International Journal of Distributed and Parallel Systems (IJDPS) Vol.2, No.6, November 2011

Weak Estimation-Based Algorithm Fault-Tolerant Routing Wireless Sensor Networks

Reza Godaz[1]and Mohammad Reza KaghazGaran[2]

[1]Islamic Azad University, Mashhad Branch,Department of SoftwareComputer Engineering, Mashhad, Iran.
rezagodaz@yahoo.com
[2]Islamic Azad University, Mashhad Branch,Department of Software Computer Engineering, Mashhad, Iran.
kaghazgaran@yahoo.com

Abstract

Wireless Sensor Networks are characterized by the cooperative engagement of mobile nodes that constitute networks possessing continuously-changing infrastructures, the bereavement of centralized network managers, access points, fixed base stations, a backbone network for controlling the network management functions.While comprehensive studies have been carried out in the past several years for many sensor applications, they cannot be applied to the network with extremely low and sporadic connectivity, dubbed the Fault-Tolerant Wireless Sensor Network (FT-WSN). Without end-to-end connections due to sparse network density and sensor node stimulus, routing in FT-WSN becomes localized and ties closely to medium access control, which naturally calls for merging Layer 3 and Layer 2 protocols in order to reduce overheadand improve network efficiency. FT-WSN is fundamentallyan opportunistic network, where the communication linksexist only with certain probabilities and become the scarcestresource .there is a tradeoff between link utilization and energy efficiency.

KEYWORDS

wireless sensor Network, Fault-Tolerant, Medium Access Control

1. INTRODUCTION

The recent developments in sensor technology havemade it possible to gather massive information through a low cost distributed embedded sensor system. The mainstream approach is to densely deploy alarge number of small and inexpensive sensor nodes with low power, short range radio to form a connected wireless sensor network (WSN). The sensors in the network collaborate together to acquire the target data and transmit them to the sink nodes [1]. While thesemainstream approaches in the literature are well suited for many sensor applications, they cannot be applied to thescenarios with extremely low and intermittent connectivity due to sparse network density and sensor node mobility.

A Fault-Tolerant Wireless Sensor Network (FT-WSN) has been proposed in [2], [3] for pervasive information gathering in the aforementioned applications. Indeed FT-WSN aims to gather a massive amount of information from a statistic perspective and to update the information base periodically.

The FT-WSN consists of two types of nodes, i.e., the wearable sensor nodes and the high-end sink nodes. The former are attached to people, gathering target information and forming a loosely connected mobile sensor network for information delivery.

Without end-to-end connections, routing inFT-WSN becomes localized and ties closely to second Layer protocol, which naturally calls for merging Layer 3 and Layer 2 in order to reduce overhead and improve network efficiency. In this research, we first described a unipath On-Demand Distance Vector Routing Protocol (AODV protocol) and then other protocol called aunicast Dynamic On-Demand Routing Protocol (DSR protocol).Next we willdiscussa problem and solution of Weak estimation-based fault tolerant routing then we will explainit with Weak Estimation-base algorithm.

2.The AODV routing protocol

AODV is classified as a uni-path on-demand distance vector routing protocol. It, therefore, functions by using both a route discovery phase and a route maintenance phase by incorporating multi-hop routing in the intermediate nodes between the source and destination. In the AODV, every mobile node functions as a specialized router .It has been shown to be scalable with the increase in the number of mobile nodes in a WSN. It is characterized by its ability to provide loop-free route information in which broken links are resolved by repairing existing links or introducing new ones. Since there is no assumption on the presence of periodic advertisements by the nodes, there is a little requirement on the amount of bandwidth that should be available for the mobile nodes with comparison to the protocols that require the presence of advertisements.

AODV borrows basic route establishment and maintenance mechanisms from the Dynamic Source Routing (DSR) [14] protocol, and hop-to-hop routing vectors from the Destination-Sequenced Distance-Vector (DSDV) routing protocol [15].

