra energy on snooping its neighbor's packets. Hence, the energy consumed on neighbor packet snooping largely depends on the number of neighbors in the vicinity. Figure 8 (b) shows the average energy consumed by sensors, on packet snooping arranged in topologies, as shown in Figures 4 (a) and (b) respectively, with average node degrees 3.36 and 5.76 . It was found that the energy consumed on packet snooping for BCP-VAR is approximately average node degree $\times$ the energy consumed on packet reception for normal BCP.

4.3 Performance of BCP-VAR with malicious misrouting

To evaluate the performance of BCP-VAR amidst malicious nodes indulging in packet misrouting, the sensor motes are arranged in a mesh topology as shown in Figure 4 (b). Eight sensor motes are chosen randomly to misroute data packets to random neighbors. Remaining sensors, except sink, are sources generating data packets at 1.0 pps.



(a) (b)

Figure 8: (a) Average retransmission counts experienced by the packets (b) Average Energy consumed on packet snooping for the topologies in Figure 4 at 95% confidence intervals. BCP-T1 refers to BCP-VAR using topology in Figure 4 (a). BCP-T2 refers to BCP-VAR using topology in Figure 4 (b)

In the mesh topology considered for experimentation, there are ten interior motes which are surrounded by other motes. These motes can judge the routing decisions of the neighboring motes present in the border of the network with full neighbor visibility. The false positive and false negative rates along with the detection rates of individual nodes in the sensor network are shown in Figure 9. It can be seen that the border nodes can evaluate the routing decisions of interior nodes with partial visibility whereas the interior nodes can better judge their neighbors in the border of the network. In spite of this limitation, it was found in Figure 10 (a) that the packet loss percentage is greatly minimized in BCP-VAR trust to 30% at source data rates of 2 pps. Figure 10 (b) shows that the packet delay for BCP-VAR is also minimal in the presence of malicious misrouters at 4.9 sec for source data rates of 2 pps, whereas in normal BCP, the packets take a circuitous path to the sink, incurring a delay of 7.8 secs.

4.4 Performance of BCP-VAR amidst header/content modification attacks

Compromised nodes which modify the header or data of the message are introduced into the network with random topology and the performance of BCP-VAR is shown in Figure 10 (c) for source data rate of 1.0 pps. In the presence of 40% malicious nodes, the protocol performs reasonably well, with less than 15% of modified packets arriving at the sink. Few modified packets still arrive at the sink because the nodes learn about data modification attacks by promiscuously listening to its malicious neighbor over a period of time. In due course, the node lowers its trust for their malicious neighbors.

4.5 Performance of BCP-VAR with nodes advertising false queue sizes

The performance of BCP-VAR is tested by varying the number of sensor devices which advertise low queue sizes in a network with random topology at source data rates of 1.0 pps. After attracting the traffic, these sensors are made to drop the data packets. The neighboring nodes quickly sense the behavior of these malicious nodes and lower their trust and confidence values. The rest of the packets are forwarded through trustworthy neighbors and the throughput of the network is maintained at 75%, in the presence of 40% compromised sensors as shown in Figure 10 (d).

Table 3 Simulation parameters

Parameter	Value
Simulation Area	$800 * 800m^2$
Transmit Power	-18dBm
Number of devices	≈ 300
Inter-packet arrival time	Exponential
Data Packet Size	27 bytes (constant)
Data Rate	250 Kbps
Number of trust metrics considered (n)	6
VAR time lag (p)	2
Epoch duration (μ)	30 minutes
Trust metric weight	0.17 (equal weights)
Parameter to weight between trust and confidence (α)	0.5

5 Performance Analysis of the VAR trust model

The performance analysis of the VAR model is compared with SRAC and SLSP/SMT in an IEEE 802.15.4 network scenario and the results are presented in this section. Simulations are carried out using OPNET Modeler for wireless networks. The link encryption key size for SRAC is 256 bits. The nodes are deployed in a random manner in an area of 800 X 800 m^2. Other simulation setup parameters and the VAR trust parameters are listed in Table 3.

