# IMPLEMENATION OF FAULTY NODE DETECTION USING DFD ALGORITHM IN DTN'S

[1]Dr. B. VAMSEE MOHAN, [2]D.SUGANDINI MIRIYAM

[1]Professor, Dept. of CSE, Visvodaya Engineering College, Kavali, AP, India.

[2]PG Scholar, Dept. of CSE, Visvodaya Engineering College, Kavali, AP, India.

***Abstract*** – Spreading of faulty data is serious issue. In situation like Delay Tolerant Networks (DTN) in specifically, the infrequent meeting actions need that nodes are better to share information each other properly. For that reason, schemes to quickly detect possible misbehaving nodes should be industrial. Distributed Faulty Node detection (DFD) has been considered in the survey in the situation of sensor and vehicular networks, but previous solutions struggles from long delays in detecting and separating nodes generated faulty information. This is inappropriate to DTNs where nodes gather only infrequent. This paper suggested a fully distributed and effortlessly enforceable method to allow each DTN node to quickly detect whether its sensors are generating false information. The dynamical conduct of the proposed algorithm is approximated by some constant time state conditions, whose equilibrium is described. The nearness of acting up nodes, attempting to bother the defective node location process, is likewise considered. Location and false alert rates are evaluated by contrasting both hypothetical and recreation results. Numerical outcomes evaluate the adequacy of the proposed arrangement and can be utilized to give rules for the algorithm design.

*Keywords* – Delay tolerant network, malicious node detection, DFD Algorithm.

## I. INTRODUCTION

Delay Tolerant Networks (DTN) are testing systems described by unique topology with successive detachments [1]. Instances of DTNs incorporate Vehicular DTNs (VDTNs) [2] where 2 nodes can speak with one another lone when they are firmly found. This association is irregular as the hubs are movingvehicles. Because of this scanty and irregular availability, induction and learning over DTNs is substantially more confused than in customary systems [3-8].

This paper thinks about the issue of distributed faulty node detection (DFD) in DTNs. A node is considered as flawed when one of its sensors every now and again

reports mistaken estimations. The distinguishing proof of such defective nodes is imperative to spare correspondence resources and to avert wrong estimations contaminating appraisals given by the DTN. This recognizable proof issue is very confused in DTNs when communications are basically between sets of experiencing nodes. A large portion of the old style DFD algorithms are utilizing estimations of spatially-associated physical amounts gathered by numerous nodes to decide the nearness of anomalies and recognize the nodes delivering these exceptions. On account of pairwise collaborations, the bungle between estimations given by two distinct nodes can in any case be recognized, however distinguishing legitimately which node produces incorrect estimations is beyond the realm of imagination.

This paper introduces a completely conveyed and effectively implementable algorithm to enable every node of a DTN to decide if its own sensors are imperfect. We expect as in [9] that nodes don't know about the status (great or damaged) of their sensors, while their algorithm and correspondence abilities stay fine, regardless of whether a portion of their sensors are blemished. The majority of the nodes of the DTN are expected to carry on in a judicious manner and are happy to know the status of their sensors. A few nodes, be that as it may, possibly get into misbehaving, attempting to irritate the identification procedure.

As in [9-13], a Local Outlier Detection Test (LODT) is thought to have the option to distinguish the nearness of anomalies in a lot of estimations, without essentially having the option to figure out which the exceptions are. This is a run of the mill circumstance when just pairwise connections are considered, where estimations from sensors of just 2 nodes are looked at. The conventional LODT is portrayed by its probabilities of discovery and false caution. At the point when two nodes meet, they trade their nearby estimations and use them to play out the equivalent LODT. The LODT results help the two nodes to refresh their estimate of the status of their own sensors.

At the point when, for a given node, the extent of gatherings during which the LODT proposes the nearness of anomalies is bigger than threshold, this node chooses its sensors might be faulty. For this situation, it ends up quiet. As needs be, it doesn't transmit any more its estimations to its neighbors however continues gathering estimations from nodes met and refreshes the estimate of the status of its sensors. It might then have the chance to change its estimate and impart once more.