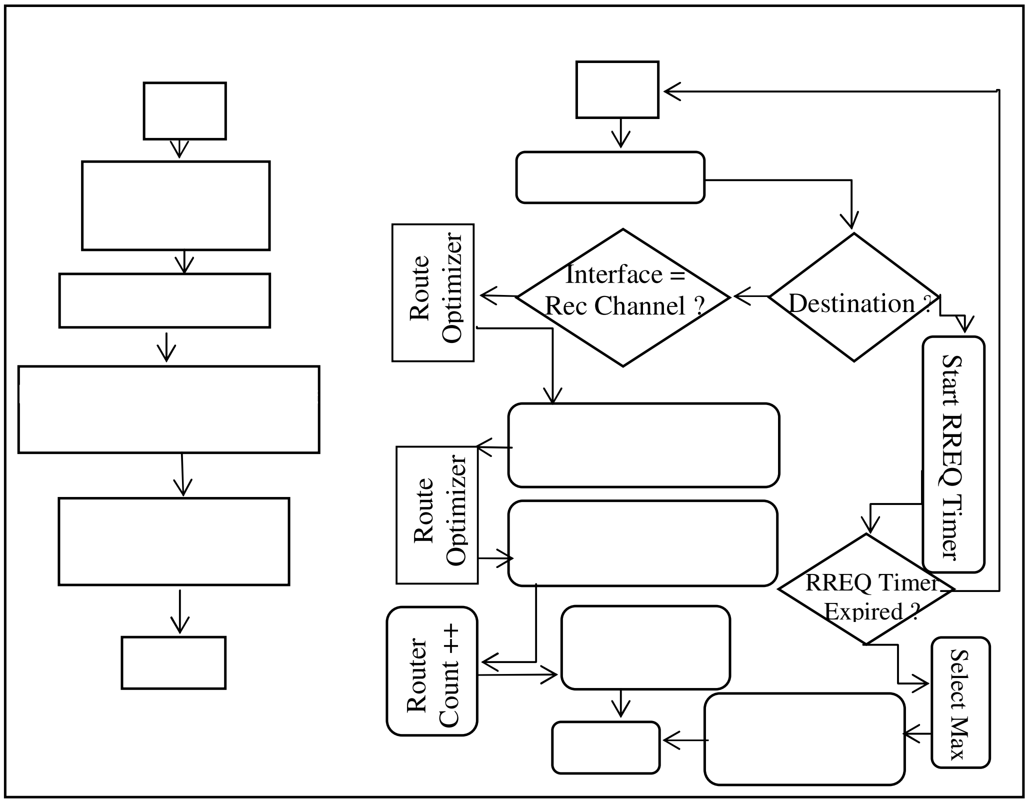


Fig. 1: Route Establishment

AODV has primarily two phases of operation:

(1) The route discovery phase

(2) The route maintenance phase

When one node needs to communicate with another node since there is no routing information in its table, the route discovery phase is triggered. The source specifies the destination node to which information needs to be transmitted, and floods the network with a so-called Route Request (RREQ) packet.

The node that receives the request is checked to see if it is identified as the destination node by the RREQ packet, or can be served as an intermediate node to transmit information to another node in the network. If the node generates a unicast Route Reply Packet (RREP),it is sent back along the reverse path in which the RREQ packet was originally sent by the source node. Once the source receives the RREP packet, then it knows where and how to transmit the packet.

The route maintenance phase is triggered whenever a broken link is detected by any node, and when that node attempts to forward a packet to the next hop. In the route maintenance phase, once the next hopis found to be unreachable, the upstreamnode sends an unsolicited RREP packet which is possessing a new sequence number that isgreater than the previously-known sequence number by unity. It also sends a hop count of "∞" to all the neighboring upstream nodes, which, in turn, replay that information to their active neighbors, until all active source nodes are notified [9].

3. The DSR protocol

Like the AODV, the DSR is a unicast Dynamic On-Demand Routing Protocol. It is a source routing protocol, where the source explicitly provides a packet with the complete information of the route to follow, which is subsequently used by the intermediate nodes to forward the packet to the correct destination node [7]. The DSR only routes packets between hosts that want to communicate with one another. When two nodes need to communicate with each other, the sender node determines a route. This is done based on the information which is stored in it's cache, or based on the results of a route discovery phase, depending on whether the information about the destination node is already available to the source node [5] or not.

The DSR requires that the sender determines and stores in the packet's header of the source route, where the address of each host in the network is explicitly provided until it can reach the intended destination node. The source finds out the complete route to the destination from a route cache which stores the routing information to different nodes in the network. If such an entry is found, the sender uses this route to send the packet, if such an entry is not found, a route discovery exercise, similar to the one discussed for the AODV protocol which is initiated by the source route. After the next destination is successfully identified, the packet is then sent to the first hop in the identified sequence of nodes by the source.

The DSR is characterized by its ability to quickly adapt itself to routing changes in environments in which there are frequent and rapidly-occurring host movements. One of the important aspects of the DSR is that there is no requirement for periodic route advertisements, as is frequently required in many routing protocols. This reduces an overall overhead on the network bandwidth, especially because most mobile nodes in WSN networks are operated over battery power, and there are often situations in such networks when there are no periodic routing advertisements to taking place [5].

4. Multipath routing protocols

Multipath routing protocols proposed in the literature are of different types, some of them are based on the foundational principles behind the AODV and DSR protocols. However, all multipath routing protocols share a common characteristic, i.e., they discover multiple routes between a pair of source-destination nodes. Multipath routing protocols take advantage of the inherent redundancy observed in networks to find multiple routes from one source node to a destination node. This becomes advantageous for ad hoc networks because they are characterized to be very dynamic, and unpredictable in nature [7].