First, we consider the average time taken to detect malicious behavior and it is compared for VAR, SRAC

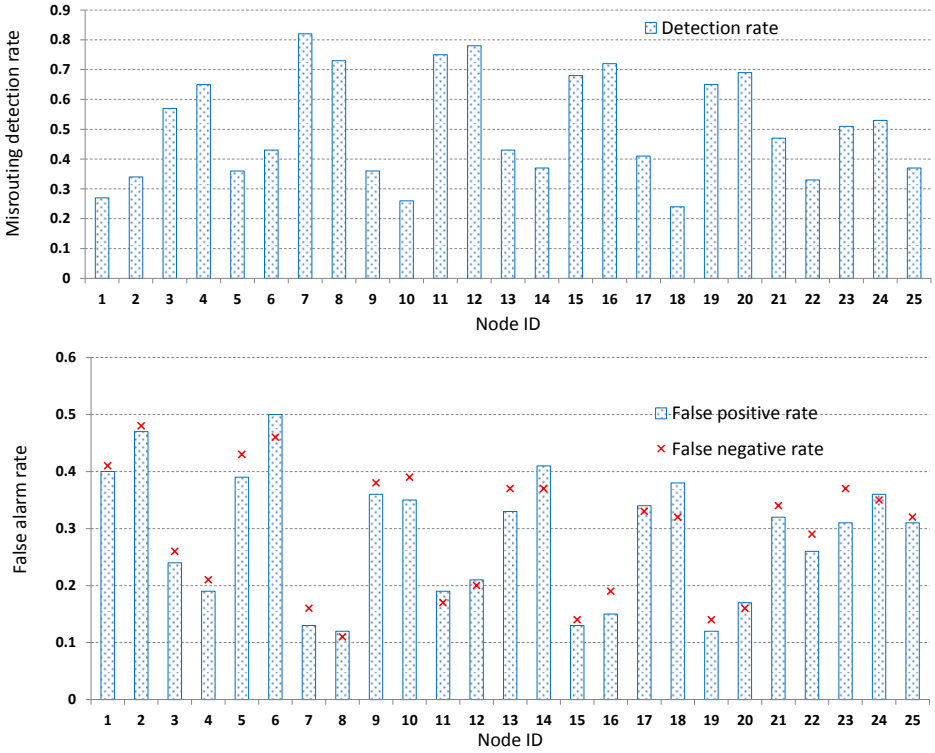


Figure 9: The malicious misrouting detection rates (top) and False alarm rates (bottom) raised by individual motes in a scenario where the motes 3,6,8,13,14,19,21,23 are malicious misrouters in the mesh topology of Figure 4 (b)

