ISSN:0975-9646

Sabareesan M et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3503-3507

Overall performance of Flooding Sequence Protocol in Wireless Sensor Networks

Sabareesan M , Srinivasan S* ,John Deva Prasanna D S

Department of Computer Science and Technology, Hindustan University, Chennai, India

***Abstract*—In this project we are going to study about the performance of family of four flooding sequence protocols in wireless sensor networks using simulation and analyze which protocol having better performance compared to other three protocols.. Flooding is a communication technique that can be used by the base station of the sensor network requires to send a message to each and every sensors in the network. When a sensor receives the flood message, it needs to check whether it receives for first time or not. If it received for first time it is fresh message or it already exists means the message is redundant. If the sensor receives the fresh message it will alert the base station to evacuate the people to safer place. To avoid the redundancy, the family of flooding sequence protocols are designed to distinguish the fresh message and redundant message. The family of flooding sequence protocol are sequence free protocol, linear sequence protocol, circular sequence protocol and differentiated sequence protocol. We analyze the stabilization properties of four protocols and compared the each performance of protocols using simulation, using over settings of sensor networks.**

***Key terms*—sensor networks, stabilization property, flooding sequence protocol, sequence number**

I. INTRODUCTION

A sensor[1] is a battery operated small device with an antenna and a sensing board that can able to sense sound, heat, magnetism, vibration, pressure, environmental and physical changes in the environment. Sensors in a network can communicate and collaborate with other sensors to complete the given tasks.

Flooding[2] is a communication technique that can be used by the base station of a sensor network needs to send a copy of flood message to each and every sensor in the network. The execution of a flood starts by the base station sending a message to all its neighbours. when a sensor receives a flood message the sensor needs to verify, whether it receives the message for the first time or not. If the sensor receives the message for the first time, the sensor keep the copy of the message and forward that message to all its nearest neighbours.

When a sensor receives a flood message for the first time it is fresh message or it received already means the message is redundant. If the sensor networks receives the fresh message it will alert the base station to evacuate the people to the safer place. If the sensor receives the redundant message it will be discarded.

To distinguish between the fresh message and redundant message, the sensor networks selects the sequence number, attached to every messages sequentially before the base station broadcasts the message. When a sensor receives a flood message, the sensor determines based on the sequence number in the received message if the message is fresh or redundant. The sensor accepts the message if it is fresh and discards the message if it is redundant. The family of flooding sequence protocol that uses sequence numbers to distinguish between fresh flood messages and redundant flood message.

The redundancy can be occurred due to some faults or corrupts in sensor networks. To avoid redundancy we designed the family of four flooding sequence protocol to distinguish the fresh flood message and redundant flood message, attached with sequence number. The family of flooding sequence protocol[2] are sequence free protocol, linear sequence protocol, circular sequence protocol and differentiated sequence protocol. we analyze the stabilization properties of flooding sequence protocol and also comparing the performance of four flooding sequence protocol using simulation, using various settings of wireless sensor networks.

The stabilization properties[3],[10] of the flooding sequence protocols are useful for sensor network designers to select a proper flooding sequence protocol that satisfies the needs of a target sensor network.

We proposed a new protocol named Flooding Time Synchronization Protocol[4] can be used to maintain the performance of sensor by measuring its battery consumption, bandwidth. The Flooding Time Synchronization Protocol (FTSP), especially modified for applications requiring severe precision on resource limited wireless platforms. The proposed FTSP uses low communication bandwidth and it is robust against node and link failures. The FTSP achieves its robustness by utilizing periodic flooding of synchronization messages, and implicit dynamic topology update.