This redundancy increases the reliability in the transmission of the information [17], implying that there is a much greater chance (than in uni-path routing) that at least one of the paths will be able to successfully deliver the packet. Some of the multipath routing algorithms are also capable of providing load balancing in the network by carefully selecting a mechanism to split traffic along different routes to avoid overloading any single route. This is often quite advantageous in wireless network environments because while, sometimes, it might be difficult to guarantee the reservation of a large portion of the bandwidth through a single path, it might be possible to reserve small portions of the bandwidth over multiple routes through many paths that taken together [8].

The multiple paths discovered in multipath routing may take different forms which categorized as being node disjoint, link disjoint, or non-disjoint routes. In node disjoint routes, there are no overlapping nodes or links. In link disjoint routes, there are no overlapping links, while in non-disjoint routes one permits overlapping nodes or links. The advantage of having disjoint routes is that they provide greater fault-tolerance, in the sense that if one of the nodes/links fail, it is quite unlikely that the failure will affect any of the other routes. Route maintenance in multipath routing is similar to the one done in uni-path routing, except that the protocol requires a decision to be made as to when a route discovery phase needs to be triggered, i.e., when a broken link is identified. It is because triggering a route discovery every time that a failure is identified, introduces more traffic, and results in a degraded network performance. On the other hand, if one waits for all the disjoint routes between a pair of source-destination nodes to fail before invoking a route discovery, it might result in an unreasonable amount of delay [9].

5. Related Work

Research on Fault-Tolerant Wireless Sensor Network (FT-WSN) is motivated mainly by the Delay Tolerant Network (DTN) and its applications in sensor networks and mobile ad hoc networks. An overview of FT-WSN is presented in [1], outlining its application scenariosand unique characteristics, such as sensor mobility, looseconnectivity, fault tolerability, delay tolerability, and bufferlimit. Following that, an in-depth study of FT-WSN is given in [5], where two basic data delivery approaches in FT-WSN, namely, direct transmission and flooding, are discussed with their performance analyzed by using queuing models. That are controllable by the users through the Internet.

The SEDNFT (Sensor Networking with Delay/Fault Tolerance) project targets at developing a proof-of-concept sensor network for lake water quality monitoring, where the radio connecting sensors are mostly turned off to save power, thus forming a loosely connected FTN network [8]. FTN/SN focuses on the deployment of sensor networks that are inter-operable with the Internet protocols [9].The Shared Wireless Sensor Info-Station (SWSIM) system isproposed in [13, 14] for gathering biological information of radio-tagged whales. It is assumed in SWIM that the sensor nodes move randomly and thus every node has the same chance to meet the sink. A sensor node distributes a number of copies of a data packet to other nodes so as to reach the desired data delivery probability.

In [12] the mobile sensors are employed to support wildlife tracking for biology research. The [12] project targets at building a position-aware and power-aware wireless communication system. A history based approach is proposed for routing, where the routing decision is made according to the node's pass success rate of transmitting data packets to the base station directly. When a sensor meets another sensor, the former transmits data packets to the latter if the latter has a higher success rate.

6. Fault-tolerant WSNs

Due to the mobility of the nodes and the associated rapidly-changing topologies, the reliability of the correct transmission of messages is an important concern for WSNs. Hence, at present we consider strategies that would guarantee the delivery of packets in adversarial environments, and in the presence of node/link failures. The well-known WSN routing algorithms listed above (e.g., DSR, multipath routing etc.) are unsuitable as fault-tolerant routing algorithms for WSNs. Since the DSR chooses the shortest path route for packet transmission in adversarial environments, it can be shown that it will achieve a low packet delivery rate. On the other hand, multipath routing algorithms are strong in their fault-tolerance ability, because they send multiple copies of packets through all possible (disjoint) routes between a pair of source-destination nodes.

However, the disadvantage of the multipath routing algorithms is that they introduce an unnecessary amount of overhead on the network. Without a mechanism that "tolerates" route failures due to malfunctioning nodes (while making routing decisions), the performance of ad hoc network protocols will necessarily be poor, and the routing decisions made by those protocols would be erroneous. The algorithms that attempt to solve the fault-tolerant routing problem do so by:

1. Either "flooding" the network with multiple redundant packets along different paths between a pair of source-destination nodes (thus, increasing the probability of a successful transfer);

2. Follow a dynamic on-demand routing protocol, where the source explicitly provides, a priori, the transmitted packet with the complete information of the route need to be followed as well , and hence minimizing the number of multiple redundant packets being transmitted;

3. Seeking a "happy" medium between the latter strategies, namely, by estimating the potential profitability of maintaining selected paths.

7. Message Fault Tolerance

Each sensor has a data queue that contains data messages ready for transmission. The data messages of a sensor come from three sources:

(a) After the sensor acquires data from its sensing unit, it creates a data message, which is inserted into its data queue;

(b)When the sensor receives a data message from another sensor, it inserts the message into its data queue;

(c) After the sensor sends out a data message to a non-sink sensor node, it may also insert the message into its own data queue again, because the message is not guaranteed to be delivered to the sink.

