Dynamic Packet Scheduling for Wireless Control System with Multiple Physical Systems

Wenchen Wang, Daniel Mosse
Real-time Systems Lab
Computer Science Department
University of Pittsburgh

Daniel Cole
Nuclear Engineering Research Lab
Mechanical Eng and Materials Science Department
University of Pittsburgh

***Abstract*—Wireless control systems (WCS) have been gaining a lot of attention, due to its easy deployment comparing to wired control systems. However, network delay and packet losses caused by wireless network significantly influence the control system performance for different control application demands. In addition, surprisingly, few research works study the WCS with multiple control systems given it is a new trend with the Industrial Internet of Things. Motivated by these observations, we propose a dynamic packet scheduling solution to minimize the performance error of WCS with multiple control systems, by dynamically determining the packet priorities of different control systems and characteristic of network paths. We consider two cases for network path selection: (1) network delay only by developing a worst-case end-to-end delay analysis; (2) network delay + reliability by proposing a new network quality model. We conducted a case study of a modern nuclear power plant with several Small Modular Reactors. We validated our end-to-end delay analysis and show accuracy within 2% of a state of the art simulator. Also, our extensive simulation results varying noise and redundancy levels show that dynamic scheduling solution is effective and can compensate for the wireless delays and loss.**

I. Introduction

Wireless control systems (WCS) comprise controllers, sensors, relay nodes, and actuators connected via a wireless network. WCSs operating over multi-hop wireless (sensor) networks have received significant attention in recent years [4], [8], [11], [18], due to the ease of deployment. However, *network-induced imperfections* (e.g., network delay and packet losses) degrade control system performance, especially when the physical system is undergoing changes. Prior research [8], [7], [18] has modeled this impact through mathematic analysis or case studies for wireless control system with one single physical system. To the best of our knowledge, this is the first study on wireless real-time control system with multiple physical systems. It is an important problem, since the situation of multiple physical systems utilize one shared wireless network will be increasingly common, especially in IoT (Internet of Things) systems and IIoT (Industrial IoT).

In this paper, we consider a WCS of multiple control systems with one shared wireless network. In a shared network, a real-time wireless network typically has multiple different network paths to transmit messages in parallel (some paths may have redundancy). Each path may have a different characteristic in terms of delay and reliability (e.g., in WirelessHart Protocol [1], one can choose between more reliable and higher delay versus lower delay but less reliable paths).

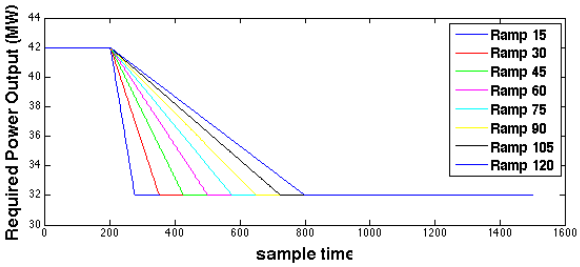

Fig. 1: Control system power reference functions; each sample time is 20^{-1}s

Also, different control system may have different application demand. For example, one control system has urgent demand, such as reducing temperature by 10°C within one minute while another system has less urgent demand, such as increasing the temperature by 2°C within one hour. Our solution follows our intuition: to get better overall control system performance, we should assign the messages of the control system with urgent demand to fast and reliable paths and assign the messages with less urgent demand to slower or less reliable paths.

To test our intuition, we implemented a wireless control system for a nonlinear primary heat exchanger (PHX) system in a nuclear power plant (NPP), whose main function the exchange of heat from inside to the outside of the reactor. Figure 1 shows 8 different reference functions (ramp functions) of a PHX when the controller decides to reduce the output power from 42MW to 32MW within different amount of time (control sampling period is 0.2s). For example, ramp30 means to reduce the power from 42MW to 32MW within 30s.

To motivate how important loss and delay are, in Figure 2 we show the effect of network delays and power output reference functions (from Figure 1); we measure system performance through power RMSE[1] (Root Mean Square Error of the power output). We also varied sensing delivery ratio (SDR, the percentage of sent messages arriving at the controller from measurement sensors) and actuation delivery ratio (ADR, the percentage of sent messages that arrive at the actuator from the remote controller), and we apply the same network routing scheme for both sensing and actuation. As SDR and ADR

[1]The metric measures the RMS error between the closed-loop responses with wireless and wired control (we assume there are no packet drops and no network delay in wired control).

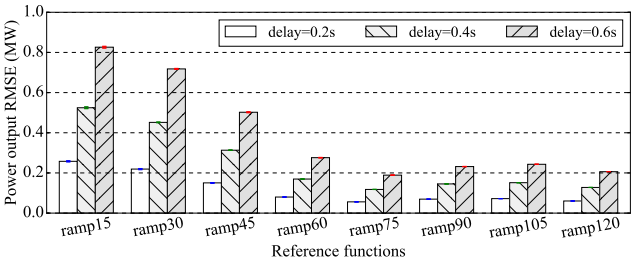

Fig. 2: Power output RMSE for different reference functions with different network delay for a single PHX (DR=0.9 with random packet drop)

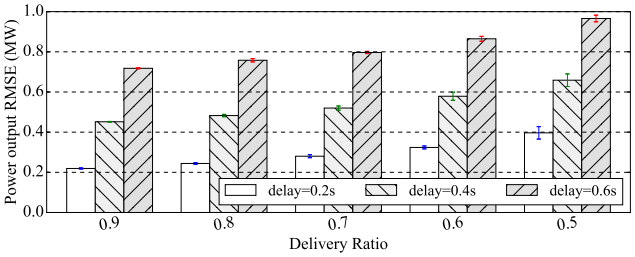

Fig. 3: Power output RMSE for different reference functions with different network delay and delivery ratio for a single PHX (reference function: ramp30)

are symmetric, we call it DR and show only DR=0.9 in the rest of our paper (other values of DR show similar trends of the RMSE). Figure 3 shows the power output RMSE for different network delays and DRs when the reference function is ramp30 (similar trends for the other reference functions).

We have two observations:

1) As shown in Figure 2, for the same network delay and DR, the steeper the reference function, the larger the RMSE. This is because when the reference function is steep, it requires the control system to reduce its power output aggressively (in much less time), and thus it will have a more transient response, causing larger RMSE. However, if the time required to change the power output is longer then 60 seconds (i.e., ramp60), the control system has approximately the same error due to the slow reaction required by the NPP.
2) As shown in Figure 3, for the same reference function, the higher the network delay and lower delivery ratio, the larger the RMSE. For the steeper reference functions, the network delay and delivery ratio become more significant on the control system performance.

Based on the two observations above, network imperfections will impact each control system differently, depending on the control system's application demand (e.g. reference function).

Given the above, our goal is to reduce the overall control system RMSE caused by network-induced imperfections. We propose an approach to dynamically schedule measurement packets of different physical systems to the appropriate network paths (with redundancy or not) using a TDMA approach for both measurements and actuation packets. Our approach has two parts: (1) priority assignment of the measurement packets (highest priority for most urgent physical plant); (2) network path selection. For the second part, we consider two cases: (2a) **network has no packet losses.** We came up with an end-to-end delay analysis for network paths in a general case when it is possible the network deadline is greater than its period. We assign the highest priority packets to the fastest network path. To the best of our knowledge, this is the first paper that discusses the end-to-end delay analysis for network deadline greater than the control sampling period in the real-time WCS with traffic in both directions. (2b) **network with packet losses.** We propose a network path quality model to combine the impact of network delay and packet loss on the control systems together. Quality here is from the perspective of the control system: higher quality brings higher performance to the control system, which fills the gap between network imperfection and control system performance. The highest priority packet is assigned to the highest quality path.

To evaluate our approach, we first show how our analysis can determine the worst-case end-to-end delay on the TDMA network with multiple paths, deadlines longer than periods, and traffic in two directions. Then, we carried out a case study on three primary heat exchanger systems (PHXs) in a modern, SMR (Small Modular Reactor)-based NPP. Note that our approach is general and can be applied to other WCSs. The results demonstrate: (1) our worst-case end-to-end delay analysis is accurate (2) our packet schedule approach is effective and able to compensate for delay and packet loss incurred by the network during the transition between steady-states of multiple physical systems when they vary their demands simultaneously, and create a performance close to a wired network.

The contributions of this paper encompass:

- a heuristic method to determine the priority of physical system measurement packets;
- end-to-end delay analysis for network paths (redundant and not) using TDMA, when the network deadline is greater than its period;
- a general network quality model for wireless control system considering both network delay and packet loss;
- a case study that evaluates our worst-case end-to-end delay analysis and dynamic packet scheduling approach for a NPP.

II. Related Work

Although a WCS has the advantages of easy deployment and maintenance, one of its biggest challenges is network-induced imperfections [22]. The solutions of recent research works are typically divided into three categories: control only, network only, and control+network co-design solutions.

