FLOODING TIME SYNCHRONIZATION PROTOCOL IN WIRELESS SENSOR NETWORK

D. Brintha Sweetly (Assistant professor)¹, P. Joselin Vinisha (Assistant professor)²
¹,²Department of computer science
Nanjil Catholic College of Arts and Science, Kaliyakavilai, K.K.Dist

Abstract: The accurate and efficient operation of many applications and protocols in wireless sensor networks require synchronized notion of time. To achieve network-wide time synchronization, a common strategy is to flood current time information of a reference node into the network, which is utilized by the de facto time-synchronization protocol Flooding Time-Synchronization Protocol (FTSP). In FTSP, the propagation speed of the flood is slow because each node waits for a given period of time to propagate its time information about the reference node. It has been shown that slow-flooding decreases the synchronization accuracy and scalability of FTSP drastically. Alternatively, rapid-flooding approach is proposed in the literature, which allows nodes to propagate time information as quickly as possible. However, rapid flooding is difficult and has several drawbacks in wireless sensor networks. In this paper, our aim is to reduce the undesired effect of slow flooding on the synchronization accuracy without changing the propagation speed of the flood, our main contribution is to show that the synchronization accuracy and scalability of slow-flooding can drastically be improved by employing a clock speed agreement algorithm among the sensor nodes.

Keywords: Time Synchronization, Flooding Time Synchronization Protocol, Slow Flooding, Rapid-Flooding, Propagation Speed, Clock Speed Agreement Algorithm.

I. INTRODUCTION

Sensor nodes in wireless sensor networks (WSNs) are equipped with cheap hardware clocks which frequently drift apart due to their low-end quartz crystals. Since the drift can be different for each sensor node, the hardware clocks of the nodes may not remain always synchronized although they might have been synchronized when they are started up. Lack of synchronized time leads to inaccurate and inefficient operation of many applications and protocols in WSNs. Hence, a time synchronization protocol is required so that all nodes exchange their time information to synchronize their clocks for minimizing their synchronization error, i.e. clock skew.

A common method in order to achieve networkwide time synchronization in WSNs is to flood current time information of a reference node into the network. Flooding Time Synchronization Protocol (FTSP), the de facto time synchronization protocol in WSN world, utilizes this method by allowing nodes to propagate their time information about the reference node after waiting for a given period of time. It has been shown that slow-flooding decreases the synchronization accuracy and scalability drastically due to the waiting times at each Node. In order to prevent these side effects, PulseSync offers rapidflooding by allowing nodes to propagate their time information quickly and reliably, and hence increasing the propagation speed of the flood. Multicast is an efficient method for transmitting data from a single source to several destinations. Especially, in wireless networks using a broadcast medium, a single transmission can be received by all nodes within a transmission range, which makes it easy to implement the multicast. The broadcasting medium, however, makes the wireless network vulnerable to various security attacks since anyone can easily eavesdrop on messages transmitted in the air. In order to implement the multicast, i.e., the delivery of data only to the members of a group, in wireless networks, we need to an access control mechanism for the
broadcasted messages, which guarantees confidentiality, protects digital contents, and facilitates accurate accounting. These multicast services in wireless networks.

II. RELATED WORKS

In FTSP, a dynamically elected reference node periodically floods the value of its clock (current time information) into the network. Using the flooded information, each slave node uses least-squares regression to establish a linear relationship between its hardware clock and the clock of the reference node. Using this relationship, each sensor node can predict future clock values of the reference node without communicating frequently. The nodes broadcast their predicted clock values of the reference node to their neighbors for the synchronization of the whole network. However, the predicted time information is not broadcasted quickly upon receiving a new synchronization message. Instead, each node waits until the expiration of its broadcast period. In [4] The Flash flooding protocol avoids this problem by allowing concurrent transmissions among neighbor nodes. It relies on the capture effect to ensure that each node receives the flood from at least one of its neighbors, and introduces new techniques to either recover from or prevent too many concurrent transmissions. In [5] The FTSP achieves its robustness by utilizing periodic flooding of synchronization messages, and implicit dynamic topology update. The unique high precision performance is reached by utilizing MAC-layer time-stamping and comprehensive error compensation including clock skew estimation. In [6] Having access to an accurate time is a vital building block in all networks; in wireless sensor networks even more so, because wireless media access or data fusion may depend on it. In contrast, we present PulseSync, a new clock synchronization algorithm that is asymptotically optimal. Larger networks, PulseSync offers an accuracy which is several orders of magnitude better than FTSP e.g. media access and local skew.

III. FLOODING TIME SYNCHRONIZATION PROTOCOL

Wireless sensor network applications, similarly to other distributed systems, often require a scalable time synchronization service enabling data consistency and coordination. This paper describes the Flooding Time Synchronization Protocol (FTSP), especially tailored for applications requiring stringent precision on resource limited wireless platforms. The proposed time synchronization protocol 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. Although rapid flooding strategy improves the synchronization accuracy and scalability of flooding based time synchronization dramatically, it has several drawbacks. First, rapid-flooding in WSNs can also be slow due to neighborhood contention because the nodes cannot propa-gate the flood until their neighbors have finished their transmissions.

