# Fault-Tolerant Flooding Time Synchronization Protocol for Wireless Sensor Networks

Laura Gheorghe, Răzvan Rughiniş, Nicolae Țăpuş
Politehnica University of Bucharest, Romania
laura.gheorghe@cs.pub.ro, razvan.rughinis@cs.pub.ro, ntapus@cs.pub.ro

# Wireless Sensor Networks

- Sensor nodes:
  - Low power and low cost
  - Collect and relay information about the environment
- Major applications:
  - Environmental monitoring
  - Health care
  - Mood-based services
  - Positioning and animal tracking
  - Industrial and military applications

# Time Synchronization

- Also known as clock synchronization
- Required for consistent distributed sensing and control
- Applications that require high precision are:
  - Low-power TDMA schedule
  - Acoustic data processing
  - Suppressing redundant information
  - Calculating velocity

# Fault tolerance

- Cause of failures of sensor nodes:
  - Energy depletion
  - Hardware failure
  - Communication link error
  - Malicious attacks
- Non reliable components:
  - Sensing unit
  - Wireless transceiver
- Communication also affected
- Fault tolerance = The ability to operate in the presence of faults

# Related work

- Network time protocol (NTP)
  - NTP servers synchronize from external sources
  - NTP clients synchronize from servers
  - Not suited for WSNs
    - MAC layer of the radio stack
    - High computation overhead
    - Message transfer overhead
- Reference Broadcast Synchronization (RBS)
  - First protocol to achieve performance in WSNs
  - The reference node sends a broadcast packet
  - Nodes exchange messages to compare timestamps
  - To eliminate non-deterministic delay

# Related work (cont.)

- Timing-sync Protocol for Sensor Networks (TPSN)
  - Overcome the limitations of RBS
  - Builds a spanning tree of the network
  - From the root, each parent synchronizes its children
  - MAC layer time-stamping
  - Precision two times higher than the RBS
- Lightweight Tree-Based Synchronization (LTS)
  - Minimize complexity instead of maximizing accuracy
  - Two algorithms: centralized and distributed
  - require the nodes to synchronize themselves to a reference node

# Flooding Time Synchronization Protocol

- Single broadcast time-stamped message
  - To synchronize the time of the sender to the one of the receivers
- MAC layer time-stamps
  - To eliminate errors and achieve precision
- Linear regression
  - To compensate clock drift
- Multi-hop networks:
  - Elects a root that maintains global time
  - Root synchronizes all other clocks
  - Uses intermediate nodes to transport global time
  - Builds an initial tree
  - Allows topology changes

# Fault-tolerant Flooding Time Synchronization Protocol

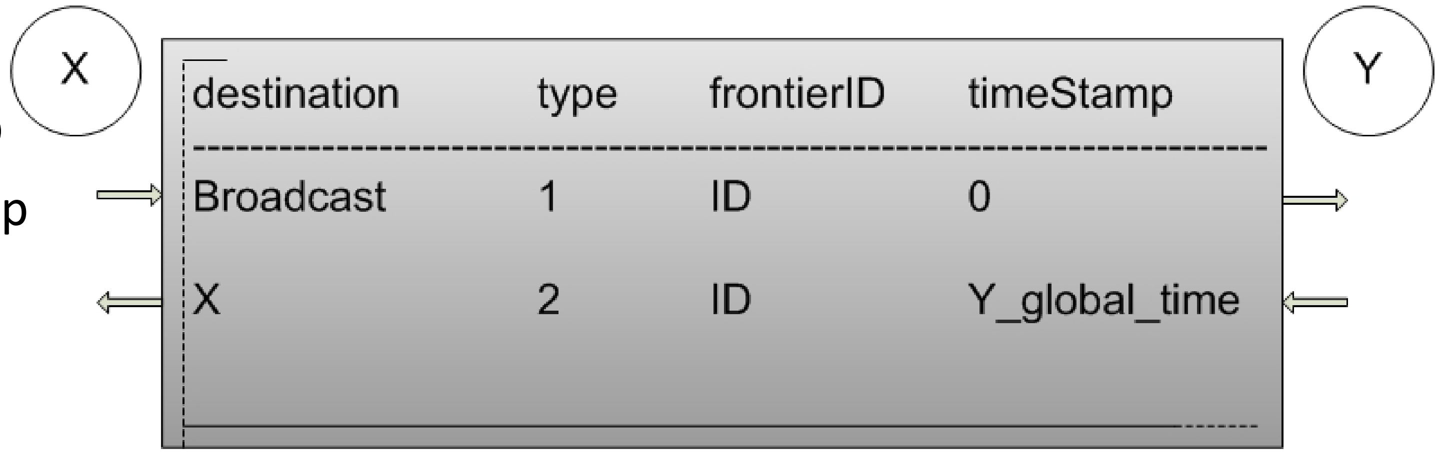
- Allows correct synchronization in the presence of failures
- The algorithm includes three steps:
  - Fault detection
  - Asking for help and receiving help
  - Decision
- Fault detection:
  - The node becomes aware of the inconsistent clock value
  - TIME_ERROR_LIMIT
    - Maximum difference between sequential received global times
  - behavior of FTSP in this case:
    - Erase the regression table and accept the new global time
  - FFTSP behavior:
    - Store local time
    - Ask nodes from the same frontier about their received global times

# Fault-tolerant Flooding Time Synchronization Protocol

- Asking for and receiving help
  - The node requests the latest global time from neighbors
    - In the same broadcast range
    - In the same frontier
  - The node stores each value received from neighbors
  - The node waits for replies for a period of time
- Deciding
  - Uses stored global times received from neighbors
  - Computes a global time using median method
  - Result incompatible with initially global time
    - Sender malicious, initial value ignored
  - Result compatible with initially global time
    - Initial value and stored local time form a reference point

# Implementation

- FTSP already included in TinyOS
- The code was modified to incorporate fault-tolerance
- Testing was done using TOSSIM
- Messages include fields:
  - destination
  - type
  - frontierID
  - timeStamp



# Discussion

- Several scenarios were tested
- The protocol corrected time stamps
  - In the presence of less than [(n+1)/2] incorrect replies
- Problems:
  - Node receives [(n+1)/2] or more incorrect time values
    - Node is forced to accept a false time stamp
    - Same situation if neighbors lie
  - False frontierID
    - To prevent the communication between neighbors
    - Neighbors will ignore request messages
    - The period will expire
    - Node is forced to accept the incorrect global time

# Conclusion

- FFTSP extends FTSP in order to provide fault-tolerance
- Detects faults - malicious nodes or transmission errors
- Ability to detect and correct time stamps
- Can perform correction
  - Presence of less than [(n+1)/2] incorrect time stamps
- Energy consuming
  - Extra messages exchanged
  - Trade-off between energy consumption and fault-tolerance
- Future work:
  - Extensive testing
  - Energy consumption and scalability