In FT-WSN, multiple copies of the message may be created and maintained by different sensors in the network, which is resulting in redundancy. The fault tolerance degree (FTD) is introduced to represent the amount of redundancy and to indicate the importance of a given message. More specifically, each message copy is associated with an FTD, which is defined as the probability that at least one copy of this message is delivered to the sink by other sensors in the network. Let f_i^Mdenote theFTD of message M in the queue of sensor i. $f_i^M$is determined when the message is inserted into the queue. For anew message generated from the sensing unit, $f_i^M= 0$, i.e.,with the highest importance. During data transmission, themessage FTD will be updated. Without loss of generality,let's consider a sensor i, which is multicasting data message

M to a set of nearby sensors denoted by Φ. The messagecopy transmitted to neighbor node j is associated with anFTD of f_j^M,

$$f_j^M= 1-(1-[f_i^M])(1-\xi i) \prod_{m \in \Phi, m\neq j}^{n}(1-\xi m),(2)$$

and the FTD of the message at sensor i is updated as

$$f_j^M= 1-(1-[f_i^M]) \prod_{m\in\Phi}(1-\xi_{\psi m}),$$

where $[f_i^M]$ is the FTD of message M at sensor i before multicasting.

The above process repeats at each time when message j is transmitted to another sensor node. In general, the more times a message has been forwarded, the more copies of the message are created, thus increasing its delivery probability. As a result, it is associated with a larger fault tolerance.

8. Weak Estimation-Based fault tolerant routing

As mentioned earlier, the objective of the weak-estimation based fault tolerant routing solution proposed in [15] was to minimize the overhead by sending the least possible number of redundant packets, while guaranteeing a certain rate for the delivery of packets. We again emphasize in this chapter that there is a tradeoff between the rate of delivery of packets and the overhead. It is possible to achieve a very high packet delivery rate if the number of packets sent is not a concern (e.g., by using the multipath routing scheme). On the other hand, it is possible to achieve a very low overhead, if we do not care about the number of packets that are successfully delivered (e.g., by using the DSR scheme).

Thus, attempting to increase one will decrease another and vice versa. What is challenging is to see how we can achieve a "balance" between both of them. In other words, we need an algorithm that will be able to minimize the overhead by guaranteeing a certain level of efficiency of the packet delivery process. To achieve our objective, we propose a stochastic learning-based weak estimation fault-tolerant routing scheme.

9. Weak Estimation Learning

In statistical problems involving random variables, the quality, reliability, and accuracy of the estimation are important considerations. Traditionally, there have been different estimation schemes proposed in the literature, which can broadly be classified as either belonging to the Maximum Likelihood Estimator (MLE) class of algorithms [3,4], or as belonging to the Bayesian family of algorithms [1,3]. Although the above estimation schemes have been proved to be quite efficient, they work under the premise that the underlying distribution in the environment is stationary, i.e., the estimated parameter does not vary with time. n the fault-tolerant routing solution presented in [15], we had used this efficient procedure for the estimation of the packet delivery probability through available paths. It is called the Stochastic Learning Weak Estimator (SLWE) scheme2 [12,13], and is based on the principles of the stochastic learning paradigm. It uses a learning parameter, γ, which does not influence the mean of the final estimate. On the other hand, the variance of the final distribution, and the speed of convergence decrease with the increase in the value of this learning parameter. We discuss the weak estimation schemeatbelow.

Let us consider a binomially distributed random variable, X, as follows:

$$\mathrm{X} = \begin{cases} 0 \ \text{ with Probability } \mathrm{S}_0 \ ; \\ 1 \ \text{ With probability } \ \mathrm{S}_1 \ ; \end{cases}$$

Such that S0 + S1= 1, Where S = $[\ \mathrm{S}_0, \mathrm{S}_1]^{\mathrm{T}}$

At any time, t, let X assumes the value x(t). In order to estimate 01 s and s , SLWE keeps track of the running estimate $p_i(t)$ of s_i at time t, where i = 0,1. In such a setting, the value of 0 p is updated using the following multiplicative scheme:

$$\mathrm{P}_0 = \begin{cases} \gamma \ \times \ P_0\,(t)\ if\ x(t) = 1 \\ 1 - \ \gamma \ \times P_1\ if\ x(t) = 0 \end{cases}$$

10 where γ is a constant (0<γ<1), called the learning parameter, and p (t+1) = 1 - p (t +1) .Now we present some of the interesting results [12] that concerning the SLWEat below.

