INTERNATIONAL JOURNAL · ENGINEERING RESEARCH

International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 7th Aug 2019. Link :http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-08

Title **DISTRIBUTED FAULTY NODE DETECTION IN DELAY TOLERANT NETWORKS: DESIGN AND ANALYSIS**

Volume 08, Issue 08, Pages: 323–327.

Paper Authors

K.VENKATA KRISHNA, N.MAHESH

QIS Institute of Technology, Ongole, AP, India

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DISTRIBUTED FAULTY NODE DETECTION IN DELAY TOLERANT NETWORKS: DESIGN AND ANALYSIS

[1] K.VENKATA KRISHNA, [2] N.MAHESH
[1,2] Dept. of CSE, QIS Institute of Technology, Ongole, AP, India

ABSTRACT: Although web application attacks have existed for over the last 10 years, simple coding errors, failed input validation and output sanitization continue to exist in web applications that have led to disclosures for many well-known companies. The most prevalent web application attacks are SQL Injection, Cross Site Scripting and OS Command Injection. With an increased number of companies conducting business over the Internet, many attackers are taking advantage of lax security and poor coding techniques to exploit web applications for fame, notoriety and financial gain.There are multiple ways to detect and prevent these vulnerabilities from being exploited and leaking corporate data on the Internet. One method involves using SQL Injection to detect the attack and block or alert appropriate staff of the attack

Keywords: Delay tolerant networks, fault detection, iterative algorithms, distributed estimation, equilibrium analysis

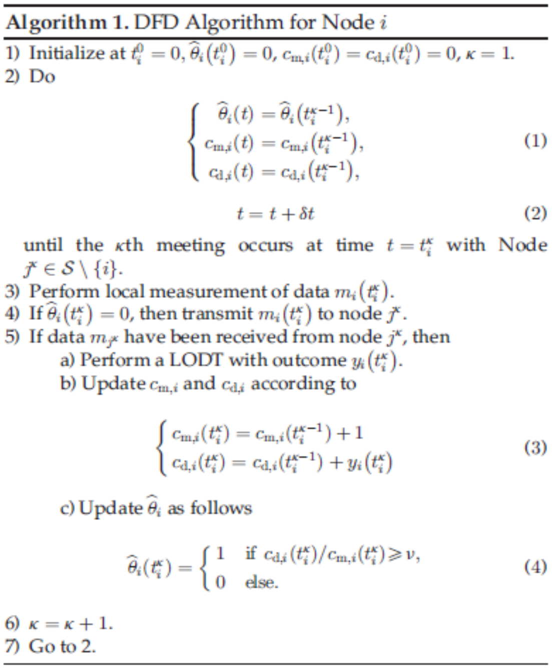
I INTRODUCTION

Delay Tolerant Networks (DTN) are challenging networks characterized by dynamic topology with frequent disconnections [1]. Examples of DTNs include Vehicular DTNs (VDTNs) [2] where two nodes can communicate with each other only when they are closely located. This connection is intermittent as the nodes are moving vehicles. Due to this sparse and intermittent connectivity, inference and learning over DTNs is much more complicated than in traditional networks, see, e.g., [3]–[8]. This paper considers the problem of distributed faulty node detection (DFD) in DTNs A node is considered as faulty when one of its sensors frequently reports erroneous measurements. The identification of such faulty nodes is very important to save communication resources and to prevent erroneous measurements polluting estimates provided by the DTN. This identification problem is quite complicated in DTNs when interactions are mainly between pairs of encountering nodes.Most of the classical DFD algorithms are using measurements of spatially-correlated physical quantities collected by many nodes to determine the presence of outliers and identify the nodes producing these outliers. In case of pair wise interactions, mismatch between measurements provided by two different node scan still be detected, but identifying directly which node produces erroneous measurements is not possible. This paper presents a fully distributed and easily implementable algorithm to allow each node of a DTN to determine whether it so when sensors are defective. We assume as in [9] that nodes are not aware of the status (good or defective) of their sensors, while their computation and communication capabilities remain fine, even if some of their sensors are defective. Most of the nodes of the DTN are assumed to behave in a rational way and are willingtoknow the status of their sensors. Some nodes, however, may be misbehaving, trying to perturb the detection process.As in [9]–[12], a Local Outlier Detection Test (LODT) is assumed to be able to detect the presence of outliers in a set of measurements, without necessarily being able to determine which are the outliers. This is a typical situation when only pair wise interaction sare considered, where measurements from sensors of only two nodes are compared.