Control solutions for dealing with network imperfections are promising. The closed-loop system is modeled as a switched system in [9], considering both time delays and packet losses at the actuator nodes. Other examples include [10], [13], [17] that use the model-based predictive control approach, which

obtains a finite number of future control commands besides the current one for handling both time-varying delays and packet drops. However, these works only consider network as a black box and there is no packet scheduling mechanism taking into account different control system application demands.

For the network solutions, online dynamic link layer scheduling algorithms have been proposed [5], [23] to meet the deadline of a rhythmic flow and minimize the number of dropped regular packets in a centralized and distributed way, respectively, based on a rhythmic task model proposed in [6]. However, these two works did not consider different control system application demands. Also, they assume network external disturbances occur sporadically, which is different from ours. [16] and [14] analyze the worst-case end-to-end delay for source and graph routing based on wirelessHart standard to guarantee the real-time communication in WCS. However, they both consider the network flow deadlines are smaller than their periods. We focus on a general case when it is possible that the transmission deadlines are greater than their periods. To the best of our knowledge, there is no other works studying this case before, but it is common in real-time WCSs [18].

For the co-design solution that is the closest to ours, the integration of wireless network and control are studied in [2], [8], [7], [12], [18], [20], [21]. The co-design of fault-tolerant wireless network and control in nuclear power plants are studied in [18], [20], [21]. The work in [18] shows that the network delay and reliability both could affect the control system performance. In [8], the authors show how the network reliability affects the control system failure ratio via a water tank case study. In [7], the authors discuss how the routing scheme affects the control system performance. A co-design of network topology conditions and control system stability is explored in [11]. In [2], the authors design a control-aware random access communication policy of shared wireless medium for multiple control systems to guarantee the system stability. However, there are still three important shortcomings of these approaches. First, there is still a gap to describe the relationship between network performance and control system performance. There is no such a model to describe this gap, and thus we propose a general network quality model to describe this gap in terms of network delay and message loss. Second, these works discussed either wireless control system with single physical system or wireless control system with multiple control system considering only the system stability. Third, none of them addressed the interaction between dynamic packet scheduling and control, which is the focus of our work.

III. Problem Formulation and Solution Overview

A. Problem formulation

There are N physical systems that share one wireless network. We define a series of time steps T=$\{t_0, t_1, ..., t_w\}$, where T is the interval of time during which any physical system is in transition (system is in non-steady state). We have a set of N reference functions R=$\{r_1(T), r_2(T), .., r_N(T)\}$ that define different physical system application demands. Similar to [15], there are k choices of network paths/flows P=$\{p_1, p_2, ..., p_k\}$, each path $p_i \in P$ is characterized by a delay D_i and a delivery ratio $dr_i(t)$, which depends on the redundancy in the paths as well as the scheduling and routing schemes. Each network path delivers one message with the measurements of one physical system to the remote controller; the controller sends messages to the actuators after running its control algorithm.

For each physical system i, we can compute $RMSE_i$, defined in Equation 1, where $wired_i(t)$ and $wireless_i(t)$ are the wired (no losses, no delay) and proposed control system power output of physical system i at time t. Our objective is to minimize the $RMSE_{total}$, defined by Equation 2. Our scheme produces the network path selection for physical systems over all time steps, S= $\{[s_1(t_0), s_2(t_0), ...s_N(t_0)]$, $[s_1(t_1), s_2(t_1), ...s_N(t_1)]$, ..., $[s_1(t_w), s_2(t_w), ...s_N(t_w)]\}$, where $s_i(t)$ is the selected network path for the i^{th} physical system transmission at time t.

$$RMSE_i = \sqrt{\frac{1}{w}\sum_{j=0}^{w}(wired_i(t_j) - wireless_i(t_j))^2} \quad (1)$$

$$RMSE_{total} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} RMSE_i^2} \quad (2)$$

B. Solution Overview

In essence, our solution is to determine which network path to transfer which physical system's measurement for a series of time steps T. Let us consider a brute-force way to solve the problem. We first consider the case when all the network paths are very reliable ($\forall i, \forall t\ dr_i(t) = 1, p_i \in P, t \in T$). At each time step, we try all possible combinations of network paths $C(N, k)$ and choose the best path selection over w time steps, S that has minimum $RMSE_{total}$ over w time steps. The complexity is $O(C(N, k)^w)$. Even if we simplify our problem by assuming that $\forall i, \forall t\ dr_i(t) = 1$, the complexity still remains exponential. The complexity when network paths have message loss ($\forall i, \forall t\ dr_i(t) < 1, p_i \in P, t \in T$) would be much higher. Therefore, though the brute force approach is optimal, it is impractical due to its high computation time and storage costs.

To alleviate these problems, we propose to solve the problem in two steps. We first propose a heuristic method to determine which physical system has the most urgent application demand and impose a priority order for the measurement packets (Section IV). We then consider two cases: (1) $dr_i(T) = 1$, we develop an analysis of the worst-case end-to-end network delay for each network path and assign the most urgent measurement packet to the network path with the shortest delay (Sections V and VI); (2) $dr_i(T) < 1$, we propose a network path quality model to consider both end-to-end delay and reliability of network path. We assign the most urgent measurement packet to the network path that can

deliver the measurement with as high reliability and as short delay as needed by the specific physical system to result in small RMSE to the control system (Section VII).

IV. Priority Assignment of Measurement Packets

The basic idea of priority assignment of measurement packets is to give high priority to measurement packet of the system that would yield low performance, to avoid increasing RMSE and thus $RMSE_{total}$. We propose a heuristic method to determine the measurement packet priority. Since our objective is to minimize the control system $RMSE_{total}$, the heuristic is based on the following: the higher RMSE, the more necessary to transmit its message as soon and reliably as possible (thus reducing the RMSE). Since we cannot get the RMSE comparing with wired control system output at run time, we track each system $rRMSE_i(t)$ comparing with its reference function $r_i(t)$ for each physical system at run time at each time step, shown in Equation 3, where $wireless_i(j)$ is the i^{th} system measured power output at time j. At current time step t_x, we calculate $rRMSE_i(t_x)$, sort the rRMSEs of N physical systems and assign the highest priority to the measurement packet of the system with the highest current $rRMSE(t_x)$.

$$rRMSE_i(t_x) = \sqrt{\frac{1}{x}\sum_{j=0}^{x}(r_i(t_j) - wireless_i(t_j))^2} \quad (3)$$

V. Network Model

In our paper, we focus on a wireless network disjoint with multiple network paths that can transmit messages in parallel. Each path has one or more lines of relay nodes for redundancy. As shown in Figure 4, there is one primary line of relay nodes (marked as black) and zero or more lines of backup relay nodes (marked as gray). We use the bitvector protocol [19], which modifies the TDMA scheduling for optimizing redundancy and guaranteeing real-time transmissions. The relay nodes broadcast messages level by level towards the controller, then back to the actuator. Within each level, the primary node will broadcast first, then the first, second, and third backup nodes, in order. Therefore, the more relay nodes in the network, the more messages are sent (one message sent per node but received by all nodes in the next level), and thus the higher DR and network delay.

We assume that there are n hops from sensors to the controller (source to destination) and l lines of relay nodes in our path; it takes l time slots on each level to transmit a message (one slot per node). To be reliable, both the sensor nodes and the controller will send out l messages to the relay nodes (i.e. takes l time slots). We denote current time slot as t ($t = 0, 1, 2, ...$), current level as h ($h = 0, 1, ..., n$), and both control sampling period and sensing sampling period the same as p. The number of time slots during one sampling period is $p_s = \frac{p}{\Delta t}$, where Δt is the duration of the time slot. To more easily follow the figures in this paper, we say the message is sent "up" to the controller and "down" to the actuators; thus, message m_0 sent at time $t = 0$ up to

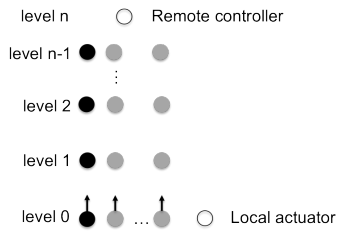

Fig. 4: One network path with one or more lines of relay nodes

the controller is at level $h(m_0) = \lfloor \frac{t}{l} \rfloor$ ($\lfloor \frac{t}{l} \rfloor < n$) and the same message on its way down to the actuator is at level $h(m_0) = 2n - \lfloor \frac{t}{l} \rfloor$ ($\lfloor \frac{t}{l} \rfloor > n$). More generally, a message m_i sent out at time $t = ip_s$, ($i = 0, 1, ...$) traveling up is at level $h(m_i) = \lfloor \frac{t-ip_s}{l} \rfloor$ ($\lfloor \frac{t-ip_s}{l} \rfloor < n$) and traveling down is at level $h(m_i) = 2n - \lfloor \frac{t-ip_s}{l} \rfloor$ ($\lfloor \frac{t-ip_s}{l} \rfloor > n$).