IV. EXISTING SYSTEM

Using the flooded information, each slave node uses least-squares regression in order to establish a linear relationship between its hardware clock and the clock of the reference node. Using this relationship, each sensor node can predict future clock values of the reference node without communicating frequently. The nodes broadcast their predicted clock values of the reference node to their neighbours for the synchronization of the whole network. However, the predicted time information is not broadcasted quickly upon receiving a new synchronization message. Instead, each node waits until the expiration of its broadcast period. Due to the slow propagation speed of the flood arising from the waiting times, the estimation error of least-squares is amplified at each hop.

V. PROPOSED SYSTEM

The accurate and efficient operation of many applications and protocols in wireless sensor networks require synchronized notion of time. In order to achieve network-wide time synchronization, a common strategy is to flood current time information of a reference node.
into the network, which is utilized by the de facto time synchronization protocol Flooding Time Synchronization Protocol (FTSP). In FTSP, the propagation speed of the flood is slow since each node waits for a given period of time in order to propagate its time information about the reference node. It has been shown that slow-flooding decreases the synchronization accuracy and scalability of FTSP drastically. Alternatively, rapid-flooding approach is proposed in the literature, which allows nodes to propagate time information as quickly as possible. However, rapid flooding is difficult and has several drawbacks in wireless sensor networks. In this paper, our aim is to reduce the undesired effect of slow-flooding on the synchronization accuracy without changing the propagation speed of the flood.

A. SLOW FLOODING TIME INFORMATION IN WSN
In this section, we consider the execution of FTSP in order to analyze the effect of waiting times on the synchronization accuracy. The pseudo-code of FTSP with a fixed reference node is given in Algorithm 1. It should be noted that extra controls are omitted and only the general strategy is presented in the pseudo-code. It is also assumed that each node knows the identifier of the reference node.

B. SLOW FLOODING WITH CLOCK SPEED AGREEMENT
In this section, we describe Flooding with Clock Speed Agreement (FCSA) approach, where all nodes agree on a common logical clock speed by employing a clock speed agreement algorithm and synchronize to a reference node which floods stable time for the whole network. As a consequence, the amplification of estimation errors due to the waiting times is minimized and a synchronization error which grows with the square root of the network diameter is obtained.

C. NEIGHBOUR DETECTION AND REMOVAL
FCSA can be implemented by employing a simple neighbor detection and removal strategy. Whenever a sensor node receives a synchronization message from a new neighbor, it assigns a free slot for that neighbor and starts collecting its information. If any node does not receive a synchronization message from one of its neighbors for a predefined amount of time, it empties the slot which is assigned for that neighbor in the neighbor repository. As a final point, when a new node joins the network, it does not participate in the clock speed agreement immediately. Instead, it first listens a few synchronization packets from its neighbours to achieve initial synchronization. It is infeasible for sensor nodes to keep track of all their neighbors when the number of their neighbors is greater than the capacity of their neighbor repository which is specified in advance. The decision of which neighbors to keep track and which ones to discard is a crucial problem. The communication graph, constructed by considering the neighborhood relations in the neighbor repositories of the sensor nodes, needs to be connected for all times to achieve clock speed agreement. One simple solution that prevents the occurrence of this problem is to specify the capacity of the repository as the maximum node degree in the network. However, due to the memory constraints, this solution may not work for networks with high neighborhood density.

D. DRAWBACKS OF FLOODING
Reliable rapid flooding in sensor networks is difficult due to packet losses. Retransmission, as a simple solution to recover lost packets, may lead to broadcast storm problem. More sophisticated solutions for loss recovery may decrease the speed of the flood and increase energy consumption of the nodes. Severe, drawbacks of rapid flooding mentioned above bring us to the following question: Is it also possible to achieve scalable and tight synchronization in WSNs with slow flooding. In this paper, showing that the undesired effect of slow-flooding on the synchronization accuracy.
VI. CONCLUSION
Achieve scalable and tight synchronization in WSNs with slow-flooding. We emphasized that the performance of drift estimation mechanism for flooding-based time-synchronization protocols. Hence, we revealed that the smaller the error of the drift estimation and, hence, the difference between the speeds of the clocks, the smaller the undesired effect of waiting times on the synchronization accuracy. As a main contribution of this paper, we showed that the synchronization quality of slow-flooding-based time synchronization can drastically be improved by employing a clock speed agreement algorithm among the sensor nodes. We introduced FCSA protocol that forces all nodes to run at the same speed by employing an agreement algorithm and synchronizes them to a reference node that floods stable time for the whole network.

REFERENCES