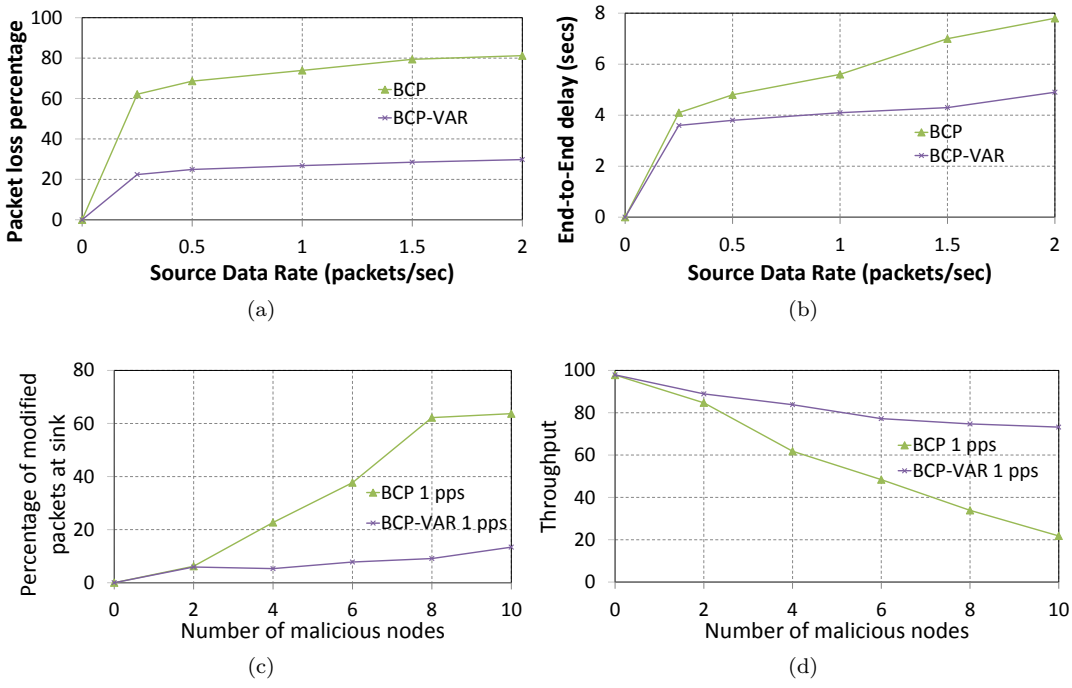


Figure 10: (a) Packet loss-percentage in the presence of malicious misrouters (b) End-to-end delay incurred by the packets in the presence of 32% malicious misrouting sensor motes (c) Percentage of modified packets arriving at sink in the presence of malicious sensors launching content modification attacks (d) Throughput of the network in the presence of malicious sensors advertising low queue sizes

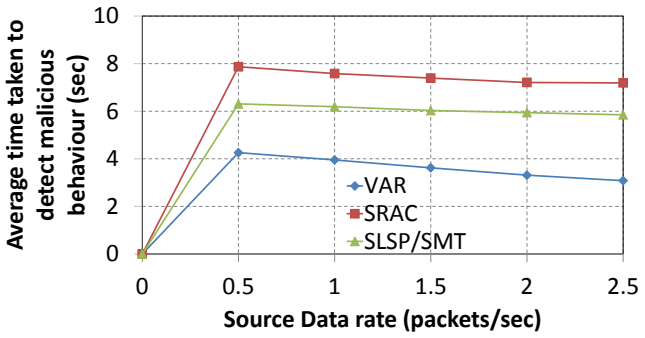


Figure 11: Average time taken to detect malicious behavior at 95% confidence intervals varying the source data rate

and SLSP/SMT trust models in a scenario where multiple attacks are launched at different instants of time. Figure 11 shows the performance in a network where 40% of the malicious nodes launch different attacks like no forwarding attacks, flooding attacks, content modification attacks, etc. It can be seen that the VAR trust model takes minimum time to detect these malicious behaviors, due to the presence of a strong evidence collection phase in the trust evaluation process. It can also be seen that this detection time is further reduced when the source data rate is increased. This can be attributed to the fact that with more number of evidences in a given time interval, the trust metrics will be able to quickly identify the malicious behavior of the neighboring nodes.

Next, the throughput of the network is measured for these trust models by varying the percentage of malicious nodes, which launch multiple attacks at different simulation time intervals. Figure 12 shows that SRAC is not equipped with handling multiple attacks and hence, the throughput of the network reduces to 42% in the presence of 50% malicious nodes. Similarly, in SLSP / SMT, the path survival probabilities are computed for every path from source to the sink. Both SRAC and SLSP / SMT are efficiently able to handle attacks where the adversaries dropped all or part of the data packet transmission. But, they are unable to detect those adversaries that launch other attacks like corrupting the data packet or data flooding attack. It was found that VAR model was able to rightly identify the compromised nodes at a very early stage and choose those paths with trustworthy nodes to reach the sink. Hence, with VAR model, 73% throughput was achieved in the presence of 50% malicious nodes.

Finally, we examine the energy consumption of the nodes using various trust models under consideration and the results are shown in Figure 13. It was found that the VAR trust model, when compared to SRAC and SLSP / SMT offers minimum average energy consumption at different source data rates. This is achieved in spite of the additional energy spent on neighbor packet snooping. The energy consumed