VI. End-to-end Delay Analysis for Network Path

The end-to-end delay analysis is necessary in this paper for two reasons. First, real-time communication is critical for WCS since missing a deadline may lead to system instability or equipment destruction. Knowing the worst-case end-to-end delay allows us to design a network that guarantees meeting the control system deadline. Second, after we determine the measurement packet priority for different control systems (Section IV), we first look into the case when $dr_i(t)$=1 to determine the delay of network paths. We assign the highest priority measurement packets to the path with shortest worst-case delay. Note that in this section, we calculate the worst-case end-to-end delay for network path shown in Figure 4. In our network, there are multiple paths like Figure 4 that can transmit messages in parallel.

We want to determine the worst-case end-to-end delay in the general case, when it is possible that the network/control system deadline is greater than its period, namely when the network delay is greater than the control sampling period. That is, when $2nl > p_s$, a subsequent message will start going up while there is a message going down. We focus on the delay analysis for fixed priority scheduling where message transmissions are scheduled based on most recent message first and oldest message first schemes. We only do our proof based on the most recent message scheme, given that the derivation for the oldest message first is symmetric. We denote the priority of a message m_i as $pri(m_i)$. In other words, the current message will conflict with the messages with higher priority and induce more network delay; our goal is to determine this network delay. We first analyze the conflicts that could happen during the message transmission. We get the schedulability condition (the condition that messages can be delivered to the destination) from the analysis. Based on the schedulability condition, we then estimate the worst-case end-to-end delay by estimating the maximum number of conflicts and the delay without conflict.

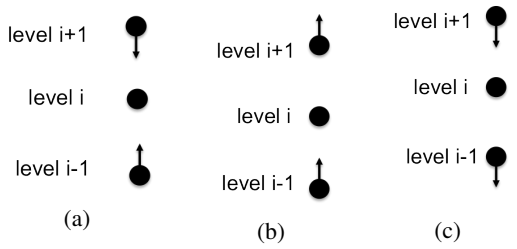

Fig. 5: Three conflict situations

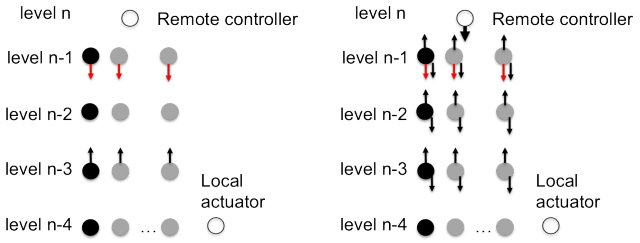

Fig. 6: Conflict situation when $\frac{p_s}{l} = 4$

A. Conflict analysis

There are three canonical situations that two messages will conflict with each other. As usual in wireless networks, conflicts arise when simultaneous transmissions arrive at the same node. The three scenarios are shown as conflict situations 1, 2, 3 in Figure 5a, 5b and 5c, respectively for a single line of relay nodes (no backups), when a message is going up while another is going down, and two messages are going in the same direction but very close together. The conflicts start to happen when the level difference, Δh, of two conflicting messages is 1 or 2 (while the $\Delta h \geq 3$, messages can still make progress).

In general, for conflict situation 1, when the $\Delta h = 1$, it will take $2l$ time slots to resolve the conflict, given that the high-priority message will go up two levels while the low priority message waits. At this time the conflict is resolved. Similarly, when the $\Delta h = 2$, the conflict will be resolved in $3l$ time slots. In general, when message m_i starts going down, the level difference between m_i and m_{i+j}, $\Delta h(m_i, m_{i+j})$ can be odd or even. When Δh is odd, the two messages will make progress on one level at a time, until they are separated by exactly one level. Similarly, when Δh is even, they will make progress until they are separated by exactly 2 levels.

For conflict situation 2 and 3, it will take $4l$ or $5l$ time slots to resolve the conflict, when the level difference is 1 or 2, respectively.

Let us consider consecutive messages, m_0 and m_1, m_2, ..., m_i that are sent at $t = 0$, $t = p_s$, $t = 2p_s$, ..., $t = ip_s$, respectively. We assume that we apply most recent message first scheduling scheme, where $pri(m_0) < pri(m_1) < ... < pri(m_i)$. The delay of a message without conflicts with others is $2nl$. A message can conflict with other messages with higher priority when $2nl > p_s$. Three cases are discussed below: (1) $\lfloor \frac{p_s}{l} \rfloor \leq 2$, (2) $3 \leq \lfloor \frac{p_s}{l} \rfloor \leq 4$ and (3) $\lfloor \frac{p_s}{l} \rfloor \geq 5$.

Lemma VI.1. *When $\lfloor \frac{p_s}{l} \rfloor \leq 2$, no message can be delivered to the destination.*

Proof. For the base case of m_0 and m_1, when both m_0 and m_1 go up, their levels are, respectively, $h(m_0) = \lfloor \frac{t}{l} \rfloor$ and $h(m_1) = \lfloor \frac{t-p_s}{l} \rfloor$ and $\Delta h(m_0, m_1) = \lfloor \frac{p_s}{l} \rfloor \leq 2$. Conflict situation 2 happens, since m_0 and m_1 are separated by less than 3 levels. Let's consider the case that m_0 is sent at time $t = 0$ from level 0. At time $t = p_s, h(m_0) = \lfloor \frac{p_s}{l} \rfloor$, and m_1 is sent out and the TDMA scheduler will prevent m_i from being transmitted until m_1 is at level $n(m_1) = \lfloor \frac{p_s}{l} \rfloor + 3$ at time $t = p_s + 3l$. However, at time $t = 2p_s < p_s + 3l$ (before the conflict of m_0 and m_1 is resolved), m_2 will start transmission and also block m_1. Since the conflict of m_0 and m_1 cannot be resolved, m_0 will never move past level $\lfloor \frac{p_s}{l} \rfloor$.

In general, the situation is similar, where after ip_s ($i = 0, 1, ...$), m_{i+1} will interrupt m_i creating a chain reaction. Therefore, all messages will be blocked by messages with higher priority and no message can be delivered to the destination. Since all messages start by going up, we do not need to consider conflicts situations (a) and (c) because they will never occur. □

Lemma VI.2. *When $3 \leq \lfloor \frac{p_s}{l} \rfloor \leq 4$, no message can be delivered to the destination.*

Proof. Let us first consider the best case (largest separation of two consecutive messages): $\lfloor \frac{p_s}{l} \rfloor = 4$.

For the base case, when both m_0 and m_1 go up ($\lfloor \frac{t}{l} \rfloor < n$), $\Delta h(m_0, m_1) = \lfloor \frac{p_s}{l} \rfloor \geq 3$ with no conflict. When m_0 goes down ($\lfloor \frac{t}{l} \rfloor > n$) and m_1 goes up ($\lfloor \frac{t-p_s}{l} \rfloor < n$), $\Delta h(m_0, m_1) = 2n - \lfloor \frac{t}{l} \rfloor - \lfloor \frac{t-p_s}{l} \rfloor \leq 2n - 2\lfloor \frac{t}{l} \rfloor + \lfloor \frac{p_s}{l} \rfloor \leq \lfloor \frac{p_s}{l} \rfloor = 4$. Let us consider the best case (largest separation of m_0 and m_1) with $\Delta h(m_0, m_1) = 4$. As shown in Figure 6a, the conflict happens when $h(m_0) = n - 1$ on the way down (red arrow represents m_0) and $h(m_1) = n - 3$ on the way up (black arrow represents m_1). As shown in Figure 6b, the conflict involves conflict situations 1 and 3: (1) during the first conflict, m_0 waits m_1 going up to the remote controller; (2) when m_1 reaches remote controller, the conflict becomes conflict situation 3 and is resolved when m_1 reaches level $n-3$. So the conflict is resolved with $7l$ time slots if m_2 and the following messages do not exist. However, after $5l$ slots of the conflict of m_0 and m_1, where m_1 is on the way down at level $n-1$, m_1 will conflict with m_2 and the previous conflict of m_0 and m_1 will never be resolved. m_0 will be blocked at level $n-1$ forever.

For general case of m_i and m_{i+1}, when m_i goes down, $h(m_i) = 2n - \lfloor \frac{t-ip_s}{l} \rfloor$ ($\lfloor \frac{t-ip_s}{l} \rfloor > n$); and when m_{i+1} goes up, $h(m_{i+1}) = \lfloor \frac{t-(i+1)p_s}{l} \rfloor$ ($\lfloor \frac{t-(i+1)p_s}{l} \rfloor < n$). Since $\Delta h(m_i, m_{i+1}) \leq 2n - 2\lfloor \frac{t-p_s}{l} \rfloor + \lfloor \frac{p_s}{l} \rfloor \leq 4$, with the largest level separation of 4, m_i will conflict with m_{i+1} as the same situation as the base case above. After $5l$ of the conflict of m_i

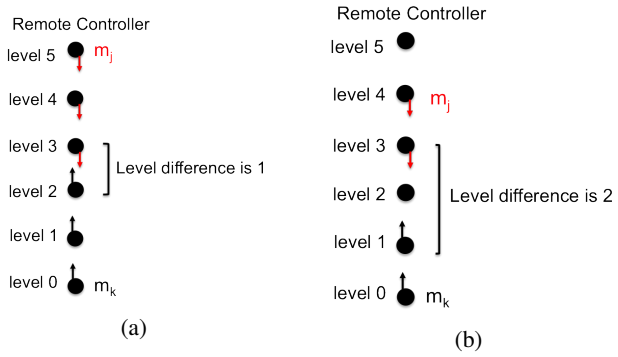

Fig. 7: The conflict of m_j when the level separation with m_k is 5 (a) and 4 (b)

and m_{i+1} (the conflict takes $7l$ to resolve), m_{i+1} conflicts with m_{i+2}, and the conflict of m_i and m_{i+1} cannot be resolved. Therefore, all the messages will be blocked by higher priority messages at level $n-1$ with $\lfloor \frac{p_s}{l} \rfloor = 4$.