In spite of the fact that the LODT considered here are those of [9], this work contrasts essentially from [9] due to the correspondence states of DTNs, which require a totally unique DFD algorithm. The investigation of the properties of the algorithm is additionally entirely unexpected. This paper shows that the behavior of the proposed DFD algorithm can be depicted utilizing Markov models and devices borrowed from control hypothesis and populace elements.

More top to bottom, the conviction of every node about the status of its sensors is quantized. The advancement of these quantized convictions is then appeared to pursue two Markov chains. A persistent time estimate of the advancement of the extent of nodes with comparable convictions is then inferred. Adequate conditions on the choice parameters to guarantee the presence and uniqueness of a harmony of the DFD algorithm are then given. Given the attributes of the LODT, upper and lower limits of the discovery rate, i.e., the extent of nodes which have viably recognized their sensors as imperfect, and of the bogus alert rate, i.e., extent of nodes which accept that their great sensors are in truth flawed, are likewise gotten. The effect of getting into misbehaving nodes, attempting to annoy the consequences of the DFD algorithm, is likewise considered.

## II. RESEARCH METHOD

In the proposed DFD algorithm, every (great or defective) node $i$ oversees two counters $c_{m,t}(t)$and $c_{d,t}(t)$ initialized at 0 at $t = 0$. Utilizing $cc_{m,t}(t)$, node $i$ tally the quantity of meeting during which it has gotten information from its neighbor, and has had the option to play out a LODT. Utilizing $c_{d,t}(t)$, it counts the number of LODT bringing about the discovery of outliers.When$\frac{c_{d,t}(t)}{c_{m,t}(t)} \geq v$, where $v$ is some consistent decision threshold, node $i$ views itself as conveying defective sensors, i.e., it sets its very own estimate $\widehat{\theta_t}(t) = 1$. Something else, it considers that its sensors are great, i.e., $\widehat{\theta_t}(t) = 0$.

At the point when a node with$\widehat{\theta_t}(t) = 1$ experiences another node, despite everything it takes estimations, however it doesn't send these information to the next hub to abstain from contaminating the system with exceptions. Any node, after getting information from another node, plays out a LODT and updates $c_{m,t}(t)$and $c_{d,t}(t)$. As a result, a hub which meets another node seeing itself as damaged transmits its information, however since it doesn't get any information, it doesn't refresh $c_{m,t}(t)$and $c_{d,t}(t)$ toward the part of the arrangement. Algorithm 1 outlines the proposed DFD

technique for a discretionary reference node $i$.

**Algorithm 1.** DFD Algorithm for Node $i$

1. Initialize at $t_i^0 = 0, \widehat{\theta}_i(t_i^0) = 0$, $c_{m,t}(t_i^0) = c_{d,t}(t_i^0) = 0, \kappa = 1$
2. Do

$$\begin{cases} \widehat{\theta}_t(t) = \widehat{\theta}_t(t_i^{\kappa-1}) \\ c_{m,t}(t) = c_{m,t}(t_i^{\kappa-1}) \\ c_{d,t}(t) = c_{d,t}(t_i^{\kappa-1}) \end{cases} \quad (1)$$

$$t = t + \delta t \quad (2)$$

Unti the $\kappa$th meeting occurs at time $t = t_i^{\kappa}$ with Node $j^{\kappa} \in S\backslash\{i\}$

3. Perform local measurement of data $m_i(t_i^{\kappa})$.
4. If $\widehat{\theta}_t(t_i^{\kappa}) = 0$, 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 $c_{m,t}$ *and* $c_{d,t}$ according to

$$\begin{cases} c_{m,t}(t) = c_{m,t}(t_i^{\kappa-1}) + 1 \\ c_{d,t}(t) = c_{d,t}(t_i^{\kappa-1}) + y_i(t_i^{\kappa}) \end{cases} \quad (3)$$

   c) Update $\widehat{\theta}_i$ as follows

$$\widehat{\theta}_t(t_i^{\kappa}) = \begin{cases} 1 \; if \; \dfrac{c_{d,t}(t)}{c_{m,t}(t)} \geq v \\ 0 \; else \end{cases} \quad (4)$$

6. $\kappa = \kappa + 1$
7. Go to 2.

