Sensors & Transducers, Vol. 218, Issue 12, December 2017, pp. 10-18

Sensors & Transducers

Published by IFSA Publishing, S. L., 2017
http://www.sensorsportal.com

A Novel Routing Fault Tolerant Reliable Protocol for Wireless Sensor Networks

[1] Islam MOURSY, [2] Mohamed ELDERINI and [3] Magdy AHMED

[1, 2] Computer and Systems Engineering Department, Faculty of Engineering, Alexandria University
Alexandria, 11432, Egypt

[3] Vice Dean for Graduate Studies and Research, Faculty of Engineering Alexandria University
Alexandria, 11432, Egypt

[1] Tel.: +20 10 0170 4152

E-mail: islam.moursy@alexu.edu.eg, elderini@alexu.edu.eg, magdy@alexu.edu.eg

Received: 10 November 2017 /Accepted: 10 December 2017 /Published: 29 December 2017

Abstract: Wireless sensor networks operate in very challenging environments that make them prone to different types of faults. Hence, there is a high need for a reliable protocol that offers an acceptable functionality in the presence of faults. In this paper, we propose the Fault Tolerant Reliable Protocol (FTRP), a novel routing protocol designed to be used in wireless sensor networks. FTRP offers fault tolerance reliability for packet exchange, as well as adaptation for dynamic network changes. The key concept in this protocol is the use of node logical clustering. The protocol delegates the routing ownership to the cluster heads, where the fault tolerance functionally is implemented. FTRP utilizes cluster head nodes along with cluster head groups as intermediate storage for transient packets. In addition, FTRP utilizes broadcast in its routing messages communication. This technique substantially reduces the message overhead as compared to classical flooding mechanisms. FTRP manipulates Time to Live (TTL) values for the various routing messages in addition to utilizing jitters in messages transmission. FTRP performance has been evaluated through extensive simulations. Aggregate Throughput, Packet Delivery Ratio and End-to-End delay have been used as performance metrics. The results obtained showed that FTRP ensures high Throughput, high Packet Delivery Ratio, and acceptable End-to-End delay in the presence of changing networking conditions. FTRP performs well in dense and sparse networks while nodes are mobile. Stationary simulations represented the worst-case behavior. This is attributed to synchronized nodes, where nodes send similar messages at the same time.

Keywords: Fault tolerance, Proactive routing, Wireless sensor networks, NS-3.

1. Introduction

Wireless Sensors Networks (WSNs) continue to present a lot of interest in both the research domain as well as the industry [1]. WSNs are highly adaptive to various domains, including - but not limited to - energy control systems, environmental monitoring, security, surveillance, health applications, area monitoring and Internet of Things [2].

Typical WSNs are networks composed of a large number of sensor nodes. Each node is equipped with sensors to detect various attributes of the surrounding environment. WSNs are built to operate for prolonged time and even in a hostile environment, which increases the need for fault tolerant reliable communication protocols [3].

There are many research papers on routing protocols. However, only few are adopted by the

industry. The Institute of Electrical and Electronics Engineers (IEEE) had adapted the topic and introduced Low-Rate Wireless Personal Area Network (Lr-WPAN) [4] as a standard Media Access Control (MAC) layer for WSNs, which opens a great opportunity for WSNs. This paper introduces a new fault tolerant reliable routing protocol for WSNs, which is efficient under mobility conditions.

Mahmoud, *et al.* [5] introduced a novel three-dimensional reference model for research in WSN reliability. The model categorizes WSN protocols into one of two techniques, which are retransmission or redundancy. Reliability is ensured within those techniques either by using a hop-by-hop or an end-to-end method to recover the lost data while maintaining either packet or event level reliability. Chouikhi, *et al.* [6] classify fault tolerance techniques according to the time at which the fault tolerance is triggered (before or after the fault occurrence). According to this, these techniques are classified as preventive or curative. Hence, the proposed protocol is classified as a proactive protocol that is retransmission based, connection oriented (end-to-end), with packet level reliability and utilizing a curative technique to achieve fault tolerance.

Fault Tolerant Reliable Protocol (FTRP) operates as a table driven proactive protocol [7]. FTRP regularly exchanges topology information with selected nodes of the network. Initially, nodes are in learning mode and broadcast a status of not being in a sensor domain in preparation to join one. If no answer is received, the nodes stay in that state until an answer is received. If an answer is received, the node evaluates the answer depending on its source and its included attributes. A cluster then begins to form according to the proposed protocol.

After cluster formation, Cluster Member (CM) nodes send data messages to their designated cluster head (CH). The CH, in turn, decides how many copies of the message to be retained until an acknowledgment (ACK) is received from the destination. The CH stores that message in the cluster head group (CHG) according to the protocol-defined parameters. The proposed protocol utilizes the following main techniques.