Clearly, if the best case of $\lfloor \frac{p_s}{l} \rfloor = 4$ causes indefinite blocking, the case of $\lfloor \frac{p_s}{l} \rfloor = 3$ will come to the same conclusion.

□

Lemma VI.3. *When $\lfloor \frac{p_s}{l} \rfloor \geq 5$, messages will be delivered to the destination.*

Proof. We prove this Lemma by showing it is true for the worst case (smallest separation of two consecutive messages) when $\frac{p_s}{l}$ is odd, that is, $\frac{p_s}{l} = 5$. The worst case for when $\frac{p_s}{l}$ is even is shown in Lemma A.1 in the Appendix. We show the Lemma is true for the base case of m_0 and m_1, and then generalize to any consecutive messages, m_i and m_{i+1}. There are three cases:

(1) When both m_0 and m_1 go up ($\lfloor \frac{p_s}{l} \rfloor < n$), $\Delta h(m_0, m_1) = \lfloor \frac{p_s}{l} \rfloor - \lfloor \frac{t-p_s}{l} \rfloor = \frac{p_s}{l} = 5$, there is no conflict. The proof is similar to the proof of Lemma VI.1, where we showed that if messages are going up one of them is blocked if they are separated by 2 or fewer levels.

(2) When m_0 goes down ($\lfloor \frac{t}{l} \rfloor > n$) and m_1 goes up ($\lfloor \frac{t-p}{l} \rfloor < n$), $\Delta h(m_0, m_1) = 2n - \lfloor \frac{t}{l} \rfloor - \lfloor \frac{t-p_s}{l} \rfloor$. The conflict only involves the conflict situation 1. Since we are dealing with the case of $\frac{p_s}{l} = 5$, which is odd, the conflict happens with $\Delta h(m_0, m_1) = 1$ and can be resolved with $2l$ time slots. After this conflict, since the level difference is $\Delta h(m_0, m_1) = \left\lfloor \frac{p_s - 2l}{l} \right\rfloor = 3$, there is no more conflict between m_0 and m_1. Figure 8 shows $\Delta h(m_0, m_1) = 5$ to start with (before conflict), going down to 3, after the conflict (because m_1 advances 2 levels while m_0 stalls).

(3) When both m_0 and m_1 go down, both m_0 and m_1 will conflict with higher priority messages, m_2, m_3, ... m_j. These conflicts involve the conflict situation 1, given that m_2, m_3, ..., m_j go up. For both m_0 and m_1, only the first conflict starts out with an odd level separation (for m_0 see case (2) above) and the rest of conflicts are all even. Therefore, as shown in Figure 8, conflicts after the first conflict are resolved $3l$ time

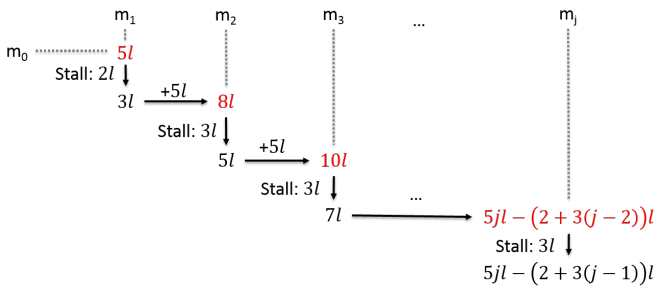

Fig. 8: The calculation process of level separations with higher priority messages for m_0 and m_1, when $\frac{p_s}{l} = 5$

TABLE I: The total stall caused by conflicts when m_0 and m_1 conflict with higher priority messages

	m_1	m_2	m_3	...	m_j
m_0	$2l$	$(2+3)l$	$(2+2*3)l$		$2l+3(j-1)l$
m_1	-	$2l$	$(2+3)l$		$2l+3(j-2)l$

slots. A similar process can be followed for m_1. Table I shows the total stall in terms of the number of time slots when m_0 and m_1 conflict with m_2, m_3, ..., m_j starting with $\frac{p_s}{l} = 5$.

In addition to situation a, we must consider also situation c, given that when both m_0 and m_1 go down and m_0 is ahead of m_1, m_0 will stall first, causing m_1 to approach m_0, further causing situation c conflicts. Below, we separate this into three subcases to show how these conflicts are resolved: (3A) m_0 and m_1 conflicting with m_2, (3B) m_0 and m_1 conflicting with m_3 and m_0 and (3C) m_1 conflicting with m_j.

Case 3A: m_0 and m_1 conflict with m_2. During the conflict of m_0 with m_2, m_1 will go down 2 levels, and during the conflict of m_1 with m_2, m_0 will go down 1 level, as follows. When m_0 starts conflicting with m_2 at time slot $t_c(m_0) = nl + p_s$, $\Delta h(m_0, m_2) = 2n - \lfloor \frac{t-2l}{l} \rfloor - \lfloor \frac{t-2p_s}{l} \rfloor = 2$, and we get $\lfloor \frac{t}{l} \rfloor = h + \lfloor \frac{p_s}{l} \rfloor$, so $h(m_0) = n - \frac{p_s}{l} + 2$. When m_1 starts conflicting with m_2 at time slot $t_c(m_1)$, $\Delta h(m_1, m_2) = 2n - \lfloor \frac{t-p_s}{l} \rfloor - \lfloor \frac{t-2p_s}{l} \rfloor = 1$, and we get $\lfloor \frac{t-p_s}{l} \rfloor = h + \frac{1}{2}\frac{p}{l} - \frac{1}{2}$, so $h(m_1) = n - \frac{1}{2}\frac{p_s}{l} + \frac{1}{2}$. Given that $\Delta h(m_1, m_0) = \frac{1}{2}\frac{p_s}{l} - \frac{3}{2} = 1$, m_0 and m_1 will conflict again with each other (this time under conflict situation 3).

To explain how long m_0 got stalled before m_1 starts its conflict with m_2, we turn to Figure 9, which shows the stall time for m_0 from I_0 to I_2 and m_1 from I_2 to I_3. The length of I_0, I_1, I_2 and I_3 is l, the time to transmit the message for one level. Since m_0 stalls for $3l$ and it can be shown that $t_c(m_1) - t_c(m_0) = \frac{1}{2}p_s - \frac{1}{2}l = 2l$ (shortest/worst case), the overlap of m_0 and m_2 is l, during I_2. During I_0 to I_1, m_0 conflicts with m_2 (and stalls), while m_1 keeps going down 2 levels and m_2 goes up 2 levels. During I_2, both m_0 and m_1 conflict with m_2 and only m_2 (highest priority) goes up 1 level. During I_3, m_1 conflicts with m_2, allowing m_0 to make progress and go down 1 level and m_2 to go up 1 level. Note that m_0 and m_1 will not conflict with m_3, since the time duration for the conflict among m_0, m_1 and m_2 is $4l < p_s = 5l$, and m_3 has not come to level x yet (see Figure 9).

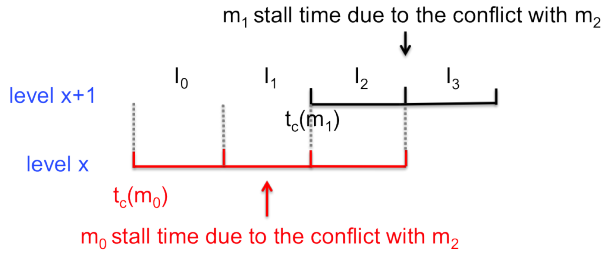

Fig. 9: The stall time for m_0 (lower red segments) and m_1 (upper black segments), when conflicting with m_2

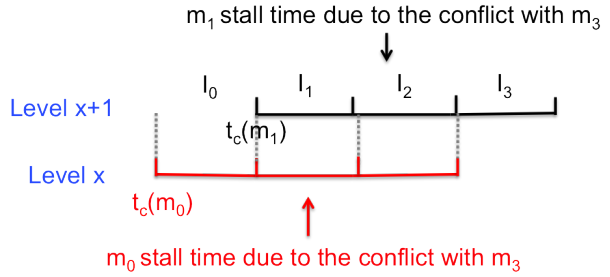

Fig. 10: The stall time for m_0 (lower red segments) and m_1 (upper black segments), when conflicting with m_3

