Joint Network and Computing Resource Scheduling for Wireless Networked Control Systems

Peng Wu[1] Chenchen Fu[2] Minming Li[2] Yingchao Zhao[3]
Jason Xue[2] Song Han[1]

[1]University of Connecticut

[2]City University of Hong Kong

[3]Hong Kong Caritas Institute of Higher Education

-RTSS 2018 WIP-

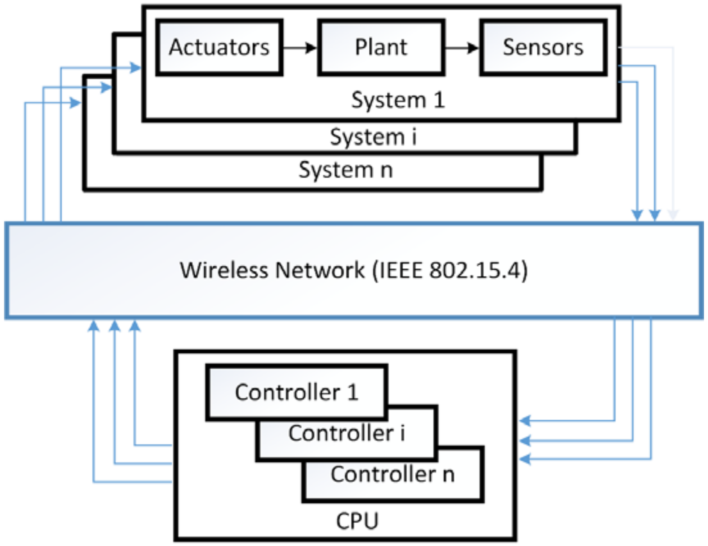
Wireless Networked Control systems

WNCSs enable wireless networked control, actuating, and feedback

- great advantages in applications such as industrial monitoring and control [1], building automation [2], manufacturing [3], etc.

Structure of WNCSs from the perspective of Control Community

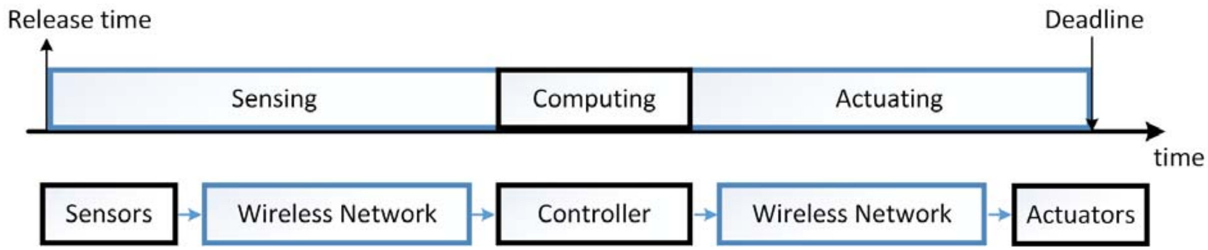
- Multiple control loops
- Shared CPU
- Shared communication medium
- Designated end-to-end timing requirement
- Strict execution order
  - Sensing
  - Computing
  - Actuating



[1] H. Ramamurthy, B. S. Prabhu, and R. Gadh, "Wireless industrial monitoring and control using a smart sensor platform," IEEE Sensors J., vol. 7, no. 5, pp. 611–618, May 2007.
[2] Y.-J. Wen and A. M. Agogino, "Wireless networked lighting systems for optimizing energy savings and user satisfaction," in Proc. IEEE Wireless Hive Netw. Conf., Aug. 2008, pp. 1–7.
[3] J. J. Evans, "Wireless sensor networks in electrical manufacturing," in Proc. Elect. Insul. Conf. Elect. Manuf. Expo, Oct. 2005, pp. 460–465.

Joint Resource Scheduling (JRS) Model

We observe one control loop of WNCSs from the perspective of real-time system community



The joint task model is defined as:

$$\tau : \vec{s}(h) \rightarrow \vec{c}(m) \rightarrow \vec{a}(k)$$

- Sensing segment $\vec{s}$ → uses network resource
- Computing segment $\vec{c}$ → uses CPU resource
- Actuating segment $\vec{a}$ → uses network resource

where we use h, m and k to represent the execution/transmission time of the three segments, respectively.

Assumption: Unit size of network and computing resource slices are the same. The wireless network has only one channel and the CPU is a preemptive uniprocessor.

Joint Scheduling Problem Formulation

Joint Network and Computing Resource Scheduling (JNCRS) Problem:

- Given a set of real-time joint tasks $\{\tau_1, \tau_2, ..., \tau_n\}$. Each task τ_i is associated with a release time r_i, a deadline d_i, and consists of three consecutive and dependent segments. The objective of the JNCRS problem is to find a feasible schedule so that the deadlines of all real-time tasks are met.

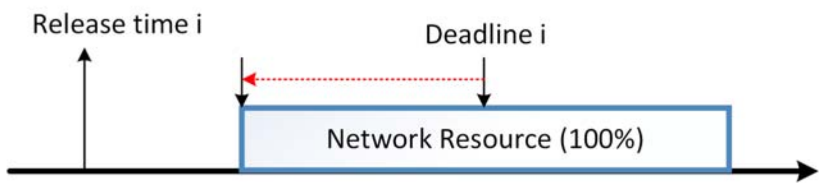
Divide the **JNCRS problem** into 4 subproblems based on the size of the execution time of each segment.

Task Model	Complexity or Solution
$\tau : \vec{s}(1) \rightarrow \vec{c}(1) \rightarrow \vec{a}(1)$	Polynomial-time solvable (Alg.1)
$\tau : \vec{s}(1) \rightarrow \vec{c}(m > 1) \rightarrow \vec{a}(1)$	Polynomial-time solvable (Alg.2)
$\tau : \vec{s}(1) \rightarrow \vec{c}(m > 0) \rightarrow \vec{a}(1)$	Unknown
$\tau : \vec{s}(h) \rightarrow \vec{c}(m) \rightarrow \vec{a}(k)$	NP-hard

Joint Scheduling Algorithms (Alg.1 and Alg.2)

Alg. 1: Unit-Size Joint Resource Scheduling (JRS-US) algorithm

- The key idea is to identify all the intervals with network resource utilization of 100% and modify the deadline of a sensing/actuating segment if its deadline is within but its release time is outside that interval.



- After the deadline modifications, Earliest Deadline First (EDF) can be used.

Alg. 2: Joint Resource Scheduling (JRS-US) algorithm under 1-m-1 (m>1)

- Modify deadline as Alg. 1, but also needs to modify release time.
- Design Conditional/definite release time/deadline for the network segment, due to different preemption cases of computing segment.

Future Work

- Continue study the remaining subproblems and design the efficient scheduling algorithms on
  - 1-m-1 model (m>0)
  - general model

- Evaluate effectiveness and practicability of the proposed algorithms in real-life systems by implementing them on our WNCS testbed.

- Relax the assumptions on
  - number of wireless network channels and number of CPU cores
  - network topology (sensors, actuators, controllers)