CS450
Network Embedded Sensing Systems
Week 11: Time Synchronization

http://hinrg.cs.jhu.edu/CS450/
Andreas Terzis @ CS JHU
Outline

- Description of the problem: axes, shortcomings
- Reference-Broadcast Synchronization
  - Reference-Broadcast Synchronization
  - Federation of broadcast domains
  - Post-facto synchronization
- Alternatives
  - TPSN
  - FTSP
- Post-mortem time reconstruction
Computer Clocks

- **Hardware clock implementation**
  - Oscillator of specified frequency
  - Counter register incremented after certain number of pulses

- **Software clock**
  \[ L(t) = \theta \cdot H(t) + \phi \]

- **Inter-node clock precision**
  - External synchronization: \( |L_i(t) - t| < \delta \)
  - Internal synchronization: \( |L_i(t) - L_j(t)| < \delta \)
Error Sources

- Nodes start at different times
- Clocks run at different rates: drift
  - Due to manufacturing defects, environmental conditions
  - Measured in parts per million (ppm)
    - 100 ppm corresponds to 1 sec every ~2.8 hours
- Oscillator frequency changes over time
- \[ |L_i(t) - L_j(t)| \] can become arbitrary large
- Need for time synchronization protocol
  - Works by adjusting \( \theta, \phi \)
- Resynchronization interval is a function of drift rate and desired accuracy
Does timesync matter?

- **Internet Time Synchronization**
  - Critical in some contexts (e.g. crypto, distributed packet traces)
  - A convenience in many other contexts

- **Sensor Network Synchronization**
  - Fundamental to its purpose: data fusion
  - *Shared reference time* needed to relate events in the *physical world*
Heterogeneity

- Time sync is critical at many layers
  - Beam-forming, localization, distributed DSP
  - Data aggregation & caching
  - TDMA guard bands
  - “Traditional” uses (debugging, user interaction…)

- But time sync needs are non-uniform
  - Maximum Error
  - Lifetime
  - Scope & Availability
  - Efficiency (use of power and time)
  - Cost and form factor
Isn’t this solved?

- NTP (Network Time Protocol)
  - Ubiquitous in the Internet
  - Variants appearing in sensor networks
- 802.11 synchronization
  - Precise clock agreement within a cluster
- GPS, WWVB, other radio time services
- High-stability oscillators (Rubidium, Cesium...)
NTP

- The “gold standard” -- used by millions
- The basic idea: measure round-trip-time

One-way delay: \[ \Delta = \frac{(t_3 - t_0) - (t_2 - t_1)}{2} \]

Offset: \[ t_1 - (t_0 + \Delta) = \frac{t_1 - t_0 + t_2 - t_3}{2} \]
So what’s wrong?

- Existing work is a critical building block

**BUT**

- This isn’t the Internet
  - Important assumptions no longer hold
    - (fewer resources available for synchronization…)
  - Sensor apps have stronger requirements
    - (…but we have to do better than the Internet anyway)

- Energy, energy energy:
  - Listening to the network is no longer free; even occasional CPU use can have a major impact
Infrastructure vs. Ad-Hoc

- “NTP provides UTC to the entire Internet”
- Infrastructure isn’t ubiquitous in sensor nets
  - GPS doesn’t work indoors, in the forest, underwater, on Mars…
- What happens without infrastructure?
“Mundane” Reasons

- **Cost**
  - We can’t put a $500 Rubidium oscillator or a $50 GPS receiver on a $5 sensor node

- **Form factor**
  - Nodes are small, extra components are large

- Not actually a mundane limitation if it changes the economics of the sensor net
Leveraging the Medium

- Strict layering and levels of abstraction prevent us from exploiting domain knowledge
- Wireless networks often use network interfaces with physical-layer broadcasts
- Reference Broadcast Synchronization takes advantage of this to remove most of the non-determinism from the critical path
Traditional sync

Problem: Many sources of unknown, nondeterministic latency between timestamp and its reception
Analysis of Delay in Transmission and Reception

- Interrupt Handling Time
- Encoding Time
- Decoding Time
- Byte Alignment Time

<table>
<thead>
<tr>
<th>Time</th>
<th>Magnitude</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send and Receive</td>
<td>0 – 100 ms</td>
<td>nondeterministic, depends on the processor load</td>
</tr>
<tr>
<td>Access</td>
<td>10 – 500 ms</td>
<td>nondeterministic, depends on the channel contention</td>
</tr>
<tr>
<td>Transmission / Reception</td>
<td>10 – 20 ms</td>
<td>deterministic, depends on message length</td>
</tr>
<tr>
<td>Propagation</td>
<td>&lt; 1μs for distances up to 300 meters</td>
<td>deterministic, depends on the distance between sender and receiver</td>
</tr>
<tr>
<td>Interrupt Handling</td>
<td>&lt; 5μs in most cases, but can be as high as 30μs</td>
<td>nondeterministic, depends on interrupts being disabled</td>
</tr>
<tr>
<td>Encoding plus Decoding</td>
<td>100 – 200μs, &lt; 2μs variance</td>
<td>deterministic, depends on radio chipset and settings</td>
</tr>
<tr>
<td>Byte Alignment</td>
<td>0 – 400μs</td>
<td>deterministic, can be calculated</td>
</tr>
</tbody>
</table>
Reference Broadcasts*

Sync 2 receivers with each other, NOT sender with receiver

RBS reduces error by removing much of it from the critical path.