Case 3B: m_0 and m_1 conflict with m_3. m_0 and m_1 will not be blocked during the conflicts with m_3: m_0 will go down for 1 level, and m_1 will go down for 1 level. When m_0 starts conflicting with m_3, $\Delta h(m_0, m_3) = 2n - \left\lfloor \frac{t-5l}{l} \right\rfloor - \left\lfloor \frac{t-3p_s}{l} \right\rfloor = 2$, $h(m_0) = n - \frac{3}{2}\frac{p_s}{l} + \frac{7}{2}$, $t_c(m_0) = nl + \frac{3}{2}p_s + \frac{3}{2}l$. When m_1 starts conflicting with m_3, $\Delta h(m_1, m_3) = 2n - \left\lfloor \frac{t-p_s-2l}{l} \right\rfloor - \left\lfloor \frac{t-3p_s}{l} \right\rfloor = 2$, $h(m_1) = n - \frac{p_s}{l} + 2$, $t_c(m_1) = nl + 2p_s$. Thus $\Delta h(m_1, m_0) = \frac{1}{2}\frac{p_s}{l} - \frac{3}{2} = 1$. The start conflicting time difference is $t(m_1) - t(m_0) = \frac{1}{2}p_s - \frac{3}{2}l = l$. Figure 10 illustrates stall intervals for m_0 conflicting with m_1 and m_3. During I_0, m_0 conflicts with m_1 and m_3, allowing both m_1 and m_3 to go down and up for 1 level, respectively. During I_1 to I_2, m_0 conflicts with m_1 and m_3, and m_1 conflicts with m_0 and m_3, allowing only m_3 to go up for 2 levels. During I_3, m_1 conflicts with m_0 and m_3, allowing m_0 to go down for 1 level and m_3 to go up for 1 level. Even though m_0 and m_1 conflict, they can still move further by 1 level. Similarly, the duration of the conflict is $4l < p_s = 5l$, and thus m_4 has not yet come to conflict with m_0, m_1 and m_3.

Case 3C: m_0 and m_1 conflict with m_j ($j \geq 3$). m_0 and m_1 will not be blocked during the conflict and can both go down by 1 level. In general, when m_0 starts conflicting with m_j, $\Delta h(m_0, m_j) = 2n - \left\lfloor \frac{t-(2+3(j-2)l)}{l} \right\rfloor - \left\lfloor \frac{t-jp_s}{l} \right\rfloor = 2$, $h(m_0) = n - \frac{j}{2}\frac{p_s}{l} + \frac{3}{2}(j-2) + 2$, $t(m_0) = nl + \frac{3}{2}(j-2)l + \frac{j}{2}p_s$. When m_1 starts conflicting with m_j, $\Delta h(m_1, m_j) = 2n - \left\lfloor \frac{t-p_s-(2+3(j-3))l}{l} \right\rfloor - \left\lfloor \frac{t-jp}{l} \right\rfloor = 2$, $h(m_1) = n - \frac{1}{2}(j-1)\frac{p_s}{l} + 2 + \frac{3}{2}(j-3)$, $t(m_1) = nl + \frac{3}{2}(j-3)l + \frac{j}{2}p_s + \frac{1}{2}p_s$. Thus, $\Delta h(m_1, m_0) = \frac{1}{2}\frac{p_s}{l} - \frac{3}{2} = 1$. The start conflict time difference is $t(m_1) - t(m_0) = \frac{1}{2}p_s - \frac{3}{2}l = l$. The stall time for both m_0 and m_j is the same as Figure 10: during the conflict, m_1 can go down for 1 level during I_0; and m_0 can go down for 1 level during I_3. This pattern will repeat itself indefinitely in the worst case.

For any two consecutive messages, m_i and m_{i+1}, we can show the message progress, similar to the process above. Conflicts always happen when the lower priority messages are going down (situation a). Even though two messages going down conflict with each other, each gets a chance to make progress when the other one is stalling due to conflicts with higher priority messages; both messages finally can reach to the destination.

As mentioned above, this proves the worst case for odd separation, while the even separation can be found in the appendix. Outside the worst case, the message density is lower, and therefore fewer conflicts and stalls will happen. □

Even though we set the fixed priority order as $pri(m_0) < pri(m_1) < ... < pri(m_i)$ to prove the three lemmas above, the reasoning is still valid if we reverse the priority order, the process becomes symmetric: all conflicts will happen while a lower priority message is traveling up.

B. Worst-case end-to-end delay determination

Based on Lemmas VI.1, VI.2 and VI.3, in this section we assume that $\left\lfloor \frac{p_s}{l} \right\rfloor \geq 5$. Assume that a message already conflicted with $(Q-1)$ higher priority messages and the total stall upperbound is $3l(Q-1)$ (given that each conflict can be resolved in at most $3l$ slots for conflict situation 1). The following formula shows the difference in levels between m_0 and m_Q; if that value is 1 or 2, the Q^{th} conflict will happen:

$$1 \leq 2n - \left\lfloor \frac{t - 3(Q-1)l}{l} \right\rfloor - \left\lfloor \frac{t - Qp}{l} \right\rfloor \leq 2 \qquad (4)$$

After algebraic manipulations, we get:

$$\left\lfloor \frac{x}{l} \right\rfloor = n + \frac{3}{2}(Q-1) + \frac{1}{2}\left\lfloor \frac{Qp}{l} \right\rfloor - 1 \qquad (5)$$

Since the conflicts happen only when a message is transmitted down, the following condition holds about the level of message m_0: $n+1 \leq \left\lfloor \frac{t}{l} \right\rfloor \leq 2n + 3(Q-1) - 1$. From that, we get $\frac{7l}{2l+p_s} \leq Q \leq \frac{2nl-3l}{p_s-3l}$ and derive the maximum Q as $\left\lfloor \frac{2nl-3l}{p_s-3l} \right\rfloor$.

After calculating the maximum number of conflicts, we can estimate the worst-case stall caused by conflicts, $D_{conflict} = 3lQ = 3l\left\lfloor \frac{2nl-3l}{p_s-3l} \right\rfloor$. The delay without conflicts for transmitting one message up to the remote controller is nl and the same amount of delay for going down. So, the delay without conflict, $D_{pure} = 2nl$. The worst-case end-to-end delay is

$$D = D_{pure} + D_{conflict} = 2nl + 3l\left\lfloor \frac{2nl - 3l}{p_s - 3l} \right\rfloor \qquad (6)$$

To determine the worst-case end-to-end delay, we multiply D by Δt, and obtain $D_{network}$, which will be used in the following sections to, for each network path: (1) design a

network that guarantees meeting the control system deadline; (2) assign the highest priority measurement packets to the path with shortest worst-case delay.

VII. Network Path Quality Model

Recall that in Section IV we determined the priority of the measurement packets and in Section VI we calculated the worst-case delay when considering a network without packet losses. Now we need to determine which network path to transmit those measurements when considering packet losses (i.e., $dr(T) < 1$). Although previous research discussed how the network reliability and network delay affect the control system performance[18], [8], to the best of our knowledge there is still no model that builds the relationship between network performance (i.e. network delay and message loss) and control system performance (i.e. RMSE). We propose a general network quality model, the *PQmodel*, which includes both network delay and losses, as described by Equation 7, that quantifies how much the network affects the control system.

$$PQ = D_{network} + \alpha \ n_{loss}p \tag{7}$$

where $D_{network}$ is the network end-to-end delay, p is the control sampling period, α is a constant, and n_{loss} is the number of consecutive packet losses. Note that n_{loss} is computed from the control system perspective, that is, if a message is received by the controller every control sampling period n_{loss} = 0. Note that this PQmodel quantifies the network imperfection impact to the control system, thus a smaller PQ value means better quality the network path.

We use α to adjust the importance between network delay and network reliability. When $\alpha = 1$, network delay and network reliability have the same importance to the control system performance.

α is set according to different control systems we are dealing with. When the network delay is smaller than the control system sampling period (e.g. like the water tank system in [8]), α is set to a very large number since network reliability is the only factor that affects the control system performance. When the control sampling period is smaller than the network delay, a more common scenario, α is a number closer to 1. For instance, when the control system uses kalman filter or any other technique to compensate for message loss, we can reduce the network reliability importance and set α to be small. α also needs to be adjusted under different network situations for the same control system. We will discuss the value of α under different network situations later in Section IX.

VIII. Nuclear Power Plan Case Study

A modern NPP design considers several Small Modular Reactors (SMRs) [3], instead of a single large reactor, due to the flexibility and cost-benefit of starting and stopping SMRs. Typically there is one primary heat exchanger (PHX) in each SMR, as well as two secondary ones. The PHX is typically modeled as a nonlinear system. For each PHX, we focus on measurements that are sent periodically to the controller, namely outlet hot leg temperature, inlet hot leg temperature, and mass flow rate.

