A Tale of Two Synchronizing Clocks

Jinkyu Koo, Rajesh K. Panta*, Saurabh Bagchi*,*
*and Luis Montestruque***

* Dependable Computing Systems Lab (DCSL)
School of Electrical and Computer Engineering
Purdue University

** EmNet, LLC

Problem Statement

- A wide-area Wireless Sensor Actuator Network (WSAN) called CSOnet is in operation in South Bend, IN for detecting and controlling wastewater flow to the treatment plant
  - CSOnet has a low duty cycle: awake 6 seconds in a 5 minute period (2% duty cycle)
  - CSOnet nodes called Chasqui nodes are deployed in the underground wastewater channels for sensing, on top of traffic poles for relays, and at traffic intersections as gateways to the cellular network
- CSOnet is meant for long-term operation – of the order of several months – and hence needs accurate time synchronization

Distinctive Challenges for Synchronization in CSOnet

1. The synchronization has to be fast
   - Ideally entire network should be synchronized within the awake period of 6 seconds
   - The projected scale of the network is large, of the order of a few hundred nodes
2. Chasqui nodes use a Real Time Clock (RTC), external to the microcontroller, for controlling wakeup
   - RTC has a low drift of 2 ppm over the large temperature range -13°F to 122°F and low power consumption
   - But RTC trades off low granularity (1 sec) for achieving the above two properties
   - Problem is wake up is controlled by a clock whose granularity is too coarse for the low duty cycle
3. A high power, long range, and high speed radio is used in Chasqui for which the firmware is not available for modification
   - A common technique for time synchronization, MAC layer time-stamping, is unavailable
   - MAC layer time-stamping has documented problems at high speeds

Our Contributions

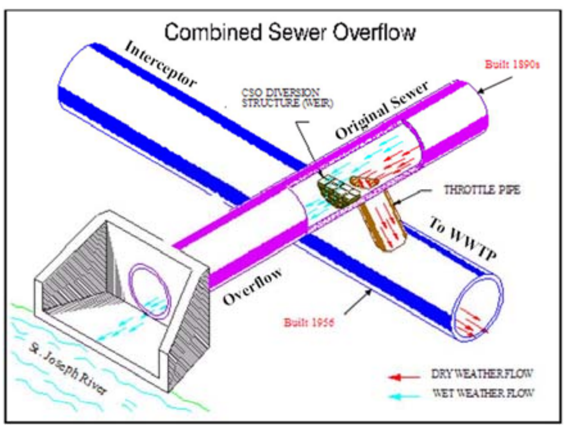
- We develop a synchronization protocol called HARMONIA that solves the above three challenges
- There are three primary design innovations
  1. It uses the high resolution microcontroller clock to synchronize the low resolution RTC
  2. It pipelines the two phases of the synchronization-related messages to achieve fast synchronization
  3. It handles transient node and link failures in a localized manner
- HARMONIA achieves the following properties
  - Synchronization error of the order of tens of microseconds
  - Fastest available synchronization protocol to date

PURDUE UNIVERSITY

Presentation Outline

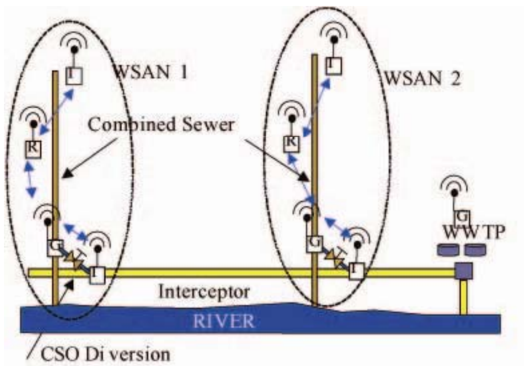
- Introduction
  - CSOnet
  - Chasqui mote
  - Synchronization issue with sleep/wake operation
- Difficulties with FTSP
- Proposed protocol: HARMONIA
- Experimental results
- Concluding remarks