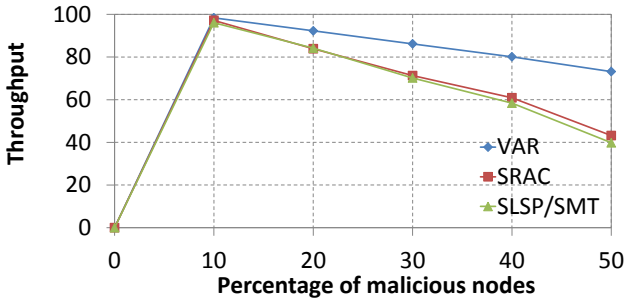


Figure 12: Throughput of the network varying the percentage of malicious nodes

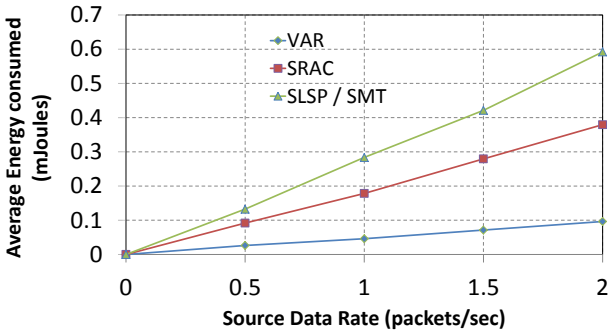


Figure 13: Average Energy Consumption for different trust models

in nodes using SRAC is due to the computational complexity involved in extensive use of cryptographic algorithms and authentication mechanisms employing digital signatures. The encryption and decryption procedures in every link is the cause of this increased overhead. In SLSP / SMT, the increased energy consumption is due to the increase in routing packet size for every link state updates by using public key infrastructure and the redundancy of data packet transmission in multiple paths to ensure reliable communication.

6 Conclusion

We have incorporated the VAR trust model into the state of the art dynamic backpressure routing protocol and shown that this model is also suited for resource constraint wireless networks. By adjusting the trust parameters in the backpressure weight computation, we have shown that the performance can be greatly improved in the presence of malicious nodes. We have also demonstrated that trust mechanisms can indeed be easily incorporated into a resource-constrained sensor network by implementing and testing our code in a real sensor network testbed. We found that with VAR trust, the nodes in a wireless sensor network are able to quickly learn the malicious behavior of any neighboring node present in the network. Accordingly, they chose an alternate trustworthy path for routing the packets.

This leads to an overall increase in the throughput performance of the network. Our simulation results show that in the presence of 50% malicious nodes that launch multiple attacks, the VAR trust model outperforms other existing trust models. Some of the interesting extensions to this work include experimentation with wireless sensor devices with multimode radios so as to enhance the snoop capabilities and integration of the VAR trust model to other routing mechanisms in wireless network applications.

Acknowledgements

This work has been supported in part by the U.S. National Science Foundation, under award 1049541. The authors would like to thank the Tutornet development and maintenance team, Univ. of Southern California, Los Angeles for granting access to carry out experimentation in the testbed. The authors acknowledge the support extended by SRM University for providing necessary infrastructure facilities.

References

Alresaini, M., Sathiamoorthy, M., Krishnamachari, B., and Neely, M. J. (2012). Backpressure with adaptive redundancy BWAR. In *IEEE INFOCOM 2012*.

Anderson, C. W., Stolz, E. A., and Shamsunder, S. (1998). Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks. *IEEE Transactions on Bio-Medical Engineering*, 45(3):277–286.

Araujo, A., Blesa, J., Romero, E., and Villanueva, D. (2012). Security in cognitive wireless sensor networks. challenges and open problems. *EURASIP Journal on Wireless Communications and Networking*, 2012:1–8.

Cerpa, A., Elson, J., Estrin, D., Girod, L., Hamilton, M., and Zhao, J. (2001). Habitat monitoring: Application driver for wireless communications technology. In *Proceedings of the Workshop on Data Communications in Latin America and the Caribbean*.

Chong, C. Y. and Kumar, S. P. (2003). Sensor networks: Evolution, opportunities and challenges. In *Proceedings of the IEEE*.

Couto, D. S. J. D., Aguayo, D., Bicket, J., and Morris, R. (2003). A high throughput path metric for multihop wireless routing. In *ACM / IEEE MobiCom*.

Culler, D., Estrin, D., and Srivastava, M. (2004). Overview of sensor networks. *IEEE Computer*, 37(8):41–49.

Dai, H. and Xu, H. (2010). Key predistribution approach in wireless sensor networks using LU matrix. *IEEE Sensors Journal*, 10(8):1399–1409.

Edgar H. Callaway, J. (2004). *Wireless Sensor Networks: Architectures and Protocols*. Auerbach Publications.

Glisic, S. G. (2006). *Advanced Wireless Networks: 4G Technologies*. John Wily.