II. MODEL OF SENSOR NETWORKS

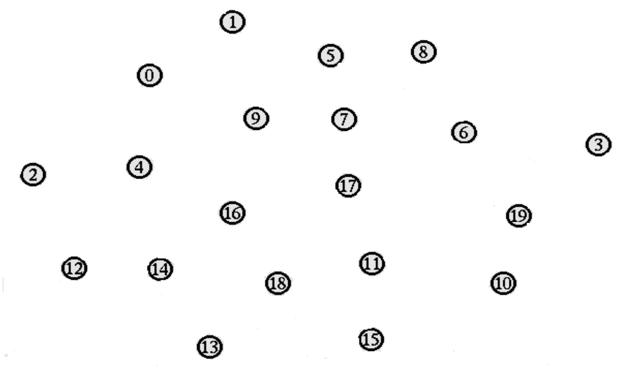


Fig 1 Model of Sensor Networks

The above figure 1 shows the model of sensor networks with some nodes. Using simulation the sensor network is created, the routing is created by AODV(Adhoc On Demand Distance Vector) routing protocol[5]. The use of AODV routing is to create a route whenever the sensor needs to communicate. It send a route request to the nearest sensors in the network, it will send each and every sensors in the network. If the route reply from the sensor it will adopt the route between the sensors. The flooding concept will be implemented to communicate the sensors in the networks.

The topology of a sensor network[6] is a directed graph where each node represents a distinct sensor in the network and where each directed edge is labelled with some probability. A directed edge (*u*,*v*), from a sensor u to a sensor *v*, that is labelled with probability p (where p > 0) indicates that if sensor u sends a message, then this message arrives at sensor *v* with probability p (provided that neither sensor *v* nor any “neighbouring sensor” of *v* sends another message at the same time).

If the topology of a sensor network has a directed edge from a sensor u to a sensor *v*, then u is called an in-neighbour of v and v is called an out-neighbour of u.

III. PREVIOUS PROBLEM

A sensor[1] is a battery operated small device with an antenna used to converts the signal which can be read by a observer or an instrument and a sensing board that can able to sense sound, heat, magnetism, vibration, pressure, environmental and physical changes in the environment. Sensors in a network can communicate and collaborate with other sensors to complete the given tasks.. A sensor network[1] is bare to various dynamic factors and faults, such as topology changes, energy saving features, unreliable communication, and hardware or software failures.

Flooding[2] has several significant uses in sensor networks. In the base station of a sensor network needs to reset the network, and it uses flood to send a reset message to every sensor in the network requesting that each sensor resets itself upon receiving the message.

When a fault occurs in the sensor networks (due to sensor failure, loss of battery power) it will corrupts the sequence number stored in the sensor, it results the network will become an illegitimate state, so the sensor will accept the redundant message and discards the fresh flood message. If the sensor accept the redundant message, the base station will be alerted by the sensor network.

In the sensor networks ,the base station needs to pass some data message to some (not necessarily all) sensors in the network. In this case, the base station uses flooding to send the data message to all the sensors in the network, but name in the message those sensors that should find the message relevant. It results collisions within the messages, collisions of consecutive message and redundant forwarding.

IV. PROPOSED SYSTEM

In this project we proposed the protocols are Flooding Sequence Protocol and Flooding Time Synchronization Protocol(FTSP).

The family of flooding sequence protocol[2] are designed to prevent the sensor network that cannot reaches the illegitimate state due to some fault. This family of protocol uses the sequence number to distinguish the fresh message and redundant message. The above protocol guaranteed to converge back it from illegitimate state to legitimate state, then every sensor accept fresh message and discard redundant message.

This models (four protocols) of sensor networks is conceptual, but it accommodates characteristics of sensor networks such as unavoidable local broadcast, probabilistic message transmission, asymmetric communication, message collision and timeout actions and randomization steps.

The term flooding[7] is that to transmit the message to each and every sensors in the network, each node act as a transmitter and receiver. Each node keep the message until it reaches the destination. This concept is used to send the fresh message to each and every node in the network, if one sensor may faulted it fails to send message. But using flooding concept each sensor must act as a transmitter and receiver, so it may forward the message to the base station. It results no redundant forwarding.

The proposed Flooding Time Synchronization Protocol (FTSP)[4], particularly modified for applications requiring rigorous precision on resource limited wireless platforms. This protocol uses low communication bandwidth and it is robust against nodes and link failures. The FTSP achieves its robustness by utilizing periodic flooding of synchronization messages, and implicit dynamic topology update.