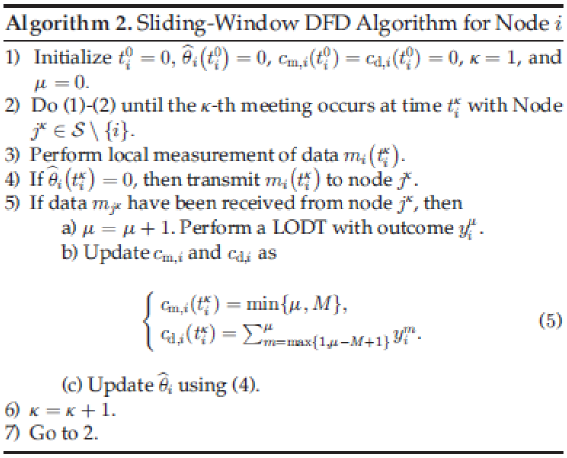
The generic LODT is characterized by its probabilities of detection and false alarm. When two nodes meet, they exchange their local measurement sand use them to perform the same LODT. The LODT results help both nodes to update their estimate of the status of their own sensors. When, for a given node, the proportion of meetings during which the LODT suggests the presence of outliers is larger than some threshold, this node decides its sensors may be defective. In this case, it becomes silent.Accordingly, it does not transmit any more its measurements to its neighbors, but keeps collecting measurements from nodes met and updates the estimate of the status of its sensors. It may then have the opportunity to change its estimate and communicate again. Although the LODT considered here are those of [9], this work differs significantly from [9] due to the communication conditions of DTNs, which require a totallydifferent DFD algorithm.

II EVOLUTION OF THE STATE OF A NODE

In this section, to simplify the presentation, the presence of misbehaving nodes is not taken into account. The impact of such nodes on the behavior of Algorithm 2



```
Algorithm 1. DFD Algorithm for Node $i$
1) Initialize at $t_i^0 = 0, \widehat{\theta}_i(t_i^0) = 0, c_{m,i}(t_i^0) = c_{d,i}(t_i^0) = 0, \kappa = 1$.
2) Do
       $\begin{cases} \widehat{\theta}_i(t) = \widehat{\theta}_i(t_i^{\kappa-1}), \\ c_{m,i}(t) = c_{m,i}(t_i^{\kappa-1}), \\ c_{d,i}(t) = c_{d,i}(t_i^{\kappa-1}), \end{cases}$   (1)
       $t = t + \delta t$   (2)
   until the $\kappa$th meeting occurs at time $t = t_i^\kappa$ with Node $j^\kappa \in \mathcal{S} \setminus \{i\}$.
3) Perform local measurement of data $m_i(t_i^\kappa)$.
4) If $\widehat{\theta}_i(t_i^\kappa) = 0$, then transmit $m_i(t_i^\kappa)$ to node $j^\kappa$.
5) If data $m_{j^\kappa}$ have been received from node $j^\kappa$, then
    a) Perform a LODT with outcome $y_i(t_i^\kappa)$.
    b) Update $c_{m,i}$ and $c_{d,i}$ according to
       $\begin{cases} c_{m,i}(t_i^\kappa) = c_{m,i}(t_i^{\kappa-1}) + 1 \\ c_{d,i}(t_i^\kappa) = c_{d,i}(t_i^{\kappa-1}) + y_i(t_i^\kappa) \end{cases}$   (3)
    c) Update $\widehat{\theta}_i$ as follows
       $\widehat{\theta}_i(t_i^\kappa) = \begin{cases} 1 & \text{if } c_{d,i}(t_i^\kappa)/c_{m,i}(t_i^\kappa) \geqslant \nu, \\ 0 & \text{else.} \end{cases}$   (4)
6) $\kappa = \kappa + 1$.
7) Go to 2.
```



```
Algorithm 2. Sliding-Window DFD Algorithm for Node $i$
1) Initialize $t_i^0 = 0, \widehat{\theta}_i(t_i^0) = 0, c_{m,i}(t_i^0) = c_{d,i}(t_i^0) = 0, \kappa = 1$, and
   $\mu = 0$.
2) Do (1)-(2) until the $\kappa$-th meeting occurs at time $t_i^\kappa$ with Node
   $j^\kappa \in \mathcal{S} \setminus \{i\}$.
3) Perform local measurement of data $m_i(t_i^\kappa)$.
4) If $\widehat{\theta}_i(t_i^\kappa) = 0$, then transmit $m_i(t_i^\kappa)$ to node $j^\kappa$.
5) If data $m_{j^\kappa}$ have been received from node $j^\kappa$, then
    a) $\mu = \mu + 1$. Perform a LODT with outcome $y_i^\mu$.
    b) Update $c_{m,i}$ and $c_{d,i}$ as
       $\begin{cases} c_{m,i}(t_i^\kappa) = \min\{\mu, M\}, \\ c_{d,i}(t_i^\kappa) = \sum_{m=\max\{1,\mu-M+1\}}^{\mu} y_i^m. \end{cases}$   (5)
    (c) Update $\widehat{\theta}_i$ using (4).
6) $\kappa = \kappa + 1$.
7) Go to 2.
```