CSO Problem



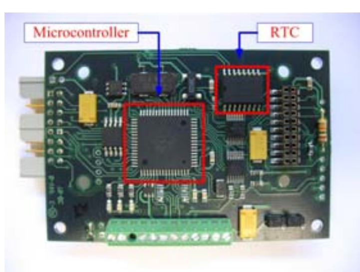
• CSOnet: Metropolitan scale, wireless sensor actuator networks, configured by about 150 nodes (diameter less than 20 hops)
• Monitors combined sewer overflow (CSO) events in 111 locations in South Bend, IN

CSOnet Solution

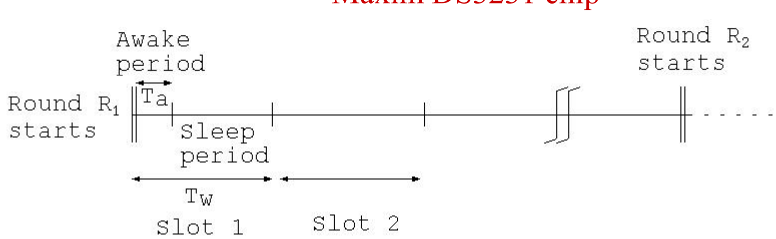


- Sensor nodes sense impending overflow
- Relay this to a centralized controller
- Controller activates the smart valves to temporarily hold the wastewater in excess channels that exist through the system

Chasqui Mote



- Based on the Crossbow Mica2 design
- Microcontroller: ATmel Atmega128L
- Longer range radio: MaxStream 9XTEND 900MHz
  - Outdoor RF line-of-sight range up to 40 miles w/high gain antenna
- Precision external real time clock (RTC) for sleep/wake operation: Maxim DS3231 chip



Domain Specific Challenges for Synchronization: Further Discussion

- Our goal is to synchronize RTCs of all nodes accurately enough to ensure all nodes in the network wake up at the same time – nodes stay awake for 6 seconds in a 5 minute window.
- Practically nodes can communicate with each other for even less than 6 seconds due to the drift in RTC when a synchronization protocol is initiated. Therefore, we target synchronizing the whole network within 2 seconds.
  - Fast synchronization is required
- External RTC is of a low drift (2 ppm), but has a coarse granularity of only 1 second.
  - Synchronizing the RTC alone may lead to a synchronization error of up to 1 sec even with a perfect protocol, which is unsuitable for CSOnet
- MaxStream 9XTEND radio does not allow its firmware modification.
  - However, it provides signals to the application layer when RF starts to transmit/receive a packet

Can We Apply FTSP for Synchronizing CSOnet?

- Flooding Time Synchronization Protocol (FTSP)[1]
  - Uses Mac layer time-stamping
  - A elected root periodically broadcasts a synchronization message
  - Each node periodically broadcasts to its neighbors its local estimate of the time at the root node
  - Each node uses a linear regression to estimate the offset and skew between the local clock and that of the root
- FTSP achieves highly accurate synchronization, but it takes a relatively long time to synchronize the whole network
  - Roughly PN_RN time required to finish synchronization
    - P: flooding period, N_R: the number of points needed to draw the regression line, N: the maximum number of hops in the network from the root
  - e.g., N=20hops within 6 seconds with P=300ms & N_R=8
  - Practically a much smaller network can be synchronized due to several factors
  - With such frequent flooding, the network is overwhelmed by the synchronization messages, barely leaving a chance for data communication

[1] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," Sensys 2004.

Two Strawman Proposals to Adapt FTSP

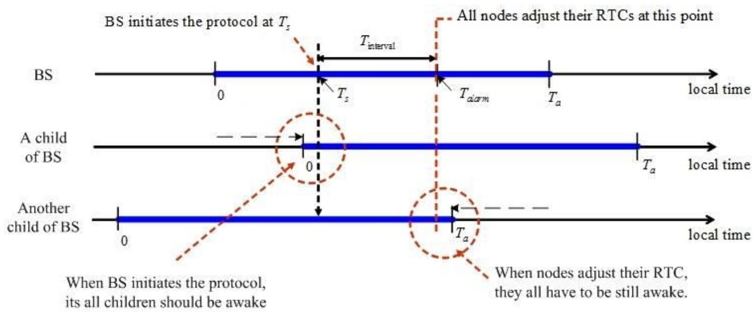
- First: Synchronize the microcontroller clock (MCC)
  - Operates in the frequency of 8MHz (for Mica2 node) → 0.125µs granularity
  - FTSP can synchronize the MCC accurately – to within a few microseconds
  - However, the MCC does not run in sleep mode, and thus cannot be used to trigger a wake up
- Second: Synchronize the RTC over multiple slots
  - Since RTC does not sleep, we can afford to synchronize it over multiple slots
  - However, RTC has a coarse granularity of 1 sec and therefore the synchronized clocks may differ by up to 1 sec

