A Distributed Sleep Scheduling Protocol under Loosely Synchronized Clocks for Wireless Networks

David (Bong Jun) Choi (Supervisor: Xuemin (Sherman) Shen)
Broadband Communications Research Lab.
Department of Electrical and Computer Engineering
University of Waterloo
Scenario 1:

- **Conference**
  - **Conditions**
    - Each hour you choose to be either at (1) the conference hotel or (2) your room
  - **Objective**
    - Two persons (you and Professor) want to meet each other
    - With the least number of visits to the conference hotel
  - **Knowledge**
    - They agreed to meet at 3:00 PM

- What would you do?
  - Attend the conference at 3:00 PM

- “Scheduled” (Synchronous)
Scenario 2:

- **Conference**
  - **Conditions**
    - Each hour you choose to be either at (1) the conference hotel or (2) your room
  - **Objective**
    - Two persons (you and Professor) want to meet each other
    - With the least number of visits to the conference hotel
  - **Knowledge**
    - Each person knows that the other person will be at the conference today
    - But, do not know when?

- **What would you do?**
  - Visit the conference hotel several times during the day?
  - How often? When?

- **“Unscheduled” (Asynchronous)**
Scenario 3:

- **Conference**
  - **Conditions**
    - Each hour you choose to be either at (1) the conference hotel or (2) your room
  - **Objective**
    - Two persons (you and Professor) want to meet each other
    - With the least number of visits to the conference hotel
  - **Knowledge**
    - Each person knows that the other person will likely be at the conference today around the lunch buffet

- What would you do?
  - Probabilistic?

- “Somewhat/Loosely scheduled” (Semi-Asynchronous)
Research Outline

DSA: Distributed Semi-Asynchronous Sleep Scheduling for Mobile Wireless Networks

- **What?**
  - Design a sleep scheduling protocol for wireless mobile devices in a distributed network topology considering loosely synchronized clocks

- **Why?**
  - Reduce energy consumption
  - Practical to assume some level of synchronization accuracy

- **How?**
  - Using slotted frame structure with predetermined allocation of active slots
  - Allocation depending on the synchronization error estimation
Overview

- **Introduction**
  - Synchronization and Energy Saving
  - Semi-Asynchronous Sleep Scheduling

- **System Model**

- **Proposed Protocol**

- **Performance Evaluation**

- **Conclusion**
Introduction:

- **Synchronization**
  - “Timekeeping which requires the **coordination of events** to operate a system in unison”

- **Applications**
  1) **WSN**: Accurate timing information for data collection
  2) **TDMA**: Slot alignment
  3) **PSM**: Sleep Scheduling
Introduction:

- Timekeeping in Wireless Communications

- NTP (Network Time Protocol)

- GPS (Global Positioning System)

a) Quartz crystal clocks  
b) Chip scale atomic clock by NIST
Introduction:

- In reality,
  - Unavailability of GPS / Intermittently available energy, cost, isolation
  - Imperfect clocks
    - Inherent oscillator error
    - External factors: temperature, pressure, noise, aging, etc.
  - Error increase with time and number of hops [TTP]

<table>
<thead>
<tr>
<th>Device</th>
<th>Accuracy</th>
<th>Power</th>
<th>Lifetime with AA battery</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>$10^{-8} - 10^{-11}$</td>
<td>180 mW</td>
<td>16.7 hrs</td>
<td>beacon, outdoor, $$$ cost</td>
</tr>
<tr>
<td>DARPA chip-scale atomic clock</td>
<td>$10^{-11}$</td>
<td>$&lt; 30$ mW</td>
<td>100 hrs</td>
<td>program goals</td>
</tr>
<tr>
<td>MCXO</td>
<td>$3 \times 10^{-8}$</td>
<td>75 mW</td>
<td>40 hrs</td>
<td>large, aging drift</td>
</tr>
<tr>
<td>TCXO</td>
<td>$6 \times 10^{-6}$</td>
<td>6 mW</td>
<td>500 hrs (21 days)</td>
<td>$&gt; 1$ PPM</td>
</tr>
<tr>
<td>Watch clock</td>
<td>$200 \times 10^{-6}$</td>
<td>1 $\mu$W</td>
<td>342 yrs</td>
<td>temperature &amp; aging drift</td>
</tr>
</tbody>
</table>

Synchronization Error & Energy Efficiency

Sleep scheduling requires accurate timing for coordinating awake/active periods.