III SYSTEM ANALYSIS

EXISTING SYSTEM

DFD is a well-investigated topic when considering Wireless Sensor Networks (WSNs). The WSNs considered in most of the papers are dense and have a static topology. DFD in DTNs is much less investigated. Classical DFD algorithms usually consist of two phases. First, an LODT is performed using data collected from neighboring nodes. LODTs aim to decide which data is erroneous. Second, the outcomes of the LODTs are disseminated to improve the decision accuracy. A problem related to DFD in DTNs has been considered in in the context of VDTN. A large number of sensor nodes are fixed and some vehicles, called mobile carriers (MC) collect data from these sensors. The sensor nodes can only communicate with the MCs in their vicinity. A MC needs to collect enough measurements to perform a test to decide which have been produced by defective sensors. Once a node is deemed defective by a MC, it is added to its blacklist. The MC provides information to sensors about their status. MCs also exchange their blacklists to accelerate the faulty node detection.

PROPOSED SYSTEM

The proposed structure works in the ceaseless auditing approach to find money related extortion inside an organization belonging to the financial area which will be our fundamental examination condition and furthermore centered around the fraud triangle hypothesis with the human factor considered as an essential component. Fraud Find is delivered with the objective of breaking down a lot of information from various sources of information for later preparing and enrollment. The specialist is an application introduced in the workstations of the clients (endpoints), so as to separate the information that they produce from the diverse wellsprings of data that reside on their gear. This application is responsible for sending the information entered by the client for ordering and characterization. Later this composed information is gotten by Log stash for its treatment.

IV IMPLEMENTATION

Service Provider

In this module, the Service Provider browses the required file, initializes nodes with digital signature and uploads to the end user (node a, node b, node c, node d, node e, node f) via Router.

End User

In this module, the End user can receive the data file from the Service Provider which is sent via Router, if malicious or traffic node is found in the router then it never forwards to the end user to filter the content and adds to the attacker profile.

Router The Router is responsible for forwarding the data file in shortest distance to the destination; the Router consists of Group of nodes, the each and every node (n1, n2, n3,n4,n5,n6,n7,n8,n8,n10,n11,n12, n13) consist of Bandwidth and Digital Signature. If router had found any malicious or traffic node in the router then it forwards to the IDS Manager. In Router we can assign the Sleeping time for the nodes and can view the node details with their tags Node Name, Sender IP, Injected data, Digital Signature, Sleeping time and status.

Attacker

In this module, the malicious node or the node details can be identified by a threshold-based classifier is employed in the Attack Detection module to distinguish DoS attacks from legitimate Sleeping Time. The Attacker can inject the fake message and generates the signature to a particular node in the router with the help of threshold-based classifier in testing phase and then adds to the attacker profile.

V SYSTEM DESIGN

SYSTEM ARCHITECTURE:

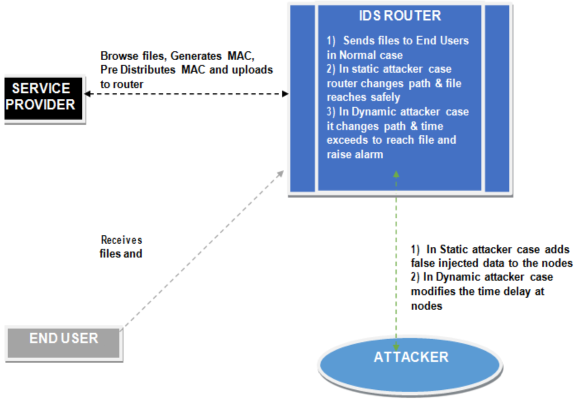

IDS ROUTER
1) Sends files to End Users in Normal case
2) In static attacker case router changes path & file reaches safely
3) In Dynamic attacker case it changes path & time exceeds to reach file and raise alarm
Browse files, Generates MAC, Pre Distributes MAC and uploads to router
SERVICE PROVIDER
Receives files and
END USER
1) In Static attacker case adds false injected data to the nodes
2) In Dynamic attacker case modifies the time delay at nodes
ATTACKER