Overview of Proposed Protocol HARMONIA

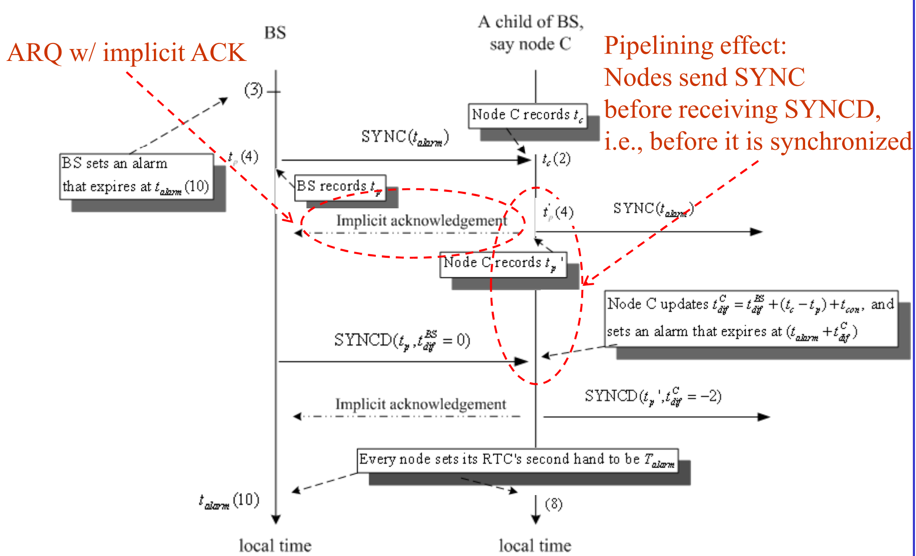
- We have two clocks in the picture
  - MCC: high resolution (0.125 µs), but poor accuracy (40 ppm); not working in sleep mode
  - RTC: low resolution (1 second), but fine accuracy (2 ppm); working even in sleep mode
- We propose HARMONIA that
  - First synchronizes the MCCs, and
  - Then adjusts the RTCs to a globally determined value at the *same time* based on the synchronized MCCs
- Strawman
  - Synchronize the MCCs using FTSP spread across multiple wake up slots
  - This does not work – MCC has to be synchronized within one slot (See paper for reason why)

HARMONIA Timeline

- Tree topology is assumed with BS as its root
- Base station (BS) initiates the synchronization procedure



HARMONIA Synchronization Messages

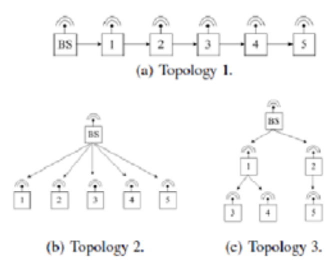


HARMONIA Failure Handling

- A node tries N_{max} times within a slot to synchronize all its children
  - Implicit acks for both SYNC and SYNCD are used to detect failure
- If a node is unable to synchronize all its children in a slot, HARMONIA's fast recovery mechanism kicks in
  - Say node A is in such a situation
  - Node A actively initiates the synchronization procedure in the next slot, targeted only to its descendant sub-tree
  - Each node maintains state such that it remembers where it was in the synchronization handshake with its parent and responds accordingly
- The fast recovery concept can handle the situation that a large network cannot all be synchronized in one slot
  - HARMONIA proceeds with as much of the network being synchronized initially as possible
  - The unsynchronized parts of the network are handled through fast recovery

Experimental Results (1/3)

- Comparison with FTSP in terms of synchronization time and synchronization error
- Three different topologies considered