- Larger Synchronization Error
- Longer Awake Periods (Guard Periods)
- Higher Energy Consumption
System Model

- Clock Model: $C(t)$

$$C_i(t) = (1 + k_i) \int_{t_0}^{t} \omega(\tau) d\tau + C_i(t_0), \ i = 1, 2, \ldots, n$$

$$1 - \rho \leq \frac{dC_i(t)}{dt} \leq 1 + \rho, \ \forall \ t$$

$$\rho \in [10, 100] \text{ ppm}$$

0.6 ~ 6.0 ms error in 60.0 s
Synchronous vs. Asynchronous vs. Semi-Asynchronous

Scenario 1:

a) Synchronous:

<table>
<thead>
<tr>
<th>t</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>node i</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>node j</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Scenario 2:

b) Asynchronous:

<table>
<thead>
<tr>
<th>t</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>node i</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>node j</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
Synchronous vs. Asynchronous vs. Semi-Asynchronous

Scenario 3:

node i

node j

c) Semi-asynchronous:
System Model

Sleep Scheduling Model

Frame Structure

One Slot per Frame!
Existing synchronous protocols only use slot 0

GAPS!
Existing asynchronous protocols does NOT work

Search Interval \((T_{SI})\)

Determined by [AEB, STAR]

---


Scenario 4: Individual Differences

- **Conference**
  - **Conditions**
    - Each hour you choose to be either at (1) the conference hotel or (2) your room
  - **Objective**
    - Two persons (you and Professor) want to meet each other
    - With the least number of visits to the conference hotel
  - **Knowledge**
    - Each person knows that the other person will likely be at the conference today around the lunch buffet
    - Professor loves to be at the conference hotel, but you don’t

- What would you do?
  - Probabilistic?

- “Somewhat/Loosely scheduled” (Semi-Asynchronous)
- Individual differences in the “Time Window”
System Model

- **Synchronization Error Model [TTP, RBS, TPSN]**
  - Sync error *increase* with time
  - Sync error *decrease* by means of synchronization algorithm

\[ P_{err} = Pr(|\varepsilon| \leq \varepsilon_{max}) = \text{erf} \left( \frac{\varepsilon_{max}}{\sqrt{2}\sigma_{\varepsilon}} \right) \]  

- \( L_f \) changes with time

---


Problem Definition

- Need to guarantee overlapping of at least one active slot between two frames with time shifts

- **Semi-Asynchronous:**
  - \( L_f < T_{SI} \)
  - **Bounded Error \( \rightarrow \) Bounded Shift:** \( \pm n_{\text{max}} = \pm \left\lceil \frac{\epsilon_{\text{max}}}{L_s} \right\rceil \)
  - cf.) Synchronous \( (L_f = "1 slot") \), Asynchronous \( (L_f = T_{SI}) \)

---

**Definition 1. (Shift Schedule)** Given \( i \in \mathbb{N}_0 \) and a universal set \( U = \{-i, -i + 1, \ldots, i - 1, i\} \), let \( E_i \subseteq U \) and \( E_i \neq \emptyset \). We call \( S(E_i, h) = i + h \) a shift schedule of \( E_i \).

**Definition 2. (Shift Intersection Property)** Given \( E_i \) and \( E_j \), shift intersection property is satisfied if and only if \( S(E_i, h_1) \cap S(E_j, h_2) \neq \emptyset \) for all \( h_1 \in \{-i, \ldots, i\} \) and \( h_2 \in \{-j, \ldots, j\} \).

\[
P_{\text{shift}}\{E_i, E_j\} = \frac{1}{2(i + j) + 1} \sum_{h=-i-j}^{i+j} \beta_h
\]  

where \( \beta_h \) is a binary variable determined as

\[
\beta_h = \begin{cases} 
1 & \text{if } S(E_i, h) \cap S(E_j, 0) \neq \emptyset \\
0 & \text{if } S(E_i, h) \cap S(E_j, 0) = \emptyset.
\end{cases}
\]
Problem Definition – An Illustrative Example

\[ \text{E}_{3,3} \]
\[ \text{E}_{4,3} \]

\( n_g = 3 \)
\[ S(\text{E}_{3,3}, 0) \]
\[ S(\text{E}_{4,3}, -7) \]
\[ S(\text{E}_{4,3}, -6) \]
\[ \vdots \]
\[ S(\text{E}_{4,3}, 0) \]
\[ S(\text{E}_{4,3}, +6) \]
\[ S(\text{E}_{4,3}, +7) \]
Proposed Protocol

- CGI: Traditional Method (= Continuous)

\[ E_n = \{-n, -n+1, \ldots, n-1, n\} \quad (7) \]
\[ R = \sum_{h=-n_{\text{max}}}^{n_{\text{max}}} f(h) \left( \frac{2|h| + 1}{2n_{\text{max}} + 1} \right) \quad (8) \]