Figure 1: System Architecture

Flow Chart Diagram

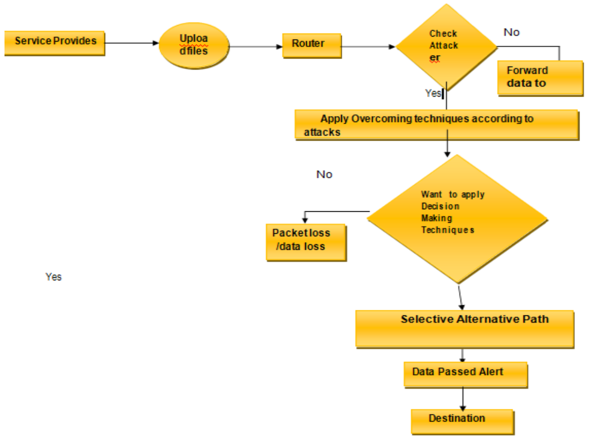


Figure 2: Flow Chart Diagram

VI CONCLUSION

This paper presents a fully distributed algorithm allowing each node of a DTN to estimate the status of its own sensors using LODT performed during the meeting of nodes. The DFD algorithm is analyzed considering a Markov model of the evolution of the proportion of nodes with a given belief in their status. This model is then used to derive a system of ordinary differential equations approximating the evolution of the proportions of the nodes in different states.

The existence and uniqueness of an equilibrium is discussed. Interestingly, the proportions at the equilibrium follow a binomial distribution. The approximations of these proportion so fondest equilibrium provide insight to properly choose the decision parameter of the DFD algorithm. In the simulations, a jump motion model, a Brownian motion model, as well as data bases containing traces offender-contact time in stands are considered .The results show good match with theory.The convergence speed of the DFD algorithm depends on the inter-contact rate and on the proportion of nodes with defective sensors p1. Nevertheless, p1 has not a significant impact on the non-detection and false alarm rate sat equilibrium, showing the robustness of the approach also in case of a large number of defective nodes. The impact of the presence of misbehaving nodes has also been considered, showing the robustness of the proposed DFD algorithm.

VII REFERENCES

[1] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, “Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges,”

IEEE Commun. Surveys Tuts., vol. 14, no. 2, pp. 607–640, Apr.–Jun. 2012.

[2] P. R. Pereira, A. Casaca, J. J. Rodrigues, V. N. Soares, J. Triay, andC. Cervell_o-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," IEEE Commun. Su rveys Tuts., vol. 14, no. 4, pp. 1166–1182, Oct.–Dec. 2012.

[3] K. Wei, M. Dong, J. Weng, G. Shi, K. Ota, and K. Xu, "Congestionaware message forwarding in delay tolerant networks: A community perspective," Concurrency Comput.: Practice Experience, vol. 27, no. 18, pp. 5722–5734, 2015.

[4] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay-tolerant networks," IEEE Trans. Mobile Comput., vol. 10, no. 11, pp. 1576–1589, Nov. 2011.

[5] V. N. Soares, J. J. Rodrigues, and F. Farahmand, "GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks," Inf. Fusion, vol. 15, pp. 102–113, 2014.

[6] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 22–32, Jan. 2014.

[7] L. Galluccio, B. Lorenzo, and S. Glisic, "Sociality-aided new adaptive infection recovery schemes for multicast DTNs," IEEE Trans. Veh. Tech., vol. 65, no. 5, pp. 3360–3376, May 2016.

[8] M. Panda, A. Ali, T. Chahed, and E. Altman, "Tracking message spread in mobile delay tolerant networks," IEEE Trans. Mobile Comput., vol. 14, no. 8, pp. 1737–1750, Aug. 2015.

[9] W. Li, F. Bassi, D. Dardari, M. Kieffer, and G. Pasolini, "Defective sensor identification for WSNs involving generic local outlier detection tests," IEEE Trans. Signal Inf. Process. Over Netw., vol. 2, no. 1, pp. 29–48, Mar. 2016.

[10] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in Proc. Workshop Depend. Issues Wireless Ad Hoc Netw. Sensor Netw., 2006, pp. 65–72.

[11] J.-L. Gao, Y.-J. Xu, and X.-W. Li, "Weighted-median based distributed fault detection for wireless sensor networks," J. Softw., vol. 18, no. 5, pp. 1208–1217, 2007.

[12] S. Ji, S.-F. Yuan, T.-H. Ma, and C. Tan, "Distributed fault detection for wireless sensor based on weighted average," in Proc. 2nd Int. Conf. Netw. Secur.Wireless Commun. Trusted Comput., 2010, pp. 57–60.

AUTHORS

K.VENKATA KRISHNA is Pursuing M.Tech (Computer Science and Engineering) in QIS Institute of Technology, Ongole, Prakasam Dist,
Andhra Pradesh, India.

N.MAHESH is currently working as Asst. Professor in QIS Institute of technology, in the Department of Computer Science and Engineering, Ongole, Prakasam Dist,Andhra Pradesh, India.

.