A Distributed Network Synchronization Algorithm for Wireless Ad Hoc Networks

Carlos H. Rentel and Thomas Kunz

Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada. Emails: {crentel, tkunz@sce.carleton.ca}

Carleton University
Canada's Capital University

MobiCom 2004

Introduction and Motivation

We are concerned with the problem of achieving the synchronization of a set of geographically separated clocks located in every node of a wireless Ad Hoc network without using external information (see figs. 1 and 2). **The discovery of more efficient Network Synchronization (NS) algorithms represents an attractive research topic:**

- **NS has been used to enable and enhance the performance of Medium Access Control (MAC), security, and management protocols**. A substantial amount of research has been devoted to the study of more deterministic MAC protocols that can guarantee a minimum of performance to real-time applications (e.g., voice, video) in wireless Ad Hoc networks. The majority of these studies make use of a slotted time structure that can only be achieved through NS. It is important to offer alternatives to the traditional reliance on the Global Positioning System(GPS) to achieve a slotted time. It is also important to offer a solution tailored to wireless Ad Hoc networks and their QoS support requirements.
- In the context of Wireless LANs, the Timing Synchronization Function (TSF) of the IEEE 802.11 standard is an NS algorithm that makes possible the power management function, and the frequency hoping spread spectrum PHY. layer [1]. **Recent work has identified scalability problems of the TSF in the IBSS ("Ad Hoc") mode that need to be addressed in an efficient manner if the IEEE 802.11 standard is to be used as a platform for future Wireless Ad Hoc Networks [2]**.
- **NS is one of the key functions that enable the co-ordination of packet transmissions in the IEEE 802.16 standard**; particularly challenging and interesting is the problem of achieving efficient NS in its mesh mode of operation.
- **We argue that NS can play a fundamental role in future wireless Ad Hoc networks, therefore, there is a need for algorithms that are more accurate, resilient, autonomous, immune to environmental changes or location, and simple to implement.**

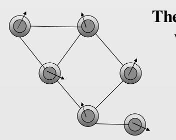

Fig.1. Clocks of the nodes of an Asynchronous Wireless Ad Hoc network

The nodes in the network of fig.1 have clocks that drift with respect to one another, as a result the nodes do not have a common time process that could be utilized, for instance, to coordinate their own packet transmissions. The proposed NS algorithm can transform this network into the one of fig.2, using IEEE 802.11 compatible radios and over-the-air procedures.

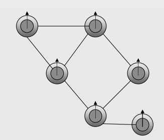

Fig.2. Clocks of the nodes in a Synchronous Wireless Ad Hoc network

Objective - The design of a network synchronization algorithm that is:

- <u>Non-hierarchical and autonomous</u>: **This implies that all the nodes have equivalent influence over the synchronization of the network. No efforts are devoted to the discovery of a node with particular characteristics (e.g., centrally located, maximum degree, fastest clock), and no information external to the network is used. Every node implements the algorithm without the assistance of a central controller. All these translate into an algorithm that is more robust to network dynamics, such as: mobility, node failures, and nodes joining or leaving the network.**
- <u>Convergent</u>:**The network synchronization algorithm should learn what are the timing differences among the clocks in the network and adjusts the time process of each clock automatically. This has the benefit of requiring less over-the-air updates to correct the drift of the clocks, and therefore implies less overhead and energy expenditure.**
- <u>Simple</u>: **This implies the use of the IEEE 802.11 framework without PHY. layer modifications (i.e., no special circuitry required to, for instance, generate pulses that can potentially occupy large bandwidth resources). <u>We will compare the performance of our approach to that of an standardized NS approach: The IEEE 802.11 TSF</u>.**

Some faulty alternatives

We depart from a solution that could be considered reasonable in order to improve the accuracy and scalability performance of the IEEE 802.11 TSF: The extended TSF. **<u>In the extended TSF a node is allowed to transmit its beacon even after successfully receiving a beacon from another node</u>**. This approach is not the most ideal, since we will incur in an increase of overall network energy consumption and overhead. However, the idea is to increase the chances of the nodes to send their beacons. We focus on the probability of successfully transmitting a beacon by a *given* node (P_{given}), and the probability of successfully transmitting a beacon by *any* node (P_{any}). Figure 3 shows P_{given} of the TSF and extended TSF. Although in a lesser degree, the extended TSF suffers from the same scalability problems of the original TSF. One could try to **<u>improve the extended TSF and allow extra beacon transmissions only from those nodes that had a larger timestamp than the timestamp received</u>**. Figure 4 shows the simulation result of this latter approach. The clocks in fig.4 drift linearly with a drift that is drawn from a uniform distribution in the range of ±25ppm, and beacons transmitted at the same time are assumed destroyed (i.e., no capture). **<u>The extended TSF might be of used for networks without energy constrain. However, it falls short in providing a scalable solution</u>.**

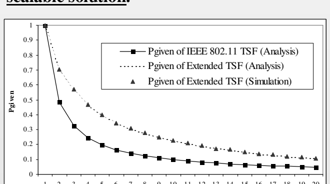

Fig. 3. P_{given} for the TSF and extended TSF

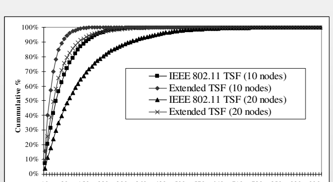

Fig. 4. Maximum time difference for the TSF and extended TSF

Proposed solutions and results

An NS algorithm that truly improves over the TSF should posses a non-hierarchical structure, this will improve the chances of spreading the timing information in highly dynamic Ad Hoc networks.P_{any} gives an indication of the potential improvement of a non-hierarchical (a.k.a democratic) NS algorithm over the more centralized TSF. **<u>Figure 5 suggests that a non-hierarchical NS algorithm based on *all* the beacons transmitted can substantially improve over the original and extended TSF</u>**.