Gujarati, D. N. (2003). *Basic Econometrics*. McGraw-Hill/Irvin.

Huang, L., Moeller, S., Neely, M. J., and Krishnamachari, B. (2011). LIFO-backpressure achieves near optimal utility-delay tradeoff. In *WiOpt*.

Karlof, C., Sastry, N., and Wagner, D. (2004). TinySec: A link layer security architecture for wireless sensor networks. In *Proceedings of Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*.

Liu, A. and Ning, P. (2008). TinyECC: A Configurable library for Elliptic Curve Cryptography in Wireless Sensor Networks(ver.2.0). In *Proceedings of the 7th international conference on Information processing in sensor networks*.

Liu, D. and Ning, P. (2003). Establishing pairwise keys in distributed sensor networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*.

Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In *Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA02)*.

Moeller, S., Sridharan, A., Krishnamachari, B., and Ganawali, O. (2010). Routing without routes: Backpressure Collection Protocol. In *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks*.

Neely, M. J. (2009). Intelligent packet dropping for optimal energy-delay tradeoffs in wireless downlinks. *IEEE Transactions on Automatic Control*, 54(3):565–579.

Papadimitratos, P. and Haas, Z. J. (2003). Secure link state routing for mobile ad hoc networks. In *IEEE Wksp. Security and Assurance in Ad hoc Networks*.

Papadimitratos, P. and Haas, Z. J. (2006). Secure data communication in mobile ad hoc networks. *IEEE Journal of Selected Areas In Communications*, 24(2):343–356.

Perrig, A., Szewczyk, R., Wen, V., Culler, D., and Tygar, J. (2001). SPINS: Security protocols for sensor networks. In *Proceedings of ACM Mobile Computing and Networking (MobiCom)*.

Pirzada, A. A., Datta, A., and McDonald, C. (2006). Incorporating trust and reputation in the DSR protocol for dependable routing. *Elsevier Computer Communications*, 29(15):2806–2821.

Priestley, M. B. (1981). *Spectral Analysis and Time Series, Volume I and II*. Academic.

Qiu, J.-P. and Chen, L.-C. (2008). Trust management for semantic web. In *International Conference on Computer Science and Software Engineering*.

Rajeev, S., Ananda, A., Chan, M. C., and Ooi, W. T. (2006). *Mobile, Wireless, and Sensor Networks: Technology, Applications and Future Directions*. John Wiley.

Szewczyk, R., Mainwaring, A., Polastre, J., and Culler, D. (2004). An analysis of a large scale habitat monitoring application. In *Proceedings of Second ACM Conference on Embedded Networked Sensor Systems*.

Theodorakopoulos, G. and Baras, J. S. (2006). On Trust Models and Evaluation Metrics for Mobile Ad hoc Networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328.

Traynor, P., Kumar, R., Choi, H., Cao, G., Zhu, S., and Porta, T. L. (2007). Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing*, 6(6):663–677.

Velloso, P. B., Laufer, R. P., de O. Cunha, D., Duarte, O. C. M. B., and Pujolle, G. (2010). Trust Management in Mobile Ad Hoc Networks using a Scalable Maturity-Based Model. *IEEE Transactions on Network and Service Management*, 7(3):172–185.

Venkataraman, R., Pushpalatha, M., and Rao, T. R. (2012a). Implementation of a regression based trust model in a wireless ad hoc testbed. *Defence Science Journal*, 62(3):167–173.

Venkataraman, R., Pushpalatha, M., and Rao, T. R. (2012b). Regression based trust model for mobile ad hoc networks. *IET Information Security*, 6(3):131–140.

Xu, N., Rangwala, S., Chintalapudi, K. K., Ganesan, D., Broad, A., Govindan, R., and Estrin, D. (2004). A wireless sensor network for structural monitoring. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*.

Yu, M., Zhou, M., and Su, W. (2009). A Secure Routing Protocol Against Byzantine Attacks for MANETs in Adversarial Environments. *IEEE Transactions on Vehicular Technology*, 58(1):449–460.

Zaidi, Z. R. and Mark, B. L. (2011). Mobility tracking based on autoregressive models. *IEEE Transactions On Mobile Computing*, 10(1):32–43.

Zhu, S., Setia, S., and Jajodia, S. (2003). LEAP: Efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security*.