Theorem 1: Let X be a binomially distributed random variable, and P (n) be the estimate of S at time ‘ n ’. Then, E[P (∞)]=S . Proof. Based on the updating scheme specified by Eq. (2), the conditional expected value of $p_1(n+1)$ given P can be seen to be:

$$\mathrm{E[P1(n+1)\mid P]} = \gamma\ \mathrm{S}_2\mathrm{p}_1 + \mathrm{S}_1 - \gamma\ \mathrm{S}_1 + \gamma\ \mathrm{S}_2\mathrm{P}_1 = (1-\gamma)\ \mathrm{S}_1 + \gamma\ \mathrm{P}_1(\ \mathrm{S}+\mathrm{S}_2) = (1-\gamma\ )\ \mathrm{S}_1 + \gamma\ \mathrm{P}_1$$

The next results which we shall prove indicate that E[P(n + 1)] is related toE[P (n)] by means of a stochastic matrix. We derive the explicit stochastic dependence, and allude to the resultant properties by virtue of the stochastic nature of the matrix. This leads us to two results, namely the mean of the limiting distribution of the vector P(n), and which concerns its rate of convergence. The mean of P (n) is shown to converge exactly to the mean of S . The implications of the “asymptotic” nature of the results will be clarified presently.

Theorem 2: If the components of P (n +1) are obtained from the components of P (n) as :

$$\mathrm{E[P(n+1)]} = M^T\,\mathrm{E[P(n)]},$$

where M is a stochastic matrix. Thus, the limiting value of the expectation of P (0) converges to S , and the rate of convergence of P to S is fully determined by γ . Consider the proof.

Since $p_1 + p_2 = 1$, we can write:

$$\mathrm{E}[\mathrm{P}_1(\mathrm{n}+1) \mid \mathrm{P}] = \begin{cases} (1-\ \gamma)\ \mathrm{S}_1(\ \mathrm{P}_1 +\ \mathrm{P}_2)\ + \gamma\ \mathrm{E}[\mathrm{P}_1(\mathrm{n})] \\ \\ (1-\ \gamma)\ \mathrm{S}_1(\ \mathrm{P}_1 +\ \mathrm{P}_2)\ + \gamma\ \mathrm{E}[\mathrm{P}_1(\mathrm{n})] \end{cases}$$

Substituting the above equalities, simplifying and taking expectations again leads to the following victoria lform:

$$E[P(n+1)]= M^{T} E[P(n)]$$

$$M = \begin{bmatrix} (1-\gamma)S_1+\gamma & (1-\gamma)S_2 \\ (1-\gamma)S_1(1-\gamma)S_2+\gamma & \end{bmatrix}$$

we now show that:

$$E[P(\infty)] = (1-\gamma)S_1\{E[P(\infty)]+E[P_2(\infty)]\}+\gamma E[P_1(\infty)] = (1-\gamma)S_1+\gamma E[P(\infty)] =$$

$$E[P_1(\infty)](1-\gamma)=S_1(1--\gamma)$$

An exact parallel argument leads to the result that $E[P_2(\infty)]=S_2$, whence the first result of the theorem is proved. Observing that $(MI-\gamma)$ has the common factor $(\gamma-1)$, it follows that the convergence of P to S , which, in general, is determined by the eigenvalues of M , is fully determined by γ . Hence the theorem. From the analysis given above, we can derive the explicit expression for the asymptotic variance of the SLWE. We show that a small value of γ leads to fast convergence and a large variance. And, a large value of γ implies slow convergence and a small variance.

9.1 The WEBFTR algorithm

Weak-Estimation Learning scheme leads to propose a new fault-tolerant routing algorithm, named the Weak-Estimation-Based Fault Tolerant Routing (WEBFTR) Algorithm, which is capable of efficiently estimating the probability of the delivery of packets through the paths available at any moment. Like the E2FT algorithm [10], the WEBFTR algorithm involves, among other steps, a route estimation phase and a route selection phase. The route estimation phase is used to estimate the packet delivery probability of all the routes at the disposal at any time instant, whereas the route selection phase is used to select those routes that are confirmed to have satisfied a certain optimization constraint, and to drop the unnecessary multipath routes between a pair of source-destination nodes.n the route estimation phase, N packets are sent along a path p.

The source node estimates the fraction of packets delivered, $\gamma(p)$ from the number of packets, N', received along that path3. In our strategy, the estimate of the packet delivery probability is refined with the increase in the number of iterations. At every iteration, a set of packets is transmitted through each of the multipath routes between a pair of source-destination nodes. We can have two possible scenarios for any path: The nodes in a path either forward the packets correctly, or they do not. Consequently, we can use a binomial estimation scheme (based on the above SLWE) as follows.

$$\vartheta_0^{\wedge}(P) = \begin{cases} \gamma + \vartheta_0^{\wedge}(P) & \text{if the path dose not forward the packet correctly} \\ 1-\gamma\times\vartheta_0^{\wedge}(P) & \text{if the path forward the packet correctly} \end{cases}$$

Whereγ is the learning parameter, such that $0<\gamma<1$, and $\gamma(p) = 1 - \gamma(p)$.

