# Demo Abstract: RATS – Rapid Time Synchronization

Janos Sallai, Branislav Kusy, Akos Ledeczi
Institute for Software Integrated Systems, Vanderbilt University, 2015 Terrace Place, Nashville, TN 37203, USA
Email: {janos.sallai, branislav.kusy, akos.ledeczi}@vanderbilt.edu

## Keywords

Wireless Sensor Networks, Routing, Time Synchronization, RATS

A large class of sensor network applications requires that sensors have knowledge of a common reference time (virtual global time). Typical use cases of a time synchronization service are coordinated action (e.g. a group of sensors coordinate to acquire sensor data at the same instant) or data fusion (i.e. timestamped sensor readings from multiple nodes are combined into high level information). RATS is a new time synchronization approach that allows for fast dissemination of global time information in a wireless sensor network.

RATS uses periodic flooding to disseminate global time information in the network. The time base of a designated root node is used as a reference, that is, RATS synchronizes the rest of the nodes with the root's clock. The root initiates synchronization periodically by broadcasting synchronization messages, which are relayed from node to node in the network.

The synchronization messages contain the following information:

- sequence number,
- timestamp of transmission of initiator message in the root's time base,
- timestamp of transmission of initiator message in the sender's time base,
- timestamp of transmission of the actual message in sender's time base (set by the timestamping service in the MAC layer on transmission, as described in [2]),
- timestamp of transmission of the actual message in the receiver's time base (set by the timestamping service in the MAC layer on receive).

Because of the nature of flooding, a node may receive multiple messages with the same sequence number through different paths. We only consider the first message with a given sequence number to keep the convergence fast. On receipt of a synchronization message, the clock offset of sender and receiver is calculated. Using this offset, the receiver node converts the timestamp of initiator message from the sender's time base to its own time base. The difference of the converted value and the timestamp of the initiator message in the root's time base will give the clock offset of the receiver node and the root. The node updates the message contents with the values it calculated and rebroadcasts the message, so that it can propagate further in the network.

The node remembers the receive timestamps of the last four messages, as well as the corresponding offsets between the root's clock and its own. From this historical data, it calculates the skew between the root's clock and the node's clock using linear regression. Using both the clock skew and the last measured offset, the nodes stay synchronized for a longer period of time than in the case of simple offset-based techniques. Since we compensate for the clock skews, there is no need for frequent synchronization updates, which results in a decrease of communication overhead.

The experiment demonstrates how coordinated action is achieved through time synchronization of RATS. In particular we demonstrate the fast convergence of the algorithm in a simulated multihop network and the low radio communication overhead which is possible by clock skew compensation.

N sensor nodes are arranged in a circle mimicking the slots of a roulette wheel. The presence of the ball in a slot is marked by the LED's of the corresponding sensor node. Clocks are synchronized using RATS. A dedicated coordinator node broadcasts a radio message that contains the global time when the spinning of the ball begins, as well as the time when it ends. On receipt of the message, slot nodes calculate the times when the ball passes over the corresponding slots, and execute the calculated schedule by maintaining the state of their LED's accordingly in a timely manner.

## References

[1] B. Kusy and P. Dutta and P. Levis and M. Maroti and A. Ledeczi and D. Culler, Elapsed Time on Arrival: A simple and versatile primitive for canonical time synchronization services, International Journal of Ad Hoc and Ubiquitous Computing, vol. 2, no. 1, 2006.

[2] Maroti M., Kusy B., Simon G., Ledeczi A.: The Flooding Time Synchronization Protocol, ACM Second International Conference on Embedded Networked