On-line Data Link Layer Scheduling in Wireless Networked Control Systems

Shengyan Hong+, X. Sharon Hu+, Tao Gong*, Song Han*

+University of Notre Dame, USA
*University of Connecticut, USA
Research Overview

Network-based Rehabilitation System

Cyberphysical Avatar

Real-Time Analytics Platform for Process Control

Remote and Real-time Welding System
However ...

Common assumption:

“The communication schedule of real-time wireless sensor and actuator networks, once constructed and distributed, stays unchanged.”
However ...

Common assumption:

“The communication schedule of real-time wireless sensor and actuator networks, once constructed and distributed, stays unchanged.”
Reality is ...

External disturbance happens sporadically in Wireless Networked Control Systems (WNCS)

More communication resources are required temporally for close monitoring and prompt response

Existing task models and static scheduling approaches need to be extended
What We Want to Achieve

- Design an on-line scheduling framework for a WNCS
  - Upon detecting a physical disturbance
  - Determine a dynamic schedule interval
  - Generate and distribute a temporary dynamic schedule $S$
  - Meet as many periodic packet deadlines as possible
Outline

- System model
- Motivation and problem formulation
- Proposed OLS framework
- Proposed OLS-mDP algorithm
- Simulation results
- Summary and future work
System Model

- **Infrastructure of WNCS**
  - A gateway, sensors, actuators, and relay nodes sharing a channel

- **Task Model**
  - Rhythmic task $\tau_0$, broadcast task $\tau_{n+1}$, periodic tasks $\tau_i$'s ($1 \leq i \leq n$)
  - An instance of task is a packet $\chi_{i,m} (0 \leq i \leq n+1)$
  - Sending a packet at hop $h$ is a transmission $\chi_{i,m}(h) (0 \leq i \leq n+1)$
Rhythmic Task Model

Period

Nominal state

Deadline

Rhythmic state

\( r_{0,m} \)

\( d_{0,m} \)

\( r_{0,m+1} \)

\( d_{0,m+1} \)

\( r_{0,m+2} \)

\( r_{0,m+R} \)

\( r_{0,m+R+1} \)

\( r_{0,m+R+2} \)

\( r_{0,q^*} \)

\( r_{0,q^*+1} \)

\( r_{0,q^*+2} \)

\( t_{n->r} \)

\( t_{r->n} \)
Requirement of a dynamic schedule (denoted to be $S$)

- The WNCS initially follows the static schedule $\bar{S}$
- From $t_{n->r}$, the workload is changed and $\bar{S}$ is not enough
- $t_{st}$: start point, $t_{sw}$: switch point

**Diagram:**

- Follow $\bar{S}$: $t_1$ to $t_2$
- Generate a dynamic schedule $S$
- Broadcast $S$ at $t_{st}$
- Use $\bar{S}$ & $S$ at $t_{sw}$
- Reuse $\bar{S}$ at $t_{r>n}$
- $t_n->r$
Outline

- Introduction and system model
- Motivation and problem formulation
- Proposed OLS framework and OLS-mDP algorithm
- Simulation results
- Summary and future work
### Motivational Example

<table>
<thead>
<tr>
<th>Task</th>
<th>Period</th>
<th>Relative Deadline</th>
<th>Rhythmic task enters the rhythmic state</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tau_0)</td>
<td>9</td>
<td>6</td>
<td>(P_{0,1}, P_{0,2}, D_{0,1}, D_{0,2})</td>
</tr>
<tr>
<td>(\tau_1)</td>
<td>9</td>
<td>7</td>
<td>N/A, N/A, N/A, N/A</td>
</tr>
<tr>
<td>(\tau_2)</td>
<td>9</td>
<td>9</td>
<td>N/A, N/A, N/A, N/A</td>
</tr>
</tbody>
</table>

- Consider the situation that rhythmic task \(\tau_0\) enters the rhythmic state at \(t_{n->r}=9\)
- Select switch point \(t_{sw}\) to 18
Two Existing Methods