The vector$X_i(t) = \left(\theta_i, c_{m,t}(t), c_{d,t}(t)\right)$ speaks to the (microscopic) state of every node $i$. As $t \to \infty$, one has$c_{m,t}(t) \to \infty$, which prompts a boundless number of potential qualities for $X_i(t)$ and the worldwide (macroscopic) conduct of the algorithm is hard to investigate. To constrain the quantity of potential states, we have considered the advancement of $c_{m,t}(t)$and $c_{d,t}(t)$ over a sliding time window containing the time moments of the last $M$ gatherings during which node $i$ has played out a LODT. Algorithm 2 is an adjusted adaptation of Algorithm1 representing this constrained skyline. It includes an extra counter $m$ for the quantity of LODT performed by node $i$. For whatever length of time that μ < M; (5) is comparable to (3).

**Algorithm2.** Sliding-Window DFD Algorithm for Node $i$

1. Initialize $t_i^0 = 0, \widehat{\theta}_i(t_i^0) = 0$, $c_{m,t}(t_i^0) = c_{d,t}(t_i^0) = 0, \kappa = 1$ and $\mu = 0$.
2. Do (1)-(2) until the $\kappa$ −th meeting occurs at time $t_i^{\kappa}$ with Node $j^{\kappa} \in S\backslash\{i\}$
3. Perform local measurement of data $m_i(t_i^{\kappa})$.
4. If $\widehat{\theta}_t(t_i^{\kappa}) = 0$, 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 $c_{m,t}$ *and* $c_{d,t}$ as

$$\begin{cases} c_{m,t}(t_i^{\kappa}) = \min\{\mu, M\}, \\ c_{d,t}(t_i^{\kappa}) = \sum_{m=\max\{1,\mu-M+1\}}^{\mu} y_i^m \end{cases} \quad (5)$$

c) Update $\widehat{\theta_i}$ using (4)

6. $\kappa = \kappa + 1$.

7. *Go to* 2.

## III. RESULTS ANALYSIS

This area analyzes the proposed DFD algorithm to some firmly related plan in the writing. Old style DFD algorithms are hard to apply with regards to DTN and no arrangements have been displayed so far in the writing for this particular situation. As needs be, so as to play out an important examination between our algorithm and a cutting edge method, we have considered the gossip algorithm talked about in [18] which speaks to the most vigorous and productive procedure with regards to order and appropriated estimation in unique situations like DTNs.

In [18], $n_S$ nodes are accepted to get an estimation

$$m_i = c + \theta_i + v_i, \forall\, i \in S,$$

of a typical amount $c$, where $v_i$ are acknowledge of free zero-mean Gaussian arbitrary factors with variance $\sigma^2$ and $\theta_i \in \{0, 1\}$ denotes the predisposition of every node. Every node is keen on the joint estimation of $c$ and $\theta_i$. Since the estimations delivered by the sensors with non-zero predisposition are bound to have bigger qualities, [18] proposes an estimator of $u_i$ dependent on a conveyed positioning of the nodes as per their measurement $m_i$. Hubs with an enormous position get an estimate $\widehat{\theta_t} = 1$; while nodes with a little position have $\widehat{\theta_t} = 0$.In request to apply the proposed DFD algorithm to this issue, consider the accompanying LODT

$$y_{i,j} = y_{j,i} = \begin{cases} 1 \; if \; |m_i - m_j| > \delta \\ 0 \; else, \end{cases}$$

where $\delta$ is an edge that outcomes in various estimations of probabilities $q_{F\,A}(2), q_D(1,1), and\; q_D(0,2)$ at that point the best possible estimation of $n$ can be set in like manner. Consider again $Infocom05$ for the reenactment with $n_1 = 10$ nodes picked arbitrarily with $\theta = 1$ and without getting into misbehaving node. Two situations are considered. In the main case, all nodes take a solitary estimation of $c$ at instatement. In the subsequent case, nodes take estimations at each gathering. Results are gotten as the normal of 200 independent Monte-Carlo reproductions. Fig. 1 analyzes the outcomes when $\sigma = 0.2$ and $\sigma = 0.3$. The order mistake and the estimation blunder are defined 〖

$$E_c = \frac{\sum|\theta_i - \widehat{\theta_i}|}{n_S} \; and \; E_e = \sum |c - \widehat{c_i}| / n_S.$$