The dropping algorithm selects a path, p_{min} , from all the available paths, π , with the minimum packet delivery estimation value, where the latter is examined to see if the following dropping condition is satisfied [10]:

$$\vartheta^1_{\mathrm{WE}\frac{1}{\mathrm{M}}}(\pi) \geq \vartheta_1^* \text{where} \vartheta^1_{\mathrm{WE}\frac{1}{\mathrm{M}}}(\pi) = 1 - \prod_{\mathrm{P} \in \pi}(1 - \vartheta^1_{\mathrm{WE}\cdot\frac{1}{\mathrm{M}}}(\mathrm{P})), \text{ and } \pi - \{\mathrm{P}_{\min}\}\ ;$$

9.2 Algorithm WEBFTR

9.2.1 Input

- A graph (network) with a set of nodes, and a set of links connecting the nodes.

- The nodes are mobile, and links connecting them can be reset with the change in the position of the nodes.

- Some of the nodes in the network are faulty with a certain packet delivery rate dependent on the distance of the node from the center of the area of mobility of the mobile nodes, which, for the purpose of this study, is the "simulation area".

9.2.2 Output

- All the incoming packets are delivered from the source node to the destination node, with the intention of maximizing the packet delivery rate, and minimizing the network overhead.

9.2.3 Algorithm

BEGIN

Step 0 (initialization): Initialize a vector WEBFTR-MP that stores all the paths in use, and WEBFTR-Nodes that stores all the nodes in the graph, along with the information about their estimated packet delivery probabilities. At each time unit, do the following:

Case 1: If the unit of time is a simulation pause then :

Step 1. Save the estimated packet delivery probability of each node in the vector WEBFTR-Nodes.

Step 2. Update the edges and probabilities in the graph to reflect the current position of the nodes, and calculate the new paths from the source to the destination.

Step 3. Use the values stored in WEBFTR-Nodes in order to calculate the estimated (using the SLWE) packet delivery probability of each path.

Case 2: At each unit of time:

Step 1. Try to confirm or drop paths. Paths dropped are removed from the WEBFTR-MP vector.

Step 2. Use all the paths in the WEBFTR-MP vector to send the packets, and calculate the number of packets that are received for each path and the total number of non-duplicated packets that are received.

END

10. Synchronous Phase

In this phase, all the data transmissions are synchronized, and thus contention-free. After obtaining information from the qualified receivers, node i decides which of them are to be selected for data forwarding, according to the D/FTD of the outgoing message M and the receivers' delivery probabilities. Let $E = \{\Phi_z \mid 1 \leq z \leq Z\}$ denote the Z qualified receivers, sorted by a decreasing order of their delivery probabilities. In order to reduce unnecessary transmission overhead, only a subset Φ of them are selected for transmission so that the total delivery probability of message M is just enough to reach a predefined threshold $\Re$. The procedure for determining Φ is as follows:

```
$\Phi=0$
For Z=1 : Z do
If $\varepsilon_I<\varepsilon_{\varphi_z}$ and $B_{\varphi_z}(f_i^m) > 0$ then
$\Phi = \Phi \cup \varphi_z$
End if
If $1-(1-f_i^m)\prod_{m\in\Phi}(1-\varepsilon_m) > R$ then
Break end if
end
end for
```

11. Performance metrics

Two metrics were used in [15] for evaluating the performance of the algorithms invoked in the experiments:

11.1 Percentage of packets delivered:

This represents the rate of the successful delivery of packets to the destination, and is calculated as follows: At each second, the packet delivery probability of all the paths in use is calculated. Then, for each packet sent at that time unit, a random number between 0 and 1 is generated. If the number is lower than the packet delivery probability, the packet is considered as having been delivered.

Thereafter, after all the iterations, the percentage of delivered packets is calculated as follows:

$$\text{Percentage delivered packets}= \frac{\text{total number of delivered packets}}{\text{total number of sent packets}}$$

11.2 Overhead:

This represents the overall number of packets sent. This Overhead index is calculated as the product of the total length of all the paths in use, and the number of packets sent per second (time unit).

12. Experimental results :

Several experiments were conducted [15] to assess the performance of WEBFTR (the proposed algorithm) with respect to the benchmark algorithms. The results of the following three sets of experiments are presented below (also available in [15]):

- Variation in Pause Time

- Variation in Spars
- Variation in the faultiness of nodes

12.1 Variation in Pause Time:

As noted earlier, the Pause Time is a parameter specific to the simulation, which indicates how much an algorithm is capable of accommodating the mobility of the nodes. The resultsof the simulation for this scenario are givenin Figure 2.