The FTSP protocol can be used to reduce the link failures occurred between the nodes, it uses low bandwidth so the battery power consumption is decreased. In sensors the battery power is not guaranteed for lifetime, so this protocol can increase the lifetime of sensors.

V. IMPLEMENTATION OF AODV ALONG WITH FLOODING

Adhoc On Demand Distance Vector Routing protocol[5] is a routing protocol for Adhoc networks such as wireless sensor networks, Mobile Adhoc Networks(MANET). It is a reactive routing protocol, it establishes a connection based on demand only. AODV avoids the counting infinity problem of other distance-vector protocols by using sequence numbers on route updates. AODV can be used in both unicast and multicast routing.

In AODV the routes are created when they are demanded by source node to reach destination node. The source made a route request to its neighbour to reach the destination, if the route reply send by the neighbours node it will made a route to the destination node.

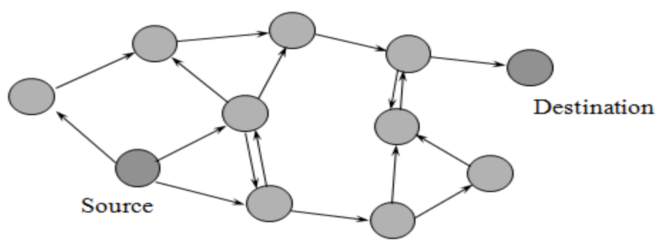


Fig 2 AODV Routing on sensor networks

The above figure 2 shows the AODV routing on sensor network, it shows the route path from source to destination.

The flooding[7] is a technique, it can be used to transmit the message each and every node in the network, it takes more time to reach the destination, the AODV routing is implemented along with flooding, to reduce the time and redundancy. For example the source node needs to transmit the message to destination, in flooding it will transmit the message to its neighbour. Each node will be act as a transmitter and receiver, so if one node fails to send the message, the another node forward the message to the destination. With the help of AODV routing whenever the node needs a route, it will create a route on demand. If suppose one node drops the packet the error message came to the source, so it will create another path to reach the destination. With the implementation of AODV along with Flooding reduces time, it consumes less battery power so the sensor lifetime is increased, uses low bandwidth and reducing redundancy.

VI. OVERVIEW OF FLOODING SEQUENCE PROTOCOL

In this section, we give an overview of a flooding protocol[8],[10],[11] that is used with our flooding sequence protocols. They are Sequence free protocol, Linear sequence protocol, Circular sequence protocol and Differentiated sequence protocol. This family of protocols are mainly designed to reduce redundancy occurs due to an illegitimate state of the sensor network.

Consider a network that has n sensors. In this network, sensor 0 is the base station and can initiate floods over the network. To initiate the flood message, sensor 0 sends a message of the form data(hmax), where hmax is the maximum number of hops to be made by this data message in the network.

If sensor 0 initiates one flood message and shortly after initiates another flood message, some forwarded messages from these two floods can collide with one another causing many sensors in the network not to receive the message of either flood, or (even worse) not to receive the messages of both floods.

To prevent message collision across consecutive flood messages, once sensor 0 broadcasts a message, it needs to wait enough time until this message is no longer forwarded in the network, before broadcasting the next message. The time period that sensor 0 needs to wait after broadcasting a message and before broadcasting the next message is called the flood period. The flood period consists of f time units. Thus, after sensor 0 broadcasts a message, it sets its timeout to expire after f time units in order to broadcast the next message.

When a sensor receives a data message, the sensor needs to decide whether the sensor accept or discard the message and forwards it as a data. To reduce the probability of message collision, any sensor *u*, that decides to forward a message, chooses a random period whose length is chosen consistently from the range and assumption of time period, and sets its timeout to expire after the chosen random period, so that *u* can forward the received message at the end of the random period. This random time period is called the forwarding period.

A. First Protocol Sequence Free

In this section, we discuss a first flooding sequence protocol where no sequence numbers are attached to each flood message, and so a sensor cannot distinguish between fresh and redundant flood messages, resulting that the sensor accepts every received message. This protocol is called the sequence free protocol.