On the off chance that nodes take a solitary estimation, the exhibition of the proposed algorithm is near the reference technique as far as $E_c$and$E_e$. At the point when nodes take new estimations at each gathering, the proposed DFD algorithm performs superior to the reference technique: the estimation of $E_c$ diminishes quicker and goes to be a lot littler. This is basically because of the node Ranking algorithm utilized in [18], which turns out to be less productive when nodes update at each gathering the amount as per which they are ranked.

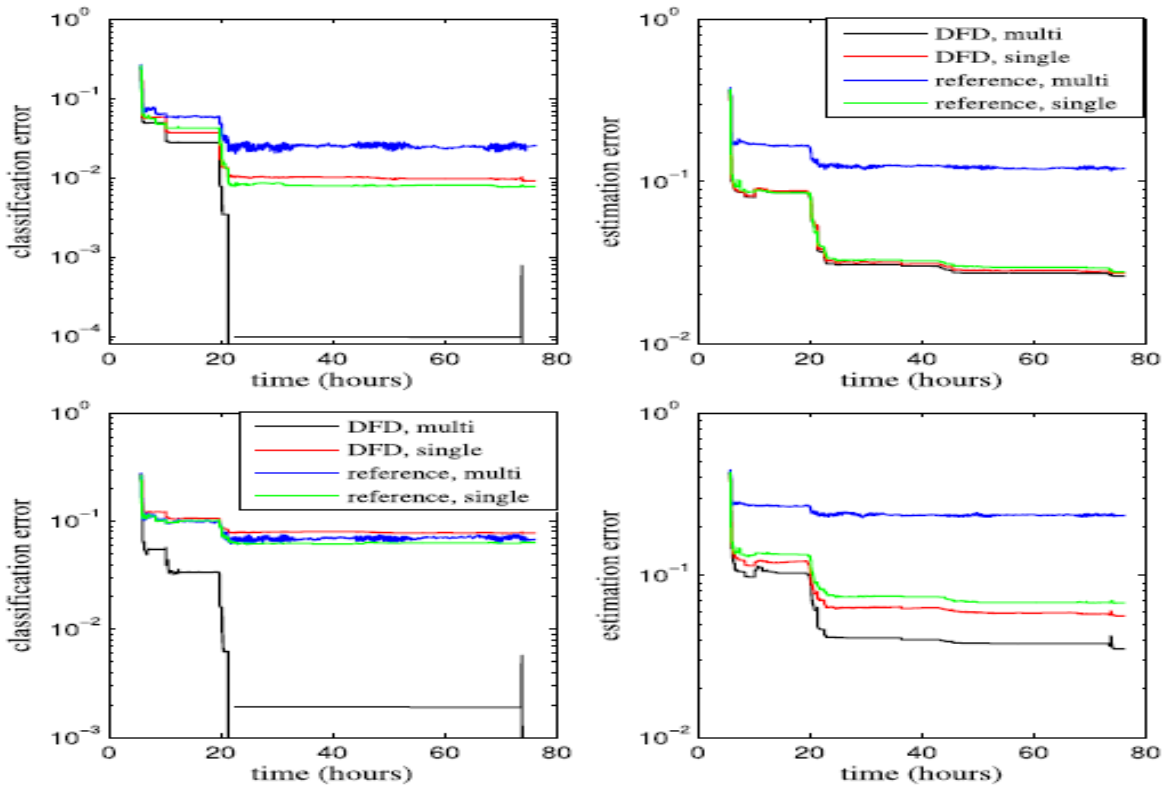


Fig. 10: Comparison of the DFD part of the estimation algorithm proposedin [18] with the proposed DFD algorithm, when $\sigma = 0.2$ and $\sigma = 0.3$ (bottom).