- DSA: Basic Idea - Frame \( E_n \) consists of…
  - “\( c \)” consecutive active slots (for all nodes)
  - Followed by active slots that are “\( c \)” slots apart (for individual error)
  - An active slot on the last slot
  - The beacon sent every \( c \) slots (not just on slot 0 as in CGI)

\[ E_n = \{-n, \ldots, -n+c, -n+2c, \ldots, -n+c \left[ \frac{2n+1}{c} \right], n\} . \quad (9) \]
\[ R = \sum_{h=-n_{\text{max}}}^{n_{\text{max}}} f(h) \left( \frac{c + \left[ \frac{2|h|+1}{c} \right] - 1}{2n_{\text{max}} + 1} \right) \quad (10) \]
Numerical Result

- **Comparison of Active Ratio (Vs. Asynchronous & CGI)**

![Graph showing comparison of active ratio](image)

- Larger error \(n_{max}\) \(\rightarrow\) larger active ratio
- Outperforming region: large sync error to \(T_{SI}\) ratio

Fig. 3. Comparison of average active ratio
Simulation Result

- Simulation Setting
  - ONE Simulator [ONE] + Sync Algorithm + Sleep Scheduling Protocol
  - Mobile Nodes, Sparse Density

Table II: Default Simulation Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Time</td>
<td>86400 s</td>
</tr>
<tr>
<td>Map Size</td>
<td>5000 m x 5000 m</td>
</tr>
<tr>
<td>Movement Model</td>
<td>Random Waypoint</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>50</td>
</tr>
<tr>
<td>Node Speed</td>
<td>1.0 m/s</td>
</tr>
<tr>
<td>Pause Time</td>
<td>0 - 300 s</td>
</tr>
<tr>
<td>Radio Transmission Range</td>
<td>100 m</td>
</tr>
<tr>
<td>Search Interval ($T_{SI}$)</td>
<td>1.0 s</td>
</tr>
<tr>
<td>ATIM Window Length ($L_s$)</td>
<td>10 ms</td>
</tr>
<tr>
<td>Power Consumption Model</td>
<td></td>
</tr>
<tr>
<td>Transmission Mode ($P_{tx}$)</td>
<td>42 mW</td>
</tr>
<tr>
<td>Receive Mode ($P_{rx}$)</td>
<td>36 mW</td>
</tr>
<tr>
<td>Idle Mode ($P_{idle}$)</td>
<td>24 mW</td>
</tr>
<tr>
<td>Sleep Mode ($P_{sleep}$)</td>
<td>0.02 mW</td>
</tr>
</tbody>
</table>

Simulation Result

Simulation Scenarios

1. Reference Node based Synchronization [DTP]
   - At least one time server
   - Client nodes tune to the reference time in the server
   - Sync Error: increase 100 us/s

2. Relative Clock Synchronization [SA, AD]
   - No time server
   - Nodes cooperatively tries to minimize relative time error
   - Averaging between contacted nodes
   - Sync Error: Initial configuration (< 100 ppm)

---


Simulation Result

- Reference Node Based

(a) Average clock offset   (b) Cumulative Average Energy Consumption

Fig. 4. Reference node based clock synchronization

- Lower speed $\rightarrow$ Slower Convergence Speed $\rightarrow$ Larger error $\rightarrow$ Higher Energy Consumption
- EC of DSA: 20\% @ 1.0 m/s, 30\% @ 5.0 m/s
Simulation Result

- (Distributed) Relative Synchronization

(a) Average clock offset

(b) Cumulative Average Energy Consumption

Fig. 5. Distributed clock synchronization

- Lower speed $\rightarrow$ Slower Convergence Speed $\rightarrow$ Larger error $\rightarrow$ Higher Energy Consumption
- Converge (assuming control parameters are all deterministic)
- EC of DSA $<$ EC of CGI (especially for large errors, fixed $c$)
Conclusion

- **Clock Synchronization & Energy Consumption**
  - Practical to assume some level of clock accuracy with error bounds

- **Distributed Semi-Asynchronous Sleep Scheduling Protocol (DSA)**
  - Periodic Active Slots + Minimum Active Slots
  - (not shown) Optimized for given distribution of synchronization error

- **Performance Evaluation**
  - Comparison of Active Ratio vs. Existing Protocols
  - Simulation Results

- **Further Work**
  - Overhead from using synchronization protocols for loosely synchronized clocks vs. Asynchronous protocols
  - **Adaptive** version of DSA for time varying distribution of sync error
  - Extensive simulations
Adaptive DSA?

- Some problems in the Relative synchronization simulation results!
- DSA is already adaptive to the time varying synchronization error
  - Min c is fixed!
- “c” can be adjusted using the distribution of the long term synchronization error

![Graphs](image.png)

(a) Average clock offset   (b) Cumulative Average Energy Consumption

---

Fig. 5. Distributed clock synchronization
Thank You

Questions & Comments?

David (Bong Jun) Choi
bjchoi@engmail.uwaterloo.ca