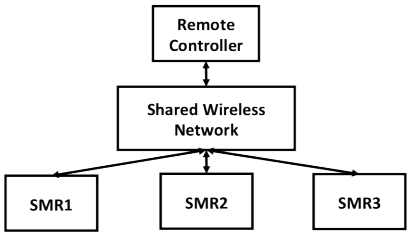

Fig. 11: System overview: three SMRs transmit measurements via shared wireless network to the remote controller

As shown in Figure 11, we conduct a case study of a NPP with three SMRs (three PHXs and six secondary heat exchangers, each of which transmits measurement data via a shared wireless network (we focus on the measurements sent periodically). Given that there are several SMRs in an NPP, the power output of each SMR may differ and the controller may decide to change the power output of each SMR dynamically, based on energy requirements and balance the power required to achieve a certain level of power output. The PHXs in SMRs are identical systems except for the reference functions, which are set by the nuclear engineer/operator based on the NPP requirement. To be general in our case study, a reference function is a ramp function, defined by: (1) power change amount (PCA) as the amount of power required to change; (2) power change duration (PCD) as the interval of time the power finishes changing; (3) start time (ST) as the time duration from time 0 to the time the power starts to change. For example, ramp30 in Figure 1 is with PCA=10MW, PCD=30s and ST=40s. The parameters in *a set of reference functions* are 3 PCAs, 3 PCDs and 3 STs to set three reference functions. Each reference function is randomly chosen by uniform distribution from the range of values of PCA, PCD and ST listed in Table II.

In order to include all the PCDs, we choose simulation time as 300s, taking into account the system settling time (even after the PCD, the system still needs sometime to settle down to the setpoint). Each PHX will generate one measurement packet (include its three measurements) and send out the packet by wireless network periodically at the sampling period 0.2s. If the measurement packet is lost during the wireless transmission, the control system uses the latest received measurement value/control signal.

For the wireless network, we use the bitvector protocol [19], which uses TDMA scheduling to guarantee real-time transmission with time slot $\Delta t = 0.01s$. Based on the deadline of one PHX system (0.586s [18]) and the end-to-end delay analysis discussed in Section VI, we design a wireless network with three paths, each of 6 hops: path 1 (p_1) has no backups (worst-case delay: 0.12s); path 2 (p_2) has two lines of relay nodes (worst-case delay: 0.3s); path 3 (p_3) has 3 lines of relay nodes (worst-case delay: 0.54s). The reliability relationship of the three paths is $dr(p_1) < dr(p_2) < dr(p_3)$. Each network

path can transmit messages independently from the others, that is, all 3 paths can transmit messages in parallel, without interfering with each other. We apply the message priority scheme as above: most recent message first.

TABLE II: Parameters and values of simulation of SMR-based NPP

Parameters	Values
Network sampling period	0.2s
Control sampling period	0.2s
Simulation time	300s
PCA	2MW, 4MW, 6MW, 8MW, 10MW
PCD	15s, 30s, 45s, 60, 75s, 90s, 105s, 120s
ST	range: [20s, 300s-PCD]
α value	range: [0.0 2.0]

We combined a state-of-the-art cyber-physical system simulator (WCPS 2.0 [8]) with a NPP simulator to mimic the WCS we consider. Our simulator allows k wireless network paths running together with k PHXs. We implement the heuristic method proposed in Section IV to assign priority to the measurement packets and the network quality model from Section VII to quantify the quality of network paths.

We use the TOSSIM network simulator (embedded in WCPS) with wireless traces from a 21-node subset of the WUSTL Testbed [4]. To evaluate the wireless control systems under a wide range of wireless conditions (e.g. different levels of noise/interference), similar to [18], we use controlled Received Signal Strength with uniform gaps to simulate various wireless signal strength (RSSI) values to change the quality of network links. As in [8], we adjust the RSSI values for the average link success ratio (LSR) to be in the range (0.71, 1.0).

IX. Quantitative Results from Case Study

Based on the wireless control system for the NPP introduced above, we first evaluate the worst-case end-to-end network delay analysis with the realistic simulation results. Second, we compare the reliability of the three network paths for different network conditions. Third, we evaluate our network path quality model. More specifically, we did sensitivity analysis of α values for different network conditions and analyze the network path selection for different network conditions. Finally, we compare $RMSE_{total}$ for both end-to-end delay approach and PQmodel approach.

A. End-to-end network delay validation

To validate our worst-case end-to-end delay analysis, we ran the simulation with different values of p, l and n, as shown in Table III. We get 100% accuracy for the feasibility of the network settings that can deliver the messages to the destination within the respective deadlines. For the feasible network settings, that is, $\lfloor \frac{p_s}{l} \rfloor \geq 5$, the worst-case delay analysis overestimates the delay in 1.866% compared with the realistic simulation results (safe and tight results). Figure 12 shows one example of message transmission process with $p = 0.1s$, $p_s = 10$, $l = 2$ and $n = 10$. As discussed in Section VI, the lower priority messages conflict with higher priority messages and are delayed when traveling "down". Regardless, they still arrive at the controller within the deadlines, because they satisfy the condition $\lfloor \frac{p_s}{l} \rfloor \geq 5$.

TABLE III: Simulation parameters and values

parameters	values
p	0.05s, 0.1s, 0.15s, 0.2s, 0.25s, 0.3s
p_s	5, 10, 15, 20, 25, 30
l	1, 2, 3, 4
n	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

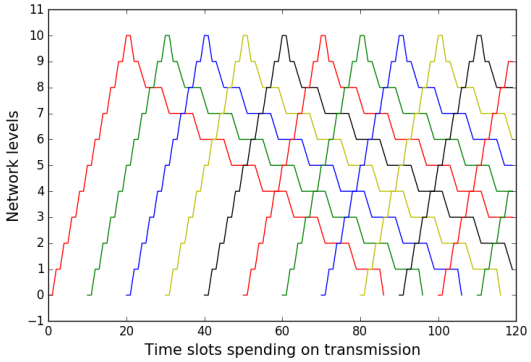

Fig. 12: One example of message transmission process with $p = 0.1s$, $p_s = 10$, $l = 2$ and $n = 10$

B. Network reliability results

Figure 13 shows the delivery ratio of three network paths under different RSSI values. The delivery ratio increases as the number of backup paths increases. Since p_1 has no backup path, the delivery ratio is about 0.6 when RSSI value is -64. At the other extreme, the delivery ratio of p_3 (two backup paths) is above 0.8 with the RSSI value -84 (poor network conditions).

C. Control system results of PQmodel approach

1) Sensitivity analysis of α value for network path quality: To evaluate the network quality model proposed in Section VII, we experiment with different α values from 0.1 to 2.0 for the heuristic method proposed in Section IV over different RSSI values on 20 sets of reference functions. Each experiment runs 20 times on the network paths given the RSSI value. Figure 14 shows the value of α that minimizes $RMSE_{total}$. The value of best α increases first, then decreases

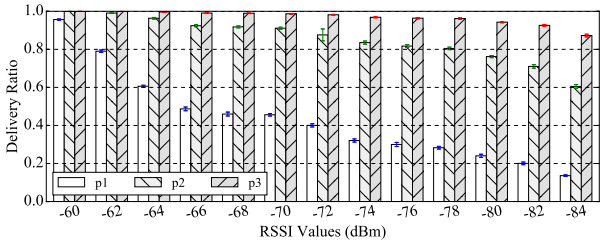

Fig. 13: Delivery ratio of three network paths under different RSSI values

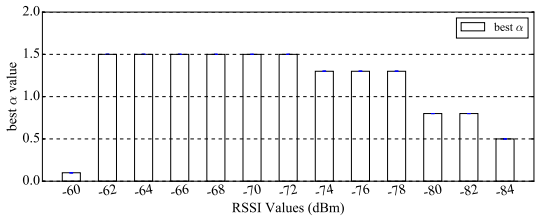

Fig. 14

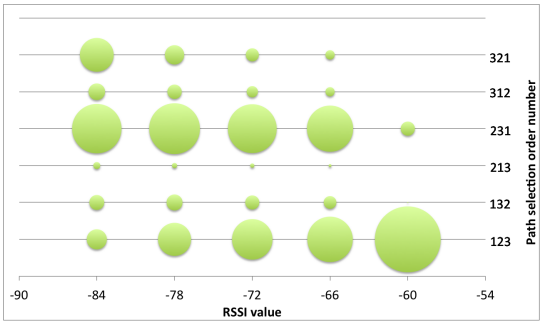

Fig. 15: The path quality order number of path selection for different RSSI values

as the interference in the network increases. It is because that when the network has less interference (RSSI=-60 dBm), the network is very reliable and the n_{loss} is less important than the network delay. When the network has a lot interference (RSSI=-84dBm), all paths loose many messages and no path is reliable, so n_{loss} is also not as important as delay.

2) Path quality order selection: Figure 15 shows the path quality order (123 means highest priority goes to path p_1) when the best α is applied in the network quality model for each RSSI values (we only show parts of the RSSI values for ease of the presentation; the trend can be easily seen from the figure). The size of the bubble shows the average amount of time the model chooses that path quality order. As the RSSI value decreases, the number of path quality order 123 decreases and path quality order 231 increases, since p_1 has more packet losses and the quality of p_1 decreases. Quality order 321 and 312 are not high, since p_3 has the highest network delay and it will only be selected when the other two paths have too many message losses. Moreover, quality order 132 is also small, because when p_1 has the highest quality, it implies the network condition is good and both p_2 and p_3 have high reliability; since p_3 has more network delay than p_2, the chance that the quality of p_3 is higher than p_2 is low.