Still use static schedule results in the deadline miss of rhythmic packet $X_{0,3}$

Dynamic schedule by EDF introduces 7 slot assignment updates
The 27th Euromicro Conference on Real-Time Systems

Proposed dynamic schedule

Proposed schedule does not drop any packet but only needs to update 3 slot assignments
On-Line Scheduling with Bounded Overhead (OLSBO)

- **Objective**
  - Minimize the number of dropped periodic packets

- **Constraints**
  - Each rhythmic packet must meet its deadline
  - Switch point has a switch point upper bound $t_{sw}^u$
  - Update bound $\Delta^u$

- **Time requirement**
  - $S$ should be generated within $[t_1, t_2]$
Reduce Solution Space

- **Extra constraint**
  - If $t$ is assigned to $\chi_{i,m}(h)$ in $\tilde{S}$ and $\chi_{i,m}(h)$ is released but unfinished at $t$ actually, then $t$ cannot be set to idle in $S$

- **With extra constraint, OLSBO becomes OLSBO$^+$**
  - Each rhythmic packet must meet its deadline

- **Theorem 2**
  - If $S$ is an optimal solution to OLSBO, then OLSBO$^+$ has an optimal solution $S^+$ such that
  - $S$ and $S^+$ drop the same number of periodic packets and
  - $S$ and $S^+$ update the same number of slots
Challenges in Solving the Problem

- **Which slot can be a switch point candidate** $t_{SW}^c$?
  - Any slot within the interval $[t_{n-r}, t_{SW}^u]$

- **Which transmissions should be scheduled by S?**

- **How to construct a dynamic schedule?**
  - Satisfy constraints of OLSBO$^+$
  - Generated within $[t_1, t_2]$
Switch Point Candidates

- $S$ can be employed immediately from $t_{sw}$ if
  - $t_{sw} \geq t_{r \rightarrow n}$ and $r_{0,p^*} = r_{0,q^*}$?
  - All rhythmic packets released earlier than $r_{0,q^*}$ meet their deadlines

(See details in Theorem 3)
Switch Point Candidates

What are the switch point candidates?

- All the nominal release times within \([t_{r->n}, t_{SW}^u]\)
- \(S\) can be reused immediately after \(t_{SW}^c\)
Construction of Transmission Set

- **Transmission set** \( \Psi(t_{SW}^c) \)
  - Rhythmic transmissions released earlier than \( t_{SW}^c \), but not finished earlier than \( t_{st} \) according to \( \bar{S} \)
  - Periodic transmissions supposed to be finished within \([t_{st}, t_{SW}^c]\) in \( \bar{S} \)
Outline

- Introduction and system model
- Motivation and problem formulation
- Proposed OLS framework and OLS-mDP algorithm
- Simulation results
- Summary and future work
On-Line Scheduling (OLS) Framework

1. Determine a set of switch point candidates
2. Given a switch point candidate $t_{sw}^c$, construct a transmission set $\psi(t_{sw}^c)$
3. Generate a basic dynamic schedule
4. Generate a regular dynamic schedule
5. Select the best schedule

There is at least one unchecked switch point candidate.
On-Line Scheduling (OLS) Framework

1. Determine a set of switch point candidates

2. Given a switch point candidate \( t^{c}_{sw} \), construct a transmission set \( \psi(t^{c}_{sw}) \)

3. Apply a time slot assignment algorithm, such as dynamic programming, EDF and others

4. Select the best schedule

There is at least one unchecked switch point candidate
Characteristics of OLS

- Primary/backup schedules
  - Both finish all rhythmic packets on time
  - Primary: meet as many deadlines of periodic packets as possible
  - Backup: drop all periodic transmissions
- If more than $\Delta^u$ updated slots are updated,
  - Select a periodic packet to drop
- Every time a schedule $S[t_{st}, t_{sw}^c]$ is generated,
  - Reuse partial slot assignments from $S[t_{st}, t_{sw}^c]$
  - When generating schedule for $t_{sw}^c (> t_{sw}^c)$
Exact Dynamic Programming (eDP)