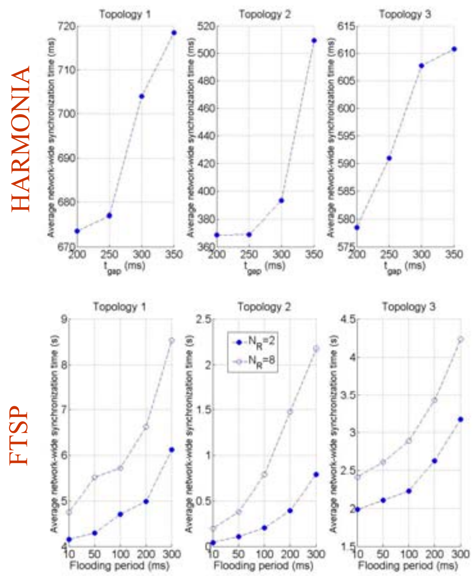


(a) Topology 1. (b) Topology 2. (c) Topology 3.

Values of parameters in HARMONIA used in the experiments.

T_a	T_w	T_s	T_{alarm}	t_{bf}	t_{con}	N_{max}
6s	5min	2s	4s	100ms	250μs	3

Experimental Results: Synchronization Time (2/3)



- t_{gap}: average time between SYNC and SYNCD when there is no retransmission
  - Corresponds to flooding period (P) of FTSP
  - Equivalence between t_{gap} and P is not perfect
- HARMONIA's time is in the order of a few hundreds of msecs
- FTSP's time is in the order of a few seconds
- In Topology 1, with t_{gap}=200ms and equivalently, P=200ms, FTSP is 7.4X and 9.8X slower than HARMONIA, for number of regression points 2 and 8 respectively.
- Regression with 2 points is risky and can lead to poor accuracy

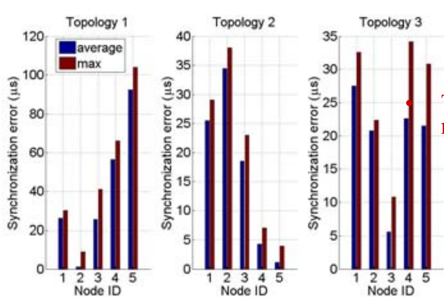
PURDUE UNIVERSITY

Experimental Results: Synchronization Error (3/3)

One-hop synchronization error

	HARMONIA	FTSP ($N_R = 8$)
average	16.77μs	1.5μs
max	38μs	3μs

HARMONIA multi-hop synchronization error



- FTSP outperforms HARMONIA in terms of synchronization error
  - HARMONIA does not compensate for (1) the differential drifts in the MCCs of two nodes; (2) the jitter in interrupt handling that occurs when the MaxStream radio gives a signal on message transmission and reception
- How frequently do we have to synchronize a one-hop network for out-of-synchronism less than 0.1 second
  - HARMONIA: (0.1s-38µs)/2µ = 49981 seconds (13.88 hours)
  - FTSP: (0.1s-3µs)/2µ = 49998.5 seconds
  - Only 17.5-second gap!
- The synchronization error in HARMONIA does not build up with increasing # hops
  - Example: The error decreases from node 1 to node 2 in Topology 1
  - The sign of the synchronization error between a pair of nodes depends on the relative frequencies of the clocks of the two nodes and could be either positive or negative
  - HARMONIA does a (signed) addition of the synchronization errors and therefore canceling effect can happen

PURDUE UNIVERSITY

Concluding Remarks

- HARMONIA is proposed for synchronizing CSOnet where nodes stay awake only for 6 seconds in a 5-minute slot
  - Uses two different clock sources, MCC and RTC, to achieve both fine granularity (0.125 µs) and fine accuracy (2 ppm)
  - Synchronize the whole network fast (674ms for 5-hop network, where FTSP requires more than 6.5 seconds)
  - Fast recovery mechanism is introduced so that localized recovery can occur while synchronizing large networks in multiple-slots
- Our proposed work includes
  - Deploying and measuring the performance of HARMONIA in the 100-node+ deployed network
  - Modifying HARMONIA to apply to Micaz platform with MAC layer time-stamping