D. End-to-end delay approach and PQmodel approach comparison

We evaluate the $RMSE_{total}$ (defined in Equation 2) for end-to-end delay approach and PQmodel approach over 100 different sets of the reference functions of three PHXs. For each set of reference functions, we run 20 times on the three wireless network paths for each RSSI value. The average $RMSE_{total}$ is shown in Figure 16. The PQmodel performs

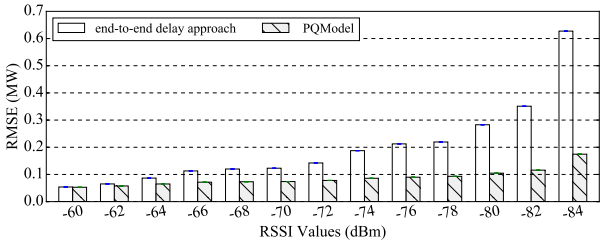

Fig. 16: $RMSE_{total}$ comparison of end-to-end delay approach and PQmodel over different network conditions

better than only considering end-to-end delay in all network conditions by 2% (RSSI=-60 dBm) to 259% (RSSI=-84 dBm). The more interference in the network, the more improvement we can get from the PQmodel because the PQmodel appropriately characterizes the relationship between network delay and message loss under different network conditions, and thus can more effectively assign the priority of network path when network. The result demonstrates that both network delay and packet loss are key factors for the overall control system performance. Our two-step approach (heuristic method + PQmodel) is effective with low $RMSE_{total}$.

X. Conclusion and Future Work

We explore the interaction between dynamic packet scheduling and the control system performance in a WCS with one shared wireless network and multiple physical systems. Motivated by the observation that network delay and packet loss have different effects on control system performance depending on the system application demand, we propose a dynamic packet scheduling solution with the goal of minimizing RMS error caused by network imperfections. Specifically, our solution has two steps: measurement packet priority assignment and network path quality determination taking account only the network delay by worst-case end-to-end delay analysis first, then considering message loss through a heuristic. From the end-to-end delay analysis, we get the schedulability condition ($\lfloor \frac{p_s}{l} \rfloor \geq 5$). To evaluate our solution, we carried out a case study on three SMRs in nuclear power plant with one shared wireless network. First, our end-to-end delay analysis is accurate within 1.866% of a realistic simulation results (always more pessimistic, but a very tight pessimism). Second, our proposed PQmodel performs better than only considering network delay between 2% and 259% (for really bad network conditions), which demonstrates that both network delay and reliability play an important role in control system performance. The results also show that our two-step solution is effective in lowering the total power output error of the nuclear power plant.

As future work, we are going to apply our approach to other WCSs as another case study to ensure our conclusions still hold.

APPENDIX

Lemma A.1. *The worst-case of even values of lemma VI.3:* $\frac{p_s}{l} = 6$*, messages will be delivered to the destination*

Proof. There are three cases:

(1) When both m_0 and m_1 go up ($\lfloor \frac{p_s}{l} \rfloor < n$), $\Delta h(m_0, m_1) = \lfloor \frac{p_s}{l} \rfloor - \lfloor \frac{t-p_s}{l} \rfloor = \frac{p_s}{l} = 6$, there is no conflict. The proof is similar to the proof of Lemma VI.1.

(2) When m_0 goes down ($\lfloor \frac{t}{l} \rfloor > n$) and m_1 goes up ($\lfloor \frac{t-p}{l} \rfloor < n$), $\Delta h(m_0, m_1) = 2n - \lfloor \frac{t}{l} \rfloor - \lfloor \frac{t-p_s}{l} \rfloor$. The conflict only involves the conflict situation 1. Since $\frac{p_s}{l} = 6$ is even, the conflict happens with $\Delta h(m_0, m_1) = 2$ and can be resolved with $3l$ time slots. After the first conflict, $h(m_1) = \lfloor \frac{t-p_s}{l} \rfloor + 3 = n - \frac{1}{2}\frac{p_s}{l} + 2 = n - 1$. Since the level difference is $\Delta h(m_1, m_0) = \left\lfloor \frac{p_s - 3l}{l} \right\rfloor = 3$ and m_0 and m_1 go two opposite directions, there is no more conflict between m_0 and m_1.

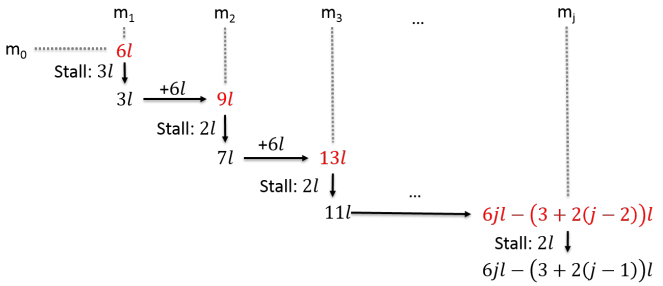

Fig. 17: The calculation process of level separations with higher priority messages for m_0 and m_1, when $\frac{p_s}{l} = 6$

(3) When both m_0 and m_1 go down, m_0 and m_1 could conflict with higher priority messages, m_2, m_3, ... m_j. These conflicts only involve the conflict situation 1, when m_2, m_3, ..., m_j go up. The level separation of m_0 just before conflict with m_1, m_2, ..., m_3 is shown as red in the upper side of Figure 8. The level separation of m_1 before conflict with higher priority messages is shown as blue in the lower side of Figure 17. For both m_0 and m_1, only the first conflict has even number of level separation, and the rest of conflicts are all odd. Therefore, only the first conflict happens when the level difference is 2 and are resolved by $3l$ time slots. The rest of conflicts happen when the level difference is 1 and are resolved by $2l$ time slots. Table IV shows the total stall in terms of the number of time slots when m_0 and m_1 conflict with m_2, m_3, ..., m_j with $\frac{p_s}{l} = 6$. Below we illustrate specifically three cases of m_0 and m_1 conflicting with m_2, m_0 and m_1 conflicting with m_3 and m_0 and m_1 conflicting with m_j. We then generalize to any two consecutive messages.

TABLE IV: The total stall in terms of the number of time slots when m_0 and m_1 conflict with higher priority messages

	m_1	m_2	m_3	...	m_j
m_0	$3l$	$(3+2)l$	$(3+2*2)l$		$3l+2(j-1)l$
m_1	-	$3l$	$(3+2)l$		$3l+2(j-2)l$

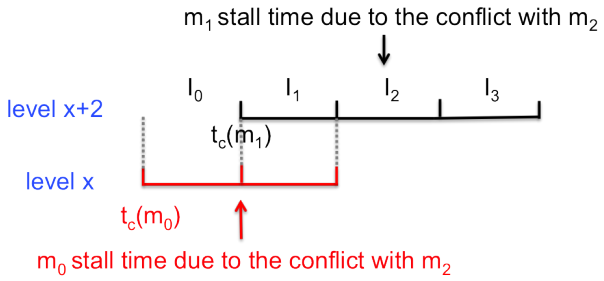

Fig. 18: The stall time for m_0 (lower red segments) and m_1 (upper black segments), when conflicting with m_2

1) m_0 and m_1 conflict with m_2. m_0 and m_1 will not be blocked during the conflicts with m_2: m_0 will go down for 2 levels, and m_1 will go down for 1 level. When m_0 starts conflicting with m_2, $\Delta h(m_0, m_2) = 2n - \lfloor \frac{t-3l}{l} \rfloor - \lfloor \frac{t-2p_s}{l} \rfloor = 1$, $h(m_0) = n - \frac{p_s}{l} + 2$, $t(m_0) = hl + p_s + l$ (the time slot number that m_0 starts the conflict with m_2). When m_1 starts conflicting with m_2, $\Delta h(m_1, m_2) = 2n - \lfloor \frac{t-p_s}{l} \rfloor - \lfloor \frac{t-2p_s}{l} \rfloor = 2$, $h(m_1) = n - \frac{1}{2}\frac{p_s}{l} + 1$, $t(m_1) = nl + \frac{3}{2}p - l$ (the time slot number that m_1 starts the conflict with m_2). $\Delta h(m_1, m_0) = \frac{1}{2}\frac{p_s}{l} - 1 = 2$, which means m_0 and m_1 will conflict again (conflict situation 3) with each other given that m_0 got stalled before m_1 conflicts with m_2. $t(m_1) - t(m_0) = \frac{1}{2}p_s - 2l = l$. Figure 18 represents the stall time for m_0 from I_0 to I_1 and m_1 from I_1 to I_3. During I_0, m_0 conflicts with m_2 (and stalls), while m_1 keeps making progress and goes down for 1 level and m_2 goes up for 1 level. During I_1, m_0 conflicts with both m_1 and m_2; m_1 conflicts with m_2; only m_2 (highest priority) goes up 1 level. During I_2 to I_3, m_1 conflicts with m_2, allowing m_0 to make progress and go down 2 levels and m_2 to go up 2 levels. Note that m_0 and m_1 will not conflict with m_3, since the time duration for the conflict among m_0, m_1 and m_2 is $4l < p_s = 5l$, and m_3 has not come to level x yet (see Figure 18).

