Dynamic Wireless Network Reconfiguration for Control System: A nuclear reactor case study

Wenchen Wang, Daniel Mosse
Computer Science Department
University of Pittsburgh

Daniel Cole, Jason G. Pickel
Mechanical Eng and Materials Science Department
University of Pittsburgh

***Abstract*—Control systems using sensors and wireless networks are becoming more prevalent, due to its ease of deployment: no wires and longer battery life. However, network delays and packet losses can degrade control system performance, which leads us to find the optimal network configuration to minimize that impact. Another main difficulty of having wireless networks for control systems is caused by interference and noise that produce time-varying fault patterns, which motivates us to do network reconfiguration at run time. We focus on the online wireless network reconfiguration in cyber-physical systems (CPS), with four main contributions: (1) a network imperfection model; (2) six online reconfiguration algorithms for wireless control system; (3) the design and implementation of the first network reconfiguration framework for CPS with offline and online components that consider time-correlated link failures; (4) a case study with 12 hops and up to 50 nodes that controls a nonlinear primary heat exchanger system in a small modular nuclear reactor. Our network imperfection model is accurate (with 0.993 Pearson correlation) and our online reconfiguration algorithms have smaller error and longer network lifetime than state-of-the-art static network configurations.**

I. Introduction

Wireless control systems (WCS) are composed of controllers, sensors and actuators connected via a wireless network. WCSs controlled over multi-hop wireless sensor network has received significant attention in recent years [7], [16], [15], [26], [25], [33], [12]. Given that the control room is usually geographically distant from the sensors and actuators, wireless networks are good for place-and-play deployment due to the lack of wires (electrical or networking). However, wireless network delays to/from the control room and packet losses can induce serious errors in the control system, which is very undesirable. For example, in a nuclear power plant (NPP), there could be the loss of efficiency, wasting megawatts of power. We focus on the two main categories of *network-induced imperfections* [36]: time delays and packet losses.

In practice, the *control sampling period* (i.e., the interval where the control loop makes decisions) in cyber-physical systems (CPS) is 2^n seconds, where $-2 \leq n \leq 9$, that is, from 250 ms to approximately 8 minutes [7]. For the network delay, there are two cases: (1) network worst-case delay is less than the control sampling period; (2) The worst-case delay is more than the control sampling period. For the first case, the *delivery ratio* (DR, defined as the ratio of arrived messages and sent messages) is the key effect on control system performance. The higher the DR, the better the control system performance. To achieve high DR, the network configuration can be set to "as reliable as possible," that is, a high level of redundancy, which requires more nodes, and thus typically induces more delay for messages to be delivered, but all messages still arrive before the end of the control sampling period and has little effect on the control system performance.

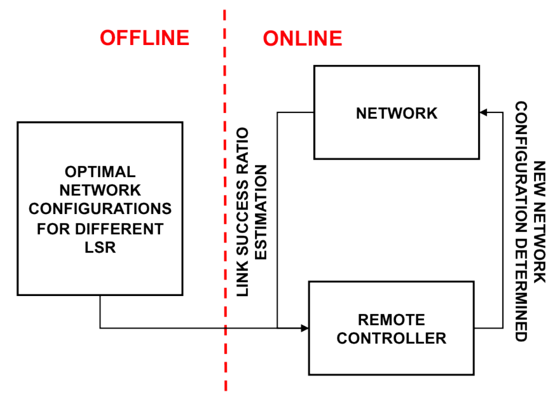

Figure 1: Network reconfiguration for control System with dynamic Interference

However, when the worst-case delay is larger than the control sampling period, there is a trade-off between network delay and packet losses for the control system performance. Although the network-induced imperfection problem has been studied from the control perspective [17], [24], [27], [32], minimizing the network-induced imperfections has not been studied from the network perspective. The trade-off between packet losses (requiring more redundant nodes) and delays (calling for fewer nodes and less redundancy) can be achieved through network configuration.

This problem can be thought of as an optimization to find the network configuration with minimum network imperfections, which can be solved off-line. However, control system environments typically change with time and space [2], [6], such as moving people/obstacles and electromagnetic and radio frequency interference (EMI/RFI). Some interference can make the nodes inaccessible and disconnected for a limited amount of time (e.g., if an obstacle, like a factory robot transporting materials, blocks the wireless transmission), which may severely impact the control system performance. Therefore, we propose an online network reconfiguration to deliver the required control system performance.

In this paper, we focus on multi-hop wireless network reconfiguration for control systems, when the worst-case network delay is bigger than the control sampling period. We only consider link failures and define average *link success ratio* (LSR) as the probability a message can be sent out successfully on that link; we use LSR as the indication of the average network interference. We solve the problem in two parts, offline and online, as shown in Figure 1. For the offline part, to quantify the network imperfection, we propose a network imperfection model to transform network delay and delivery ratio to the *total induced delay* on the control system. We estimate the total induced delay for a set of network configurations, including network topologies. We find an optimal estimated network configuration set for each LSR offline, and store them at the controller node. For the online part, at run time, the network notifies the controller what the estimated LSR is and the controller selects a network configuration for the network, from the ones computed offline. The controller then broadcasts the new network configuration to all the nodes in the network to carry out a reconfiguration. Therefore, the control node acts as a centralized network manager and decides which network configuration should choose. To see the interaction between network reconfiguration and the control system, we conduct a systematic case study with a 12-hop and up to 50-node wireless sensor network for a nonlinear primary heat exchanger system in a small modular nuclear reactor in a NPP. The results show that our network imperfection model is accurate and also that network reconfiguration algorithms performs better than state-of-the-art static network configuration.

The contributions of this paper encompass:

- a network imperfection model that formulates the network imperfection induced into the control system as total induced delay
- six online reconfiguration algorithms for the WCS
- the design and implementation of a new framework with offline and online components for time-correlated link failures
- a case study that presents the in-depth interaction between the network reconfiguration and control

This paper is organized as follows. Section II discusses the related work. Section III introduces how to determine offline optimal network configuration. Section IV discusses online network reconfiguration. Section V and Section VI present case study and results. Section VII concludes the paper.

II. Related Work

Network-induced imperfections are great challenges for WCSs. The solutions for network delay and packet losses in WCSs 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 [17], considering both time delays and packet losses at the actuator nodes. Other examples include [24], [27], [32] 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 do not consider network reconfiguration in WCSs.

For the network solutions, network reconfiguration is an essential part of the network, since the network interference is unpredictable and varies with time. Interference can make the network disconnected and becomes inaccessible for a certain amount of time and will degrade the control system performance. Research works focus on different layers of network reconfiguration. For the data link layer, an online dynamic link layer scheduling algorithm is proposed in [8] to meet the deadline of a rhythmic flow and minimize the number of dropped regular packets, based on a rhythmic task model proposed in [11]. Another example to literature that adds adaptive slot stealing scheme to the TDMA protocol for reducing the network delay is [4]. For the routing layer, a dynamic routing algorithm IDDR is proposed in [35] to simultaneously improve the fidelity for high data integrity applications and decrease the end-to-end delay for delay-sensitive applications. IDDR allows the packets with high integrity requirement be forwarded to the next hop with smaller queue length