B. Second Protocol Linear Sequence

In this section, we discuss a second flooding sequence protocol where each flood message carries a unique sequence number that is linearly increased, and so a sensor accepts a flood message that has a sequence number larger than the last sequence number accepted by the sensor. This protocol is called the linear sequence protocol.

Whenever sensor 0 broadcasts a new message, sensor 0 increases the sequence number of the last message by one, and attaches the increased sequence number to the message.

C. Third Protocol Circular Sequence

In this section, we discuss a third flood sequencing protocol where each flood message carries a sequence number that is circularly increased within a limited range, and so a sensor accepts a flood message that has a sequence number logically larger than the last sequence number accepted by the sensor. This protocol is called the circular sequence protocol.

Whenever sensor 0 broadcasts a new message, sensor 0 increases the sequence number of the last message by one circularly within the range.

D. Fourth Protocol Differentiated Sequence

In this section, we discuss the last flood sequencing protocol where the sequence numbers of flood messages are in a limited range, similar to the circular sequencing protocol. However, in this protocol, a sensor accepts a flood message if the sequence number of the message is different from the last sequence number accepted by the sensor. This protocol is called the differentiated sequence protocol.

VII. PROBLEM IMPLEMENTATION

In this section we discuss about the comparison of four family of flooding sequence protocol and find out which protocol is better for communication compared to other three protocols. We use this model to specify our flooding sequence protocols, verify the stabilization properties of these protocols and develop our simulation of these protocols.We analyze the stabilization properties and compare the stabilization properties with each protocol and produce the result by using simulation.

Stabilization Property

.

A self-stabilizing property[3],[10] is guaranteed to return to a legitimate state from illegitimate state where it performs some proposed function correctly, when some dynamic factors or faults corrupt the sensor. The stabilization properties of the flooding sequence protocols are useful for sensor network designers to select a proper flooding sequence protocol that satisfies the needs of a target sensor network. Therefore, in order to make a sensor network flexible to dynamic factors and faults, each protocol in the sensor network should be self-stabilizing.

We design stabilization property for sensor network protocols, prove their self-stabilization properties, and estimate their performance using simulation. These stabilization properties are a sentry-sleeper protocol that guard a group of sensors at the beginning of each time period, a logical grid routing protocol that builds a routing tree whose root is the base station, and a family of flooding sequence protocols can be used to differentiate fresh and redundant flood messages using sequence numbers.

The following are the stabilization properties they are

a. Time synchronization

b. Radio Message Propagation

A. Time Synchronization

The need of time synchronization is to create a common reference point between nodes and its neighbours. The goal of the FTSP[4] is to attain a network wide synchronization of the local clocks of the participating nodes. We assume that each node has a local clock exhibit the typical timing errors of nodes and can communicate over an unreliable but error corrected wireless link to its neighbors. The FTSP synchronizes the time of a sender to perhaps multiple receivers make use of a single radio message time-stamped at both the sender and the receiver sides. MAC layer time-stamping can abolish many of the errors, as observed.

The FTSP[4] time-stamping efficiently reduces the jitter of the interrupt handling and encoding or decoding times by recording multiple time stamps both on the sender and receiver sides. The time stamps are made at each byte boundary after the SYNC bytes as they are transmitted or received.

B. Radio Message Propagation

a. Send time: Time used to gather the message and send request to the MAC layer on the transmitter side. The time taken to send a message to each and every nodes in the sensor network.

b. Access Time: Time taken to access the message and delay incurred waiting for access to the transmit channel up to the point when transmission begins.

c. Transmission Time: The time it takes for the sender to transmit the message to the receiver.

d. Propagation Time: The time it takes for the message to transmit from sender to receiver.

e. Reception Time: The time taken for the receiver to receive the message. The transmission and reception times overlap in Wireless Sensor Networks as pictured in Figure 3.

f. Receive Time: Time to progress the incoming message and to notify the receiver application.

g .Interrupt Handling Time: The delay between the radio chip raising and the microcontroller responding to an interrupt