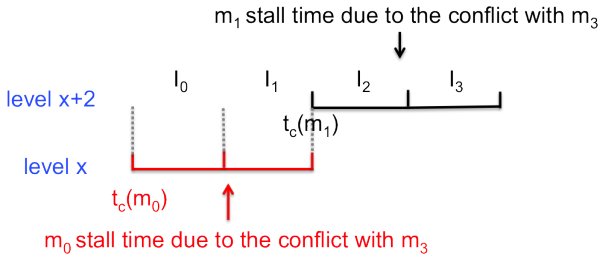

Fig. 19: The stall time for m_0 (lower red segments) and m_1 (upper black segments), when conflicting with m_3

2) m_0 and m_1 conflict with m_3. m_0 and m_1 will not be blocked during the conflicts with m_3: m_0 will go down for 2 levels, and m_1 will go down for 2 levels. When m_0 starts conflicting with m_3, $\Delta h(m_0, m_3) = 2n - \lfloor \frac{t-5l}{l} \rfloor - \lfloor \frac{t-3p_s}{l} \rfloor = 1$, $h(m_0) = n - \frac{3}{2}\frac{p_s}{l} + 3$, $t(m_0) = nl + \frac{3}{2}p_s +$

$2l$. When m_1 starts conflicting with m_3, $\Delta h(m_1, m_3) = 2n - \left\lfloor \frac{t-p_s-3l}{l} \right\rfloor - \left\lfloor \frac{t-3p_s}{l} \right\rfloor = 1$, $h(m_1) = n - \frac{p_s}{l} + 2$, $t(m_1) = nl + 2p_s + l$. The level difference between m_1 and m_0 is $\Delta h(m_1, m_0) = \frac{1}{2}\frac{p_s}{l} - 1 = 2$. The start conflicting time difference is $t(m_1) - t(m_0) = \frac{1}{2}p_s - l = 2l$. As shown in Figure 10, during I_0 to I_1, m_0 conflicts with m_1 and m_3, allowing both m_1 and m_3 to go down and up for 2 levels, respectively. During I_2 to I_3, m_1 conflicts with m_0 and m_3, allowing both m_0 and m_3 to go down and up for 2 levels, respectively. Even though m_0 and m_1 conflict, they can still move further by 2 levels. Similarly, the duration of the conflict is $4l < p_s = 5l$, m_4 has not come to level x yet (see Figure 10) and will not conflict with m_0, m_1 and m_3.

3) m_0 and m_1 conflict with m_j ($j \geq 3$). m_0 and m_1 will not be blocked during the conflict and can go down by 2 levels. In general, when m_0 starts conflicting with m_j, $\Delta h(m_0, m_j) = 2n - \left\lfloor \frac{t-(3+2(j-2)l)}{l} \right\rfloor - \left\lfloor \frac{t-jp_s}{l} \right\rfloor = 1$, $h(m_0) = n - \frac{j}{2}\frac{p_s}{l} + j$, $t(m_0) = nl + \frac{j}{2}p_s + (j-1)l$. When m_1 starts conflicting with m_j, $\Delta h(m_1, m_j) = 2n - \left\lfloor \frac{t-p_s-(3+2(j-3))l}{l} \right\rfloor - \left\lfloor \frac{t-jp}{l} \right\rfloor = 1$, $h(m_1) = n - \frac{1}{2}(j-1)\frac{p_s}{l} + j - 1$, $t(m_1) = nl + \frac{j+1}{2}p_s + (j-2)l$. The level difference between m_1 and m_0 is $\Delta h(m_1, m_0) = \frac{1}{2}\frac{p_s}{l} - 1 = 2$. The start conflict time difference is $t(m_1) - t(m_0) = \frac{1}{2}p_s - l = 2l$. The stall time for both m_0 and m_j is the same as Figure 10: during the conflict, m_1 can go down for 2 levels during I_0 to I_1; and m_0 can go down for 2 levels during I_2 and I_3.

Similar to the general case of $\frac{p_s}{l} = 5$, for any two consecutive messages, m_i and m_{i+1}, even though they conflict with each other during the downside transmission, each gets a chance to make progress and finally reaches to the destination. □

REFERENCES

[1] Wirelesshart specification. http://www.hartcomm2.org., 2007.

[2] K. Gatsis, A. Ribeiro, and G. J. Pappas. Control-aware random access communication. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–9. IEEE, 2016.

[3] S. R. Greene, J. C. Gehin, D. E. Holcomb, J. J. Carbajo, D. Ilas, A. T. Cisneros, V. K. Varma, W. R. Corwin, D. F. Wilson, G. L. Yoder Jr, et al. Pre-conceptual design of a fluoride-salt-cooled small modular advanced high-temperature reactor (smahtr). *Oak Ridge National Laboratory, Oak Ridge, TN, Report No. ORNL/TM-2010/199, Fig*, pages 8–1, 2010.

[4] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon. Reliable and real-time communication in industrial wireless mesh networks. In *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 3–12. IEEE, 2011.

[5] S. Hong, X. S. Hu, T. Gong, and S. Han. On-line data link layer scheduling in wireless networked control systems. In *2015 27th Euromicro Conference on Real-Time Systems*, pages 57–66. IEEE, 2015.

[6] J. Kim, K. Lakshmanan, and R. R. Rajkumar. Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, pages 55–64. IEEE Computer Society, 2012.

[7] B. Li, Y. Ma, T. Westenbroek, C. Wu, H. Gonzalez, and C. Lu. Wireless routing and control: a cyber-physical case study. In *ACM/IEEE International Conference on Cyber-Physical Systems*, 2016.

[8] B. Li, L. Nie, C. Wu, H. Gonzalez, and C. Lu. Incorporating emergency alarms in reliable wireless process control. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, pages 218–227. ACM, 2015.

[9] H. Li, M.-Y. Chow, and Z. Sun. Optimal stabilizing gain selection for networked control systems with time delays and packet losses. *IEEE Transactions on Control Systems Technology*, 17(5):1154–1162, 2009.

[10] A. Onat, T. Naskali, E. Parlakay, and O. Mutluer. Control over imperfect networks: Model-based predictive networked control systems. *IEEE Transactions on Industrial Electronics*, 58(3):905–913, 2011.

[11] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam. Topological conditions for wireless control networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2353–2360. IEEE, 2011.

[12] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam. The wireless control network: A new approach for control over networks. *IEEE Transactions on Automatic Control*, 56(10):2305–2318, 2011.

[13] G. Pin and T. Parisini. Networked predictive control of uncertain constrained nonlinear systems: recursive feasibility and input-to-state stability analysis. *IEEE Transactions on Automatic Control*, 56(1):72–87, 2011.

[14] A. Saifullah, D. Gunatilaka, P. Tiwari, M. Sha, C. Lu, B. Li, C. Wu, and Y. Chen. Schedulability analysis under graph routing in wirelesshart networks. In *Real-Time Systems Symposium, 2015 IEEE*, pages 165–174. IEEE, 2015.

[15] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. Real-time scheduling for wirelesshart networks. In *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*, pages 150–159. IEEE, 2010.

[16] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. End-to-end delay analysis for fixed priority scheduling in wirelesshart networks. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2011 17th IEEE*, pages 13–22. IEEE, 2011.

[17] A. Ulusoy, O. Gurbuz, and A. Onat. Wireless model-based predictive networked control system over cooperative wireless network. *IEEE Transactions on Industrial Informatics*, 7(1):41–51, 2011.

[18] W. Wang, C. D'Angelo, D. Mosse, and D. Cole. Integrating control and fault-tolerant wireless network design for small modular nuclear reactors. In *Information Reuse and Integration (IRI), 2016 IEEE 17th International Conference on*, pages 332–342. IEEE, 2016.

[19] W. Wang, D. Mosse, and D. G. Cole. Bitvector: Fault tolerant aggregation scheme for monitoring in nuclear power plants. In *ICESS 2015*.

[20] W. Wang, D. Mosse, J. G. Pickel, and D. Cole. Work-in-progress: Cross-layer real-time scheduling for wireless control system. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017 IEEE*, pages 149–152. IEEE, 2017.

[21] W. Wang, D. Mosse, J. G. Pickel, and D. Cole. Work-in-progress: Wireless network reconfiguration for control systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017 IEEE*, pages 145–148. IEEE, 2017.

[22] L. Zhang, H. Gao, and O. Kaynak. Network-induced constraints in networked control systems—a survey. *IEEE Transactions on Industrial Informatics*, 9(1):403–416, 2013.

[23] T. Zhang, T. Gong, C. Gu, H. Ji, S. Han, Q. Deng, and X. S. Hu. Distributed dynamic packet scheduling for handling disturbances in real-time wireless networks. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017 IEEE*, pages 261–272. IEEE, 2017.