1.1. Retransmission-based Reliability

Retransmission is the traditional way of ensuring reliability [5]. This is achieved by allowing the sender node to wait for an ACK for its previously sent packets. In case, no ACK is received, the packet is considered lost and retransmission takes place to ensure reliability. FTRP implementation relieves the responsibility of packet storage and retransmission to higher entity nodes (CHs, CHGs or Sinks), as will be elaborated in Section III.

1.2. End-to-End Reliability

End-to-End reliability is a connection-oriented scheme for achieving reliability in which only the two communicating end nodes (source and destination) are responsible for ensuring reliability. FTRP implementation expands the end-to-end reliability by relieving the source node from this task, and transferring it to the CH. The CH determines, according to the replicas parameters, which CHGs to be used as storage. Whenever the destination node receives the packets, it broadcasts a message only processed by CHs or CHGs to release their locally stored corresponding replicas.

1.3. Packet Level Reliability

Packet level reliability ensures that all the packets carrying sensed data from all the related nodes are reliably transported to their destinations.

The rest of the paper is organized as follows. In Section 2, the most relevant related works are presented. In Section 3, the relevant FTRP protocol operations are detailed. The performance analysis of the FTRP protocol is presented in Section 4. Finally, Section 5 concludes the paper and lists ongoing and future work.

2. Related Work

In this section, we review literature work addressing the same elements as our protocol, namely retransmission based, connection oriented (end-to-end) and packet level reliability.

Reactive routing protocols such as Ad hoc On-demand Distance Vector (AODV) protocol [8] have the ability to discover the route when required. AODV uses the flooding mechanism to broadcast the route request to determine for new route during failure and can be very expensive to perform. AODV does not distinguish failure. It relies on the link layer feedback and the distance traverse by the packet to determine whether to broadcast for new route or drop the packet. As WSNs are prone to different failures with different durations caused by neighboring nodes, external radio devices, moving object and operating environments, nodes can suffer from transient, intermittent and permanent failure. Combinations of these failures may occur and may produce a complex unpredictable behavior that cannot be addressed with a single protocol. For example, transient failures may trigger the link layer to notify failure to the AODV and result in route discovery. When the next-hop neighbor experiencing the transient failure recovers, it will respond to the request while other nodes propagate the route request to all its local nodes. This will create a ripple effect that may congest the network. It is necessary to provide a reliable mechanism for the

nodes to change their routing strategy according to the current network topology in order to re-establish the network connection. This leads to a motivation to investigate the potential of a new fault tolerant routing protocol.

Protocols such as Sensor Protocols for Information via Negotiation, (SPIN) [9] were developed to allow querying the WSN for data without being able to address particular nodes and to implement energy savings at the same time. SPIN follows an interest advertisement-request strategy in which information is described by meta-data which initially is exchanged between the nodes. Nodes, which acquired new data, advertise it via its meta-data classification. Neighboring nodes, which have an interest in that kind of data, reply with a request, on which the advertising node transmits the data to the requesting node. After receiving the new data, the requesting node advertises it to its neighbors. SPIN achieves a high-energy efficiency compared to flooding, as only requested information is transported in the network [10]. However, there is no standard meta-data format, as this is supposed to be application specific. In addition, the delivery of data is not guaranteed by SPIN's advertisement mechanism, as the nodes interested in a specific class of data might be distant from the node acquiring this data. If intermediate nodes are not interested in the given class of data, the interested node will never receive it.

To address the issues of scalability and energy preservation in a different way, the notion of hierarchy was introduced in several WSN routing protocols with the goal of avoiding an overload of sink nodes by too many received messages, as well as reducing the amount of overall message transmissions. To achieve this, nodes are grouped into clusters, which feature a node designated as cluster head. Information is relayed to this cluster head, which aggregates data to bundle the information and reduce the number of messages, which are sent to the sink nodes. With this strategy, communication is forced into a multi-hop manner, relaying information over neighboring nodes, which in turn preserves energy as the energy cost of radio communication increases with the distance. Low Energy Adaptive Clustering Hierarchy (LEACH) [11] is one of the first routing protocols applying this strategy.

Iyer, *et al.* [12] proposed the Sensor Transmission Control Protocol (STCP), an end-to-end reliability protocol with a congestion control mechanism that is sink-centric. STCP dynamically controls the application data flow by utilizing a controlled variable reliability mechanism where the application type controls the throughput. Reliability is maintained by using ACK or Negative Acknowledgement (NACK) as end-to-end retransmission mechanisms. Packets are cached locally in each node until an ACK is received from Sink. Whenever Sink receives information about congested paths, the Sink directs the downstream-congested nodes to select alternative paths. Reliability in STCP is achieved through connection-oriented explicit ACKs, which involves only the end nodes. STCP is considered scalable for a large number of nodes with high hop counts from a source node to the Sink. STCP nodes are prone to huge end-to-end delay time [5], which results in high latency and cache overflow.