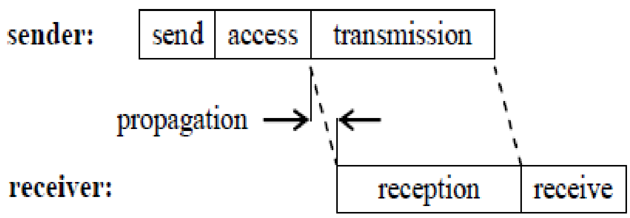
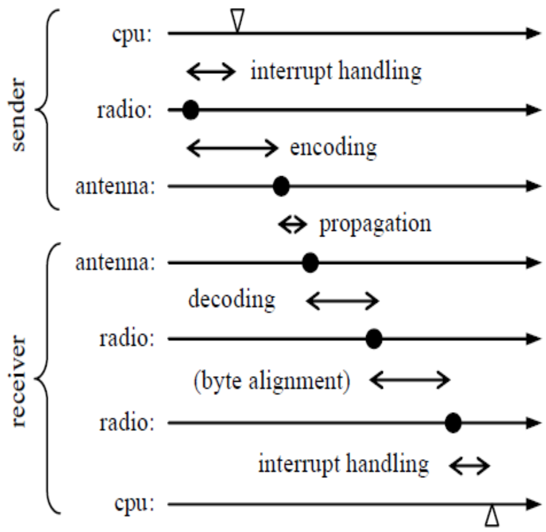


Fig 3 Transmission and reception time overlap

h. Encoding Time: The time taken by the sender to encode the message to decrypt form.

i. Decoding Time: The time taken by the receiver to transform and decode the message into binary representation.

j. Byte Alignment Time: The delay incurred because of different byte alignment of the sender and receiver.



XIII. PERFORMANCE EVALUATION

The performance of a flood sequencing protocol can be measured by the following two metrics:

a. Reach: The percentage of sensors that receive *a* message sent by sensor 0.

b. Communication: The total number of messages forwarded by all sensors in the network.

XI. CONCLUSION

In this paper, we discussed a family of the four flooding sequence protocols that use sequence numbers to distinguish between fresh flood messages and redundant flood messages. The members of our family are the sequence free protocol, the linear sequence protocol, the circular sequence protocol, and the differentiated sequence protocol. We analyzed the stabilization and stable properties of these four protocols, and also studied their performance, using simulation, over various settings of sensor networks. It was mentioned that the flood period is computed to guarantee that no two consecutive flood messages ever collide with each other. Thus, in practice, the half (or even less) of the flood period may be used without significantly degrading the stabilization property and performance of a flooding sequence protocol. Moreover, each of the flooding sequence protocols can be extended to support multiple floods within one flood period.

REFERENCES

[1] website http://www.wikipedia.org/sensor
[2] Young-ri Choi, Mohamed G. Gouda, *"Disciplined Flood Protocols in Sensor Networks"*
[3] M. Gouda and Y. Choi. A State-based Model of Sensor Protocols. In *Proceedings of 9th International Conference on Principles of Distributed Systems (OPODIS 2005)*, December 2005.
[4] Elson, J. E. *Time Synchronization in Wireless Sensor Networks.* Ph.D. Thesis, University of California, Los Angeles 2003.
[5] C. E. Perkins and E. M. Royer, "The ad hoc on-demand distance vector protocol," in Ad hoc Networking, Addison-Wesley, pp. 173–219, 2000.
[6] B. R. Bellur and R. G. Ogier. A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks. In *INFOCOM*, pages 178–186, 1999.
[7] website http://www.wikipedia.org/flooding
[8] D. Ganesan, B. Krishnamurthy, A. Woo, D. Culler, D. Estrin, and S. Wicker. An Empirical Study of Epidemic Algorithms in Large Scale Multihop WirelessNetworks. *IRP-TR-02-003*, 2002.
[9] J. Li and P. Mohapatra. A Novel Mechanism for Flooding Based Route Discovery in Ad Hoc Networks. In *Proceedings of the IEEE Global Telecommunications Conference*, 2003.
[10] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2003)*, 2003.
[11] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE Transac- tions on Parallel and Distributed Systems*, 13(1):14– 25, January 2002.