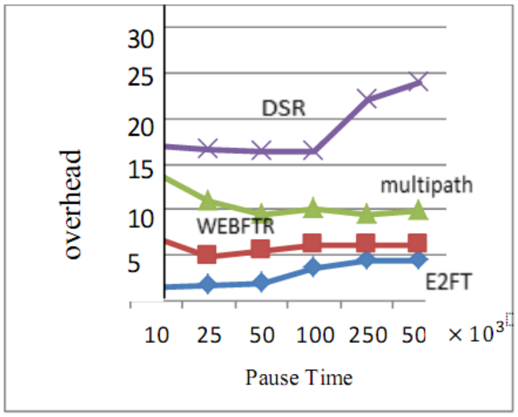


Figure.2. Plot of the Overhead versus the Pause Time have tested forthe various algorithms.

From this figure, we notice that with respect to the Overhead, while the blind multipath routing is the worst, the DSR is the best, and the metric for the E2FT lies somewhere in between the DSR and the multipath curves. This is, of course, understandable. Further our proposed algorithm improves on the performance of the E2FT scheme by decreasing the Overhead by 25-50%.

For example, when the Pause Time is 250 seconds, the Overhead for the multipath routing is 19,790, that for E2FT is 8,740, while for the WEFTR is 7,225. On the other hand, from Figure 2, we observe that the WEFTR achieves an almost similar order of performance when compared to the E2FT.

However, by examining Figures 3 and 4 together, one can infer that our proposed algorithm (WEFTR) is capable of significantly reducing the Overhead of the best fault tolerant routing algorithm (E2FT) currently available, while achieving a packet deliveryperformance guarantee of at least 80%. Thus, if one considers both of these issues simultaneously, it is clear that our algorithm always performs much better than both of the DSR and the blind multipath routing schemes.

12.2 Variation in Spars:

In the second set of experiments, we intended to study how thealgorithms compared toeach other with respect to the variation in the Spars of the nodes in the network.

As mentioned earlier, the value of Spars ranges between 0 and 1, where 0 represents the smallest percentage of Spars, and 1 represents the largest percentage of Spars. Since the nodes are mobile, the question of how often they connect with each other, depends on how close they can get to one another, and, clearly, this is directly related to the Spars. The different Sparsvalues used in our experiments indicate the relative number of edges between the nodes in the network.

12.3 Variation in Faultiness:

Faultiness is an internal simulation parameter that indicates how many nodes will be faulty in a given environment. It influences the faultiness behavior of the nodes, given their distance from the center of the region of operation of the nodes.

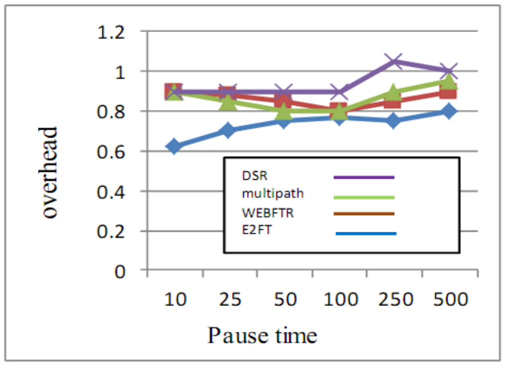


Figure.3. Plot of delivered packetspercentage versus Pause Time have tested for the various algorithms.

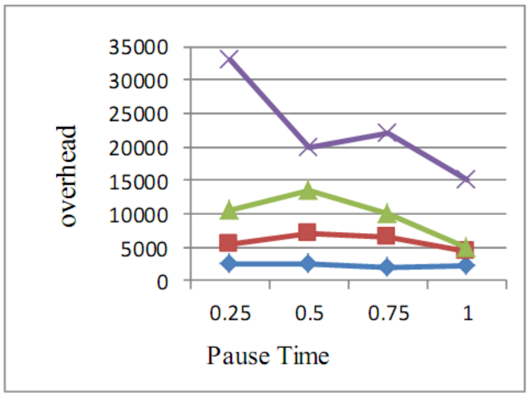


Figure.3.Plot of the Overhead versus the Spars have tested for the various algorithms.

Figures 2 and 3 depict the variation in the Overhead, and the percentage of delivered packets, with the variation in the Faultiness. In our experiments, we had used the Faultiness parameter to vary from a very low value to a very high value (on a scale of 0 to 1).

We observed that, even in this set of experiments, our proposed algorithm delivers much better performance, with comparison to the other algorithms. For example, when the Faultiness parameter has a value of 0.25, the Overhead for the blind multipath routing is 13,690, for the E2FT is 5,240, while for the WEBFTR, it is 3,150. Thus, in this case, our algorithm showed an improvement of about 62 % over multipath routing, and an improvement of about 40 % over the E2FT algorithm. All of these algorithms, however, in general, showed comparable performance with respect to the percentage of successfully delivered packets.

13. CONCLUSION

WSN hasseveral unique characteristics such as sensor mobility, looseconnectivity, fault tolerability, delay tolerability, and bufferlimit. We have established a queuing model for DFT-WSNby using Jackson network theory.