- **Key idea of eDP**
  - Give $S[t_{st}, t-1]$
  - Construct $S[t_{st}, t]$ by determining the assignment of slot $t$
  - Such a process is repeated until $t$ reaches $t_{sw}^c$

- **Challenges of eDP**
  - Which transmission can use $t$
  - Which schedules should be kept if multiple $S[t_{st}, t-1]$’s are found for $t$
  - No rhythmic packet misses its deadline
  - Satisfy the constraint of max allowed updated slot number
  - Extra constraint on space reduction is satisfied
Exact Dynamic Programming (eDP)

- \( t = t_{st}, \varphi_p = \varphi_c = \emptyset \)
- Get an \( S[t_{st}, t-1] \in \varphi_p \)
- Construct \( S[t_{st}, t] \) by determining \( S[t] \)
- Determine if \( S[t_{st}, t] \) is acceptable. If yes, \( \varphi_c = \varphi_c \cup S[t_{st}, t] \)
- If \( t < t^c_{SW} \)
- \( \varphi_p = \varphi_c, \varphi_c = \emptyset, t = t + 1 \)
- Select the best schedule \( S[t_{st}, t^c_{SW}] \) from \( \varphi_p \)
- There is at least one unchecked \( S[t_{st}, t-1] \in \varphi_p \)
- Satisfy all the constraints of problem
- Achieve the objective of OLSBO$^+$
Property of eDP

- **Optimality of eDP**
  - Assume $t_{sw}^c$ is switch point
  - eDP finds a schedule if and only if there exists a solutions to OLSBO$^+$
  - The found solution minimizes the objective function of OLSBO$^+$

- **Disadvantage of eDP**
  - Construct an exponential number of $S[t_{st}, t]$’s for time slot $t$
  - Use all $S[t_{st}, t]$’s when constructing $S[t_{st}, t+1]$’s

- **Modified Dynamic Programming (mDP)**
  - Limit the maximum allowed number of maintained $S[t_{st}, t]$’s for time slot $t$ to value $\beta$
  - Time complexity of mDP is $O(t_{sw}^c n^2 \beta^2)$
  - If there are $\kappa$ switch point candidates, OLS will call mDP for $\kappa$ times
Outline

- Introduction and system model
- Motivation and problem formulation
- Proposed OLS framework and OLS-mDP algorithm
- Simulation results
- Summary and future work
Simulation Setup

- **Infrastructure**
  - 36 nodes in a 6X6 square mesh grid
  - One gateway, 17 sensors and 18 actuators
  - Connected graph

- **Task sets**
  - 20 groups of task sets, labelled as “(n+2)-u”
    - n is the number of periodic tasks, 1 rhythmic task and 1 broadcast task
    - u is the utilization level of all tasks in a task set

- **Baseline algorithm**
  - mEDF is a slot assignment algorithm using EDF to generate dynamic schedules
Number of Solved Task Sets

- OLS-mDP can solve more task sets by 41% on average (at most 108%)
Number of Solved Task Sets

- OLS-mDP can solve more task sets by 11% on average (at most 31%)
Average Drop Rate

- OLS-mEDF drops more packets by 122% on average (at most 321%)
Average Drop Rate

- OLS-mEDF drops more packets by 128% on average (at most 263%)
Summary and Future Work

Summary

- Introduce an on-line data link layer scheduling problem
- Propose a framework which employs an effective heuristic to solve the problem
- Present simulation results

Future work

- Extend system model
  - Support multiple communication channels
  - Allow multiple rhythmic tasks to enter the rhythmic state simultaneously
- Design more efficient algorithm to generate the dynamic schedule
- Implement the algorithm in a WNCS testbed