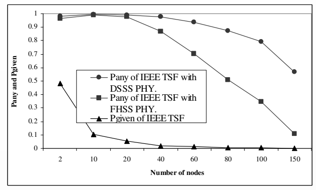

Fig.5 Comparison of P_{given} and P_{any}

<u>Our basic algorithm is as follows:</u>

The fundamental goal is to equalize the drifts of the time processes in every clock. This is achieved by multiplying every time process (read from the hardware clocks) by a correction factor $s_i(t)$; The correction factor can be computed in every node based on the difference between the time-stamp of the received beacon (**<u>coming from *any* node</u>**) and the time-stamp of the local node. That is, node *i* contends to send its time process $T_i(t), \forall i \in \{1,2,...,N\}$ in periodic beacon transmissions in the same way as the IEEE 802.11 TSF. $s_i(t)$ takes the following discrete form

$$s_i(nT) = s_i(nT-1) + K_p \frac{T_{rx\_timestamp}(nT-1) - T_i(nT-1)}{T_i(nT-1)} \quad (1)$$

Where T is the sampling period (i.e., *aBeaconPeriod* in IEEE 802.11). K_p is the proportional design gain of the algorithm,$T_{rx\_timestamp}$ is the time-stamp of the node that successfully transmitted the beacon. The proportional gain , if chosen appropriately, can average the time processes of all the clocks in the network to achieve synchronization. Figure 6 and 7show plots of the maximum time difference among the clocks using equation (1) along with a linear model of the clocks' time processes and the expression for P_{any} found in [2].

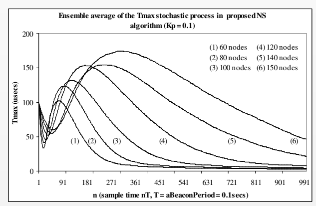

Fig.6. Maximum time difference among the clocks in a network of different sizes

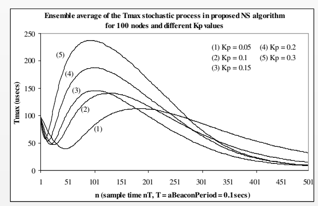

Fig.7. Maximum time difference among the clocks in a network of 100 nodes and different proportional gains

<u>Enhancements to the basic algorithm:</u>

The Rotating Master Node (RMN) algorithm. All the nodes have a counter C_i with a maximum value $C_i^{\max}$. In every TBTT, all the nodes perform the following algorithm independently: (Figs. 8. 9, 10, and 11 show several simulation results)

1. $C_i = \max\{C_i - 1, 0\}$
2. If $C_i = 0$, contend to send the beacon following the same procedure of the IEEE 802.11 TSF, otherwise (i.e., $C_i > 0$), wait until next TBTT and return to step 1 without attempting to transmit a beacon.
3. If a beacon is successfully received before the local node sends its beacon (assuming $C_i = 0$), then set $C_i = C_i^{\max}$, adjust the correction factor based on the time-stamp received (equation 1), wait for the next TBTT and return to step 1
4. If a beacon is successfully received and $C_i > 0$, then adjust the correction factor (equation 1), wait for the next TBTT and return to step 1.

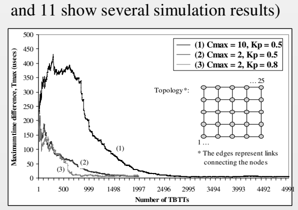

Fig.8. Maximum time deviation in a 5x5 grid

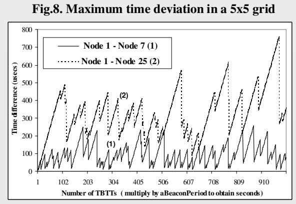

Fig.9. Time deviation in a 5x5 grid using the TSF

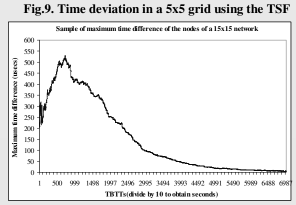

Fig.10. Maximum time deviation in a 15x15 grid

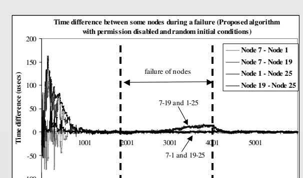

Fig.11. Maximum time deviation with simulated failures

Conclusion

The proposed algorithm is non-hierarchical (democratic), convergent, distributed, IEEE 802.11 compatible (i.e., no special circuitry, or over-the-air changes to the standard are required), and autonomous. Future work includes the implementation of this algorithm in a test-bed and a more in-depth study of its stability properties.

References:

[1] IEEE Std. 802.11. "Wireless LAN medium access control (MAC) and physical layer specification," 1999.
[2] L. Huang, T-H Lai, "On the scalability of IEEE 802.11 Ad Hoc networks," Mobihoc, June 2002, pp. 173-182