## IV. CONCLUSION

This paper exhibits a completely circulated algorithm enabling every node of a DTN to appraise the status of its own sensors utilizing LODT performed during the gathering of nodes. The DFD algorithm is examined considering a Markov model of the development of the extent of nodes with a given confidence in their status. The approximations of these extents of nodes at balance give understanding to appropriately pick the choice parameter of the DFD calculation. The assembly speed of the DFD algorithm relies upon the between contact rate and on the extent of hubs with blemished sensors $p_1$. By the by, $p_1$ has not a huge effect on the non-identification and

false caution rates at harmony, demonstrating the heartiness of the methodology likewise if there should be an occurrence of an enormous number of blemished nodes. The effect of the nearness of acting up nodes has likewise been considered, demonstrating the strongness of the proposed DFD algorithm.

**REFERENCES**

1. M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerantnetworking: A comprehensive survey on recent developmentsand 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. Cervello-Pastor, "From delay-tolerant networks to vehiculardelay-tolerant networks," IEEE Commun. Surveys 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, "Congestionawaremessage forwarding in delay tolerant networks: A communityperspective," Concurrency Comput.: Practice Experience, vol. 27,no. 18, pp. 5722–5734, 2015.

4. P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: Social-basedforwarding 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 misbehaviordetection scheme toward efficient trust establishment indelay-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 adaptiveinfection 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 messagespread in mobile delay tolerant networks," IEEE Trans. MobileComput., vol. 14, no. 8, pp. 1737–1750, Aug. 2015.

9. W. Li, F. Bassi, D. Dardari, M. Kieffer, and G. Pasolini, "Defectivesensor identification for WSNs involving generic local outlierdetection 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 ofwireless sensor networks,” in Proc. Workshop Depend. Issues WirelessAd Hoc Netw. Sensor Netw., 2006, pp. 65–72.

11. J.-L. Gao, Y.-J. Xu, and X.-W. Li, “Weighted-median based distributedfault 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 detectionfor wireless sensor based on weighted average,” in Proc. 2nd Int.Conf. Netw.Secur.WirelessCommun. Trusted Comput., 2010, pp. 57–60.

13. M. Panda and P. Khilar, “Distributed self fault diagnosis algorithmfor large scale wireless sensor networks using modifiedthree sigma edit test,” Ad Hoc Netw., vol. 25, pp. 170–184, 2015.

14. Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniquesfor wireless sensor networks: A survey,” IEEE Commun.Surveys Tuts., vol. 12, no. 2, pp. 159–170, Apr.–Jun. 2010.

15. A. Mahapatro and P. M. Khilar, “Fault diagnosis in wireless sensornetworks: A survey,” IEEE Commun. Surveys Tuts., vol. 15,no. 4, pp. 2000–2026, Oct.–Dec. 2013.

16. H. Dong, Z. Wang, S. X. Ding, and H. Gao, “A survey on distributedfiltering and fault detection for sensor networks,” Math. ProblemsEng., vol. 2014, 2014, Art.no. 858624.

17. E. F. Nakamura, A. A. Loureiro, and A. C. Frery, “Informationfusion for wireless sensor networks: Methods, models, and classifications,”ACM Comput. Surveys, vol. 39, no. 3, 2007, Art.no. 9.

18. A. Chiuso, F. Fagnani, L. Schenato, and S. Zampieri, “Gossip algorithmsfor simultaneous distributed estimation and classificationin sensor networks,” IEEE J. Sel. Topics Signal Process., vol. 5, no. 4,pp. 691–706, Aug. 2011.

19. F. Fagnani, S. M. Fosson, and C. Ravazzi, “A distributed classification/estimation algorithm for sensor networks,” SIAM J. ControlOptimization, vol. 52, no. 1, pp. 189–218, 2014.

20. F. Fagnani, S. M. Fosson, and C. Ravazzi, “Consensus-like algorithmsfor estimation of gaussian mixtures over large scalenetworks,” Math. Models Methods Appl. Sci., vol. 24, no. 2, pp. 381–404, 2014.