Traditional critical path: From the time the sender reads its clock, to when the receiver reads its clock.

RBS: Only sensitive to the differences in receive time and propagation delay.
Receiver Determinism

![Graph showing pairwise difference in packet reception time (usec) against number of trials.](image)
Basic Mechanism Description

- Some node sends $m$ broadcast “reference” messages
- Each of $n$ receivers records the time each reference message was received
- Receivers exchange their observations
- Receiver $i$ computes offset to receiver $j$ as the average of phase offsets

$$Offset[i, j] = \frac{1}{m} \sum_{k=1}^{m} (T_{j,k} - T_{i,k})$$
Comparison to NTP

- Second implementation:
  - Compaq IPAQs (small Linux machines)
  - 11Mbit 802.11 PCMCIA cards
- Ran NTP, RBS-Userspace, RBS-Kernel
  - NTP synced to GPS clock every 16 secs
- In each case, asked 2 IPAQs to raise a GPIO line high at the same time; differences measured with logic analyzer
NTP Comparison: Low Network Load

Cumulative Error Probability

Synchronization Error with Light Network Load

- RBS-Kernel
- RBS-Userspace
- NTP-Userspace

Error (usec)
Clock Resolution

- RBS degraded slightly (6 µs to 8µs)
- NTP degraded severely (51 µs to 1542 µs)

NTP Comparison: High Network Load
Multi-Hop RBS

- Some nodes broadcast RF synchronization pulses
- Receivers in a neighborhood are synced by using the pulse as a time reference. (The pulse senders are not synced.)
- Nodes that hear both can relate the time bases to each other
Time Routing

The physical topology can be easily converted to a logical topology; links represent possible clock conversions.

Use shortest path search to find a “time route”; Edges can be weighted by error estimates.
Multi-Hop RBS

Error (and std dev) over multiple hops, in usec

1 Hop: 1.85 +/- 1.28
2 Hop: 2.73 +/- 1.91
3 Hop: 2.73 +/- 2.42
4 Hop: 3.68 +/- 2.57

Error (usec)
Timing-sync Protocol for Sensor Networks (TPSN)

- Similar to NTP in many ways -- uses round-trip time measurement with 2 packets
- Achieves a network-wide synchronization by constructing a tree and synchronizing each node with its parent
- Depends on being able to modify the MAC, to do time-stamping very close to transmission
- Demonstrates 2x better performance than RBS based on analysis and experimentation
RBS vs TPSN on Accuracy

- Uncertainties in Radio Message Delivery
  - Send Time
  - Access Time
  - Transmission Time
  - Propagation Time
  - Reception Time
  - Receive Time

- RBS (receiver-receiver synchronization)
  - Eliminate impacts of the send and access time
  - Can remove the receive time with minimal OS modification
  - Source of errors: propagation and reception time
  - Does not require access to the low-levels of the OS

- TPSN (sender-receiver synchronization)
  - Remove the send, access, and receive time by MAC-layer time-stamping
  - Eliminate the propagation time via two-way handshakes
  - Require construction of a tree (level discovery phase)
The Flooding Time Synchronization Protocol

M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi

SenSys 2004
Summary

- Achieve a network-wide synchronization through one-way radio broadcast
  - Does not compensate for propagation errors as in TPSN
- MAC Layer Time-stamping
- Clock drift management
- Multi-hop time synchronization
Time Stamping

- Using periodic radio broadcast to synchronize receivers to the sender
- Time stamp of the sender is embedded in the transmission message
- Each broadcast provides a reference point (a global-local time pair) to each receiver for estimating the clock offset between the sender and the receiver
- The proposed time stamping mechanism reduces the jitter of interrupt handling and encoding/decoding times
  - Achieved 1.4μs average error and 4.2μs maximum error in experiments
Clock Drift Management

- The offset between two local clocks can change in a linear fashion due to clock drifts.
- Linear regression can be used to estimate the skew from multiple reference points as done in RBS.
Multi-hop Time Synchronization

- Basic scheme:
  - A single root is required for global synchronization
  - Each node synchronizes itself based on multiple received reference points
  - Once a node is synchronized, it broadcasts synchronization messages to its neighbors

- Synchronization Message Format
  - timeStamp
  - rootID
  - seqNum: set and increment by the root; each node inserts the most recent received seqNum to its broadcast messages

- Managing Redundant Information: message filtering
- The root election problem: through broadcast messages without additional handshakes
Experiment Results

- ID 1
  - Reset random nodes
  - Turn off odd-ID nodes
  - Turn on odd-ID nodes

---

Graph showing time (hh:mm) vs. microseconds with different lines representing average pairwise error, maximum pairwise error, and synchronized nodes percentage.