Our simulation results show that DFT-WSN achieves thehigher message delivery ratio with acceptable delay andtransmission overhead, compared with simple schemes suchas flooding and direct transmission or other approaches in the literature.

14. ACKNOWLEDGEMENTS

The authors would like to thank Deputy Research at the Mashhad Branch, Islamic Azad University support.

The authors is especially grateful to one of the anonymous referees whose comments led to an improved statistical model.

REFERENCES

[1] I. Akyildiz, W. Su, and Y. Sankarasubramaniam, “A survey on sensornetworks,” IEEE Commun. Mag., vol. 40, no. 8, pp. 102–114, 2002.

[2] Y. Wang, F. Lin, and H. Wu, “Poster: efficient data transmission in delay fault tolerant mobile sensor networks (DFT-MSN),” in Proc. IEEE International Conference on Network Protocols (ICNP 2005).

[3] Y. Wang and H. Wu, “DFT-MSN: the delay fault tolerant mobile sensor network for pervasive information gathering,” in Proc. Twenty-Fifth Annual Joint conference of the IEEE Computer and CommunicationsSocieties (INFOCOM 2006), to appear.

[4] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall,and H. Weiss, “Delay tolerant network architecture,” draft-irtf-dtnrg-arch-02.txt, 2004.

[5] K. FALL , “A delay-tolerant network architecture for challenged internets,”in Proc. 2003 Conference on Applications, Technologies, Architectures,and Protocols for Computer Communications (SIGCOMM), pp. 27–34,2003.

[6] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott,and H. Weiss, “Delay-tolerant networking–an approach to interplanetaryInternet,” IEEE Communed. Mag., vol. 41, no. 6, pp. 128–136, 2003.

[7] K. Fall, W. Hong, and S. Madden, “Custody transfer for reliable deliveryin delay tolerant networks.” Technical Report, Intel Research, Berkeley.

[8]M. Ho and K. Fall, “Poster: delay tolerant networking for sensor networks,” in Proc. IEEE Conference on Sensor and Ad Hoc Communications and Networks, 2004.

[9] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: modeling a three-tier architecture for sparse sensor networks,” in Proc. First International Workshop on Sensor Network Protocols and Applications, pp. 30–41, 2003.

[10] T. small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in Proc. ACM SIGCOMM’05 Workshop onDelay Tolerant Networking and Related Topics (WDTN-05).

[11] Y. Wang, S. Jain, M. Martonosi, and K. Fall, “Erasure-coding basedrouting for opportunistic networks,” in Proc. ACM SIGCOMM Workshopon Delay Tolerant Networking, 2005.

[12]Y. Wang and H. Wu, “Replication-based efficient data delivery scheme (RED) for DFT-MSN,” in Technical Report, 2005.

[13]M. J. Neely and E. Modiano, “Capacity and delay tradeoffs for ad-hoc mobile networks,” IEEE Trans. Inf. Theory, vol. 51, no. 6, pp. 1917–1937, 2005.

[14] D. B. Johnson, D. A. Maltz, and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)," IETF MANET, Internet Draft, 2003.

[15] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," presented at Proceedings of the SIGCOMM Conference on Communications, Architectures, Protocols and Applications, 1994.

[16] Hongyi Wu, Yu Wang, Ha Dang, and Feng Lin, "Analytic, Simulation, and Empirical Evaluation of Delay/Fault-Tolerant Mobile Sensor Networks,"IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 6, NO. 9, SEPTEMBER 2007

[17]Xin Wang, "mobile ad-hoc networks: protocol design, "Page336-354

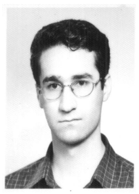
[18]M.C. Hsueh, T.K Tsai, R.K Iyer, “Fault Injection Techniques and Tools”, Computer, April 1997, pp.75-82

[19] D. Pradhan, Fault-tolerant computer system design. Prentice Hall Publisher, Englewood Cliffs, New Jersey, USA, 1996.

[20] M. Musolesi, S. Hailes, and C. Mascolo, “Adaptive routing for intermittently connected mobile ad hoc networks,” in Proc. IEEE 6thInternational Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM), pp. 1–7, 2005.

Authors

Reza Godaz



He was born in Mashhad, Iran, and received the B.Sc. and M.Sc. degree in software computer form Islamic Azad University of Najafabad, Iran, in 2002 and 2006. Since 2006, he has been a lecturer at the Islamic Azad University, Mashhad branch , Mashhad, Iran, which he joined in 2006.
His research interests are including Networks, Databases and Semantic Web.

Mohammad Reza Kaghazgaran

He was born in Mashhad-Iran and B.Sc. Software Computer Engineering of Islamic Azad University of Mashhad.
His research are Wireless Sensor Network and Cryptography and Security.