Marchi, *et al.* [13] proposed a Distributed Transport for Sensor Networks (DTSN). DTSN is non-sink centric, end-to-end and an energy oriented packet reliability protocol. DTSN is based on two mechanisms, full and differential reliability mechanisms. Full reliability is achieved via retransmission based explicit ACKs, while differential reliability is performed independently. In the full reliability mechanism, the source node keeps transmitting the packets until the number of transmitted packets equals the size of the acknowledgement window. An explicit acknowledgement request is issued from the source node to the destination to confirm message delivery. If the sequence of the packets is in order, an ACK is sent. These packets are then removed from the buffer of the source node. If a NACK is received then retransmission of the missing sequence of packets is performed. The key contribution of DTSN is the integration of mechanisms involved in achieving reliability, such as partial buffering at the source and intermediate nodes and the utilization of erasure coding. However, DTSN does not provide details on how the reliability level is maintained when network conditions change.

3. FTRP Operations

3.1. Protocol Overview

FTRP [14] operations utilize a simple messaging system to communicate different protocol statuses to the participating nodes. This messaging system is used to transition the node from one state to another in order to form a logical grouping of nodes referenced later as a cluster. FTRP tries to overcome the issues in STCP [12] by utilizing a distributed cache rather than preserving the cache at the sender node. This approach allows the cluster head to control the amount of cache allocated and where to store the data packet. FTRP introduces a retry count for locally cached entries. Whenever a packet entry reaches its max retry count, (the default is six retires), it is flushed out of the cache to overcome cache overflow. In fact, FTRP is well suited for a changing environment, where its messages update the network paths and handle nodes failure well.

FTRP communicates using a unified packet format for all data related to the protocol. This provides an easy way to combine different messages in a single packet transmission. These packets are encapsulated into User Datagram Protocol (UDP) [15] datagrams. On the other hand, FTRP messages contain a sequence number, which is incremented for each message. In such case, the recipient of a control message is able to

identify which information is more recent and to ignore those older unprocessed messages.

3.2. Definitions of Main Nodes Status

1) **Sink:** The Sink is the central node of the network, having information about all nodes. Usually, it is connected to a wired network and it has access to the wireless sensor domain.

2) **Cluster Head (CH):** The Cluster Head can be regarded as a Sink, but for a subset of nodes. It is responsible for relaying all information from and to the nodes controlled under its domain.

3) **Cluster Head Group (CHG):** CHGs are normal nodes selected by the CH as per the protocol parameters to act as local cluster storage for messages in transient.

4) **Cluster Member (CM):** CMs are normal nodes composing the cluster and are managed by the respective CH.

5) **Cluster Bridge Head (CBH):** If the CH is far away from the Sink, the CBH is the node within another cluster that links the cluster with the nearest CH.

6) **Learning:** Initially, a node is not in a cluster or it does not know route to a Sink.

7) **Swarm:** A node has identified another node that is not in its domain and it has knowledge of other nodes (nonsink).

8) **Discovered:** A discovered node is a node that is discovered from either a Sink or another cluster.

The life cycle begins with a node in a Learning state. A few nodes who have knowledge of their respective existence can form a swarm. Few swarm nodes can then transition to a discovered state upon sensing a nearby Sink. The Sink nominates a discovered node to be a CH. The CH can request nearby nodes for association as CMs. Few CMs can then be nominated as CHGs, as per the predefined configuration parameters of the protocol.

Fig. 1 depicts the state transition for nodes in FTRP.

3.3. FTRP Messaging System

FTRP inherits the general packet structure of the Optimized Link State Routing Protocol (OLSR) [16] where a packet header is appended to multiple FTRP messages and has a sequence number. The choice of OLSR packet format was made to benefit from piggybacking multiple types of messages in the same packet. A single FTRP packet can contain multiple routing messages. Messages share a common structure as well.

3.3.1. Hello Message (HELLO)

A nonsink node lifecycle begins in a Learning state, where it periodically broadcasts a hello message exposing its status and other parameters. Hello messages have their Time to Live (TTL) [17] value set to one, in order not to flood the whole network. A Hello message is populated with the sending node known attributes, and its known existing members, if any. Hello messages are broadcasted as keep alive periodically. The behavior of each node is different upon receiving a Hello message, according to the receiving node status.

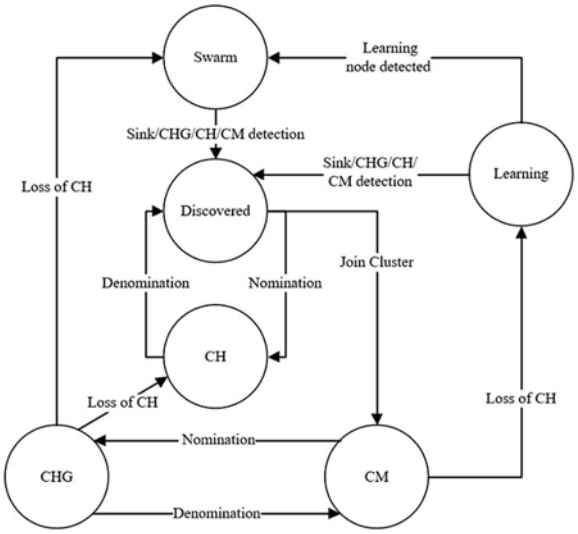

Fig. 1. FTRP State Transition Diagram.

A Sink node receiving a Hello message checks if the incoming node has not yet joined a domain, and if it is not a member of any other cluster. In that case, the Sink sends an association request. If the node had already been identified in a domain yet had not joined any cluster, the Sink will not take any action. This mechanism is adopted in order to control the allocation of CHs and to allow the network clustering formation to converge by favoring the node to join a cluster than to promote it to a new CH. The Sink will ignore any Hellos from other Sinks and will update the information received from any other CH.

3.3.2. Association Message (ASC)

ASCs are used to instruct nodes to join a cluster or domain. Only the Sink and the CHs are allowed to send association to other nodes. ASC messages have two classes.

1) **Regular association:** A regular association messages have their TTL value set to one, so that association does not flood the network.

2) **Broadcasted Association (ASCb):** A broadcasted association messages have their TTL value set to 255 in order for a CH to be nominated when it has no direct link to the Sink. It uses its nearest CBH to reach the Sink through the distress Save-Our-Ship (SOS) mechanism.

A node populates the ASC message with its members. Having that, members of a Sink are the CHs known to that Sink and members of a CH are those nodes under the CH control as fault tolerance domain.

3.3.3. Control Message (CTL)

CTLs are used as decision-making mechanism and out of band, status updates of different protocol aspects. It has the following subclasses

1) **Reject CH promotion:** Reject CH promotion is issued in the case when a Sink at some point in time decided to promote a CM to CH however, this CM was earlier acquired by another CH. In that case, rejecting the CH promotion is favored so that the CH ID pool is not depleted too fast. In return, the CM issues a Reject CH Promotion control message to notify the Sink to release the allocated CH ID.

2) **Members check:** A swarm node that was nominated to be CM or CH knows about the existence of other swam nodes whom with which a swarm was formed. This swarm must be checked against a high entity node (Sink in case the node is CH or CH in case the node is CM). The receiving node (Sink or CH) checks the incoming member list for local existence in its data structures, and then replies to the sender node with a "Release swarm members" message for those members the higher entity does not know about.

3) **Release swarm members:** When this message is received, the node drops the sending node from its local base as swarm, and sends them swarm release notify control message.

4) **Swarm release notify:** This message is processed by swarm to drop the sender from its local base.

5) **Swarm SOS:** Whenever the swarm is about to drop its last member, it issues swarm SOS to the sender of the release notify so that the sender is treated as bridgehead and relays the SOS to the Sink. The Sink will then send an ASCb, with its TTL value set to 255, to this swarm node to be nominated as new CH.

6) **Fault Tolerant message release (FT_Release):** Whenever a node successfully receives its data packet, it sends this message in broadcast mode, i.e., its TTL value set to 255, to notify CHs and CHGs to release the local copies of the messages considered for fault tolerance.

3.4. Message Emission and Jitter

To avoid synchronization of messages, jitter is introduced to allow protocol messages to be emitted such that they avoid synchronization. Emission of protocol messages from neighboring nodes may, for various reasons (mainly timer interactions with packet processing), become synchronized such that several neighbor nodes attempt to transmit messages simultaneously. This may or may not lead to collisions and hence message loss of several subsequent messages. To avoid synchronizations of messages, the following strategy is utilized. A node adds an amount of jitter to the interval at which messages are generated. The jitter is a random value for each message generated. Thus, for a node utilizing jitter:

$$FMI = OMI - \text{jitter}, \quad (1)$$

where *FMI* is the final message interval, *OMI* is the original message interval and jitter is a uniform random value selected from the interval [0, original message interval/4]. Jitter is also used when a message is to be forwarded by a node. The message is kept in the node during a short time period equals the jitter interval.

This scheme increases the opportunity to piggyback other messages in the same routing packet and contributes to the reduction of the overall number of packet transmissions.

3.5. Wrap-around

FTRP utilizes sequence number in packets and messages to be able to discard messages that are repeated or are received out of order. The limited number of bits (16 bit) for representing sequence numbers can cause repeated values to be present which is called a wrap-around (i.e. sequence number is incremented from the maximum possible value to zero). To be able to distinguish which sequence number is more recent. This recovery technique was inherited from OLSR [16] by defining the following:

$$seq = (seq + 1)\ mod\ (MaxValue + 1), \quad (2)$$

where seq is the Sequence Number and MAXVALUE is the maximum value that can be held in the number of bits defined. Thus, even in the presence of wrap-around, it is possible to determine which message contains the most recent information.

3.6. Routing Function and Fault Tolerance

The default forwarding scheme for a node is to direct the outgoing packets to its master (CH in case of a node, and a Sink in case of a CH). The scheme below also applies in case the CH or Sink is initiating a packet send. Upon the reception of a forward request, the routing function checks local parameters for replica count and then stores the message in the CHGs accordingly. Then, finally, the packet is forwarded normally.

CHs and CHGs are using a timed queue to store the packets. The receiving node, upon successful reception of a packet, generates an FT_Release message having the packet unique identification. Each receiving CHG, CH or Sink accepts this message and removes the requested message (if it exists) from its local queue. Upon the expiry of the queue timer, the local fault tolerance queue is checked for packets that

had not exceeded their retry time, and those packets are resent. Packets having expired retry time are removed from the queue and are considered undeliverable due to unreachable destination.

4. Simulation and Performance Evaluation

4.1. Assumptions

The simulation model is based on the following assumptions:

- The Sink has infinite power source, while nodes have not.
- Each node can behave as both a client and a router.
- Each node has a single interface running the FTRP protocol on that interface.
- The nodes have the same capabilities, i.e., same coverage area and same antenna.
- The nodes are randomly placed.
- The nodes follow a 2d-walk mobility pattern in mobility scenarios and follow a constant position model for stationary simulations.
- The nodes can either receive or transmit at a time.
- There is no turn around time between transmitting and receiving. Nodes can switch between transmit and receive instantly.
- Mobility is uncorrelated among the nodes and links fail independently.

4.2. Performance Metrics

The following performance metrics are used to analyze the behavior of FTRP.

1) Aggregate Throughput: This is the sum of the throughputs in the uplink and the downlink.

2) Packet Delivery Ratio (PDR): This is the number of successfully delivered packets divided by the total number of transmitted packets

3) End-to-End Delay (E-2-E): This is the sum of time taken for packets transmitted from sources to destinations divided by the total number of received packets.

4.3. Simulation Environment

The FTRP routing model is built using NS-3 network Simulator [18] on top of IEEE 802.11 MAC model of NS-3. Due to simulator limitations, model parameters have been tuned to match the 802.15.4 MAC layer.

The Random 2d-walk model [19] was adopted for driving mobile clients. In the Random 2d-walk mobility model, each instance moves with a speed and direction chosen randomly until either a fixed distance has been walked or until a fixed amount of time has passed. If a node hits one of the boundaries (specified by a rectangle) of the model, it rebounds on the boundary with a reflexive angle and speed. This model is often identified as a Brownian motion model. The speed is varied from no mobility using a constant position model, 1 m/sec to 2 m/sec. Table 1 depicts the parameters set for the simulation model that is common for all our simulations.

Table 1. Parameters for Simulation Model.

Simulation Parameter	Value
Simulator	NS-3 (version 3.25)
Operating system	Linux (Ubuntu 14.04)
Simulation time	50 secs
Simulation Area	100 m × 100 m
Number of nodes	20 for sparse, 40 for dense
Node transmission range	50 meters
Movement model (for mobility tests)	Random Walk 2d Mobility Model
Stationary model (for no mobility tests)	Constant Position Mobility Model
Nodes Position allocator	Random Disc Position Allocator
Speed of mobile nodes	1 m/sec and 2 m/sec
Traffic type	CBR
Data payload	512 bytes
Packet rates	20 p/sec to 80 p/sec
MAC Layer	802.11 DCF with RTS/CTS
Radio Frequency	2.4 GHz
Radio Channel rate	2 Mbps
Propagation loss model	Friis Propagation Loss Model
Propagation Delay Model	Constant speed propagation delay model

4.4. Results and Analysis

FTRP is simulated using various networking scenarios with the help of the NS-3 simulator. The scenarios and results along with detailed analysis are presented in the following sections.

4.4.1. Scenario I

In this scenario, we analyze the performance of FTRP in terms of throughput, PDR and E-2-E delay in a sparse network comprising of 20 nodes. The simulation is performed by varying the number of data packets sent per second, while maintaining a constant number of flows and system load. Number of packets per flow ranged from 20 packets/sec to 80 packets/sec. The simulation was repeated using no mobility model, 1 m/sec and 2 m/sec walking models. Other parameters considered for simulations are the same as shown in Table 1.

Fig. 2 depicts PDR against increasing traffic load in a sparse network. It is observed that increasing the data rate beyond 280 kb/s causes PDR to begin to drop, although not significant.

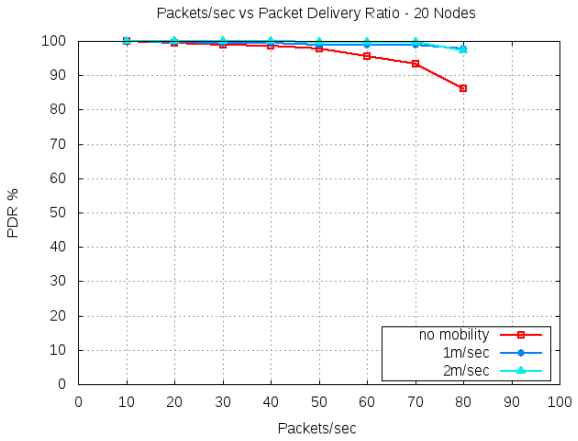

Fig. 2. PDR in a sparse network.

As per our simulation parameters, a data rate of 240 kb/s corresponds to 60 packets/sec and a data rate of 280 kb/s corresponds to 70 packets/sec. Mobile nodes achieve a good PDR with regard to the maximum data rate supported by Lr-WPAN [4] standard, which are 250 kb/s (approximately 63 packets/sec). While nodes are stationary, the obtained PDR results fall to above 94 % at the target data rate of 60 packets/sec, which is acceptable.

Fig. 3 depicts Aggregate throughput against increasing traffic load in a sparse network.

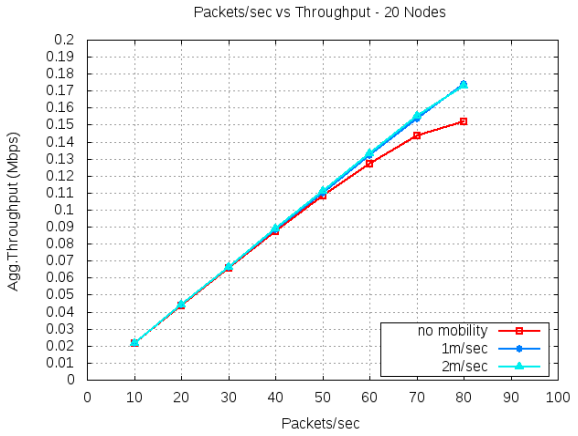

Fig. 3. Aggregate Throughput in a sparse network.

It is observed that the throughput increases as the data rate increases. Both low and high mobility scenarios achieve good throughput as data rate increases even for data rates above the targeted 250 kb/s. The stationary nodes performance is lower than that of mobile ones, which can be attributed to the nodes synchronized states.

Fig. 4 depicts E-2-E delay against increasing traffic load in a sparse network. It is observed that, as the data rate increases, the E-2-E delay increases significantly in a stationary scenario. The E-2-E delay increases within acceptable range for mobile scenarios.

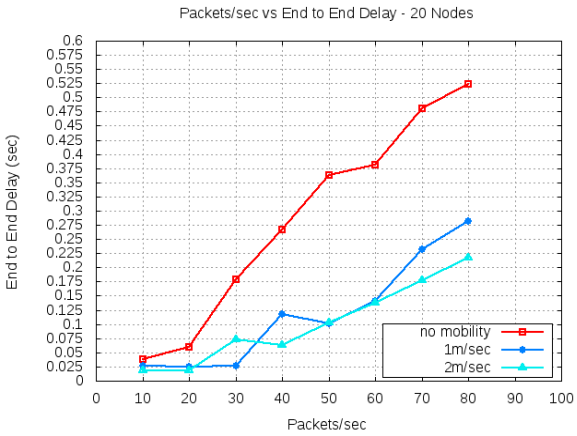

Fig. 4. End to End Delay in a sparse network.

The increase in E-2-E delay is expected due to the introduction of fault tolerance mechanism, which uses store and forward. In the stationary scenario, the increase is significant and can be justified by the nature of FTRP being too communicative. In the stationary scenario, the collision rate of packets can increase, while mobility helps to decrease collision. This can be attributed to the variations of node states. This variation reduces messages exchanged, reduces collisions and maintains good E-2-E delay.

4.4.2. Scenario II

In this scenario, we analyze the performance of FTRP in terms of throughput, PDR and E-2-E delay in a dense network composed of 40 nodes. The simulation is performed by varying the number of data packets sent per second, while maintaining a constant number of flows and system load. The number of packets varied per flow ranged from 20 packets/sec to 80 packets/sec. The simulation was repeated using no mobility model, 1 m/sec and 2 m/sec walking models. Other parameters considered for simulations are the same as depicted in Table 1. Scenario II results emphasize the results of scenario I. It is found that in a dense network with no mobility, PDR drops, Aggregate Throughput tends to saturate early and E-2-E delay increases significantly. In mobility scenarios, PDR is within acceptable ranges at 70 packet/sec rate, the Aggregate Throughput increases and E-2-E delay is within acceptable ranges.

Fig. 5 depicts PDR against increasing traffic load in a dense network.

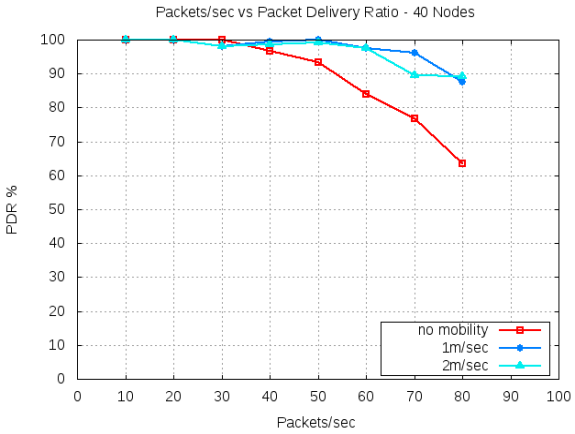

Fig. 5. PDR in a dense network.

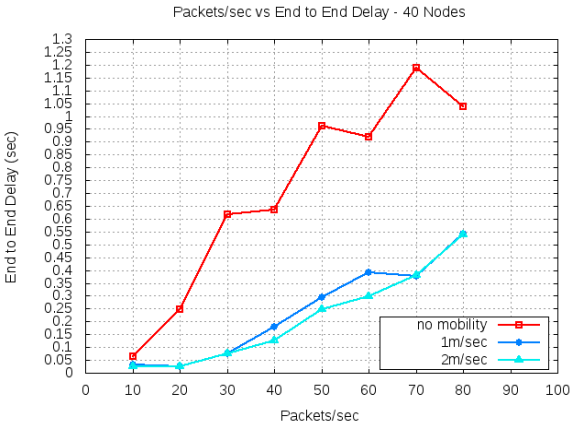

Fig. 7. End to End Delay in dense network.

It is observed that, while nodes are mobile, the PDR is almost the same. However, for data rates higher than 260 kb/s (65 packets/sec) higher mobility nodes PDR tends to saturate while for less mobile nodes PDR tends to decrease. Stationary nodes are the worst performer, result which can be attributed to synchronized nodes states.

Fig. 6 depicts Aggregate Throughput against increasing traffic load in a dense network. It is observed that, while nodes are mobile, the throughput is almost the same. However, for data rates higher than 250 kb/s, higher mobility nodes' throughput tends to increase while for less mobile nodes throughput tends to saturate. Stationary nodes are the worst performer, which can be attributed to synchronized nodes states.

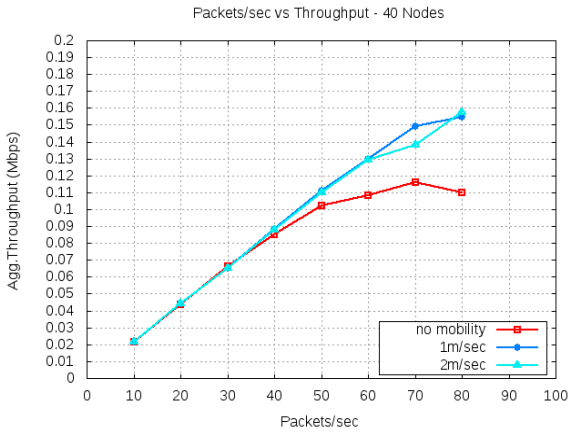

Fig. 6. Aggregate Throughput in a dense network.

Fig. 7 depicts E-2-E delay against increasing traffic load in a dense network. It is observed that, while nodes are mobile, E-2-E is almost the same and for data rates higher than 250 kb/s all mobile nodes' E-2-E tends to increase. Stationary nodes are the worst performer, which is directly linked to the fault tolerance function, in which, for every sent packet, an ACK for reception is needed to consider a packet is delivered. This increases the time when a packet is considered successfully delivered.

The ACK packet as well might get lost due to network collisions and synchronized nodes states, which in turn will cause the source node to resend the packet and wait for another ACK. This significantly affects the E-2-E delay for FTRP.

5. Conclusions

This paper introduced a novel reliable fault tolerant routing protocol, FTRP, for wireless sensors networks. FTRP creates a communication path between source and destination nodes and forwards packets on that path.

FTRP performance has been evaluated through extensive simulations using NS-3. Aggregate throughput, Packet Delivery Ratio and End-to-End delay have been used as performance metrics. In terms of Packet Delivery Ratio and Aggregate throughput, FTRP is an excellent performer in all mobility scenarios, whether the network is sparse or dense. In stationary scenarios, FTRP performed well in sparse networks; however, in dense networks, FTRP's performance had degraded, still remaining in an acceptable range. In terms of End-to-end delay, FTRP is considered a good performer in all mobility scenarios where the network is sparse. In the sparse stationary scenario, FTRP is still considered a good performer. However, in dense stationary scenarios, FTRPs performance is considered as worst-case behavior, which can be attributed to synchronized nodes states that occur when nodes send similar messages at the same time.

There are times when properly receiving a network message carrying crucial information is more important than other costs, such as, but not limited to, energy or delay. That makes FTRP suitable for a wide range of WSNs application domains, such as military applications by monitoring soldiers' biological data and supplies while on the battle field as well as battle damage assessment. FTRP can also be used in health applications by tracking and monitoring doctors and patients inside a hospital and elderly assistance, in addition to a wide range of geo-fencing, environmental monitoring, resource monitoring,

production lines monitoring, agriculture and animals tracking.

FTRP should be avoided in dense stationary deployments such as, but not limited to, scenarios where a high application response is critical and life endangering, such as biohazards detection or within intensive care units.

As future work, it is planned to improve the performance of FTRP in stationary scenarios. FTRP performance was evaluated through simulations. It is planned to extend the FTRP implementation in a WSN operating system to compare the complexity of a real system against the simulation results. The effect of varying the number of attempts to retransmit a non-delivered packet (max retry count) should be investigated. Furthermore, the energy efficiency has to be evaluated for various FTRP operations.

References

[1]. A. Ajith Kumar S., K. Vsthus, L. M. Kristense, An Industrial Perspective on Wireless Sensor Networks - A Survey of Requirements, Protocols and Challenges, *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 3, 2014, pp. 1391-1412.

[2]. C. Alcaraz, P. Najera, J. Lopez, R. Roman, Wireless Sensor Networks and the Internet of Things Do We Need a Complete Integration, in *Proceedings of the 1th International Workshop on the Security of the Internet of Things*, 2010, pp. 32-37.

[3]. S. Misra, I. Woungang, S. C. Misra, Guide to Wireless Sensor Networks, *Springer*, London, 2009.

[4]. IEEE Standard for Low-Rate Wireless Networks, 2016. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4-2015.pdf. [Accessed 7 7 2017].

[5]. M. A. Mahmood, W. K. Seah, I. Welch, Reliability in Wireless Sensor Networks: A Survey and Challenges, *The International Journal of Computer and Telecommunications Networking*, Vol. 79, 2015, pp. 166-187.

[6]. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudanec, L. Azouz Saidane, A Survey on Fault Tolerance in Small And Large Scale Wireless Sensor Networks, *The International Journal for the Computer and Telecommunications Industry*, Vol. 69, 2015, pp. 22-37.

[7]. Basu Dev Shivahare, Charu Wahi, Shalini Shivhar, Comparison of Proactive and Reactive Routing Protocols in Mobile Adhoc Network Using Routing Protocol Property, *International Journal of Emerging Technology and Advanced Engineering*, Vol. 2, No. 3, 2012, pp. 356-359.

[8]. C. E. Perkins, E. M. Royer, Ad-hoc On-Demand Distance Vector Routing, in *Proceedings of the WMCSA*, 1999.

[9]. J. Kulik, W. Rabiner, H. Balakrishnan, Adaptive Protocols for Information Dissemination in Wireless Sensor Networks, in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, USA, pp. 174-185, 1999.

[10]. K. Akkaya, M. Younis, A Survey On Routing Protocols For Wireless Sensor Networks, Ad Hoc Networks, Vol. 3, No. 3, 2005, pp. 325-349.

[11]. M. J. Handy, M. Haase, D. Timmermann, Low Energy Adaptive Clustering Hierarchy With Deterministic Cluster-Head Selection, in *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network*, Stockholm, Sweden, 2002, pp. 368-372.

[12]. Y. Iyer, S. Gandham, S. Venkatesan, STCP: A Generic Transport Layer Protocol for Wireless Sensor Networks, in *Computer Communications and Networks*, San Diego, CA, USA, pp. 35-42, 2005.

[13]. B. Marchi, A. Grilo, M. Nunes, DTSN: Distributed Transport for Sensor Networks, in *Proceedings of the 12th IEEE Symposium on Computers and Communications*, Las Vegas, NV, USA, 2007, pp. 165-172.

[14]. I. Moursy, M. El-Derini, M. Ahmed, FTRP: A Fault Tolerant Reliable Protocol for Wireless Sensor Networks, in *Proceedings of the 11th International Conference on Sensor Technologies and Applications (SENSORCOMM' 17)*, 10-14 September 2017, Rome, Italy, pp. 24-30.

[15]. RFC 768 - User Datagram Protocol - IETF, [Online], Available: https://www.ietf.org/rfc/rfc768.txt [Accessed 11 5 2017].

[16]. RFC 3626 - Optimized Link State Routing Protocol, IETF. [Online]. Available: https://www.ietf.org/rfc/rfc3626.txt [Accessed 11 5 2017].

[17]. RFC 791 - Internet Protocol, IETF, [Online], Available: https://tools.ietf.org/html/rfc791 [Accessed 07 07 2017].

[18]. T. Henderson, "NS-3 Overview," [Online], Available: http://www.nsnam.org/docs/ns-3-overview.pdf [Accessed 11 5 2017].

[19]. NS-3 Manual, [Online], Available: https://www.nsnam.org/docs/manual/ns-3-manual.pdf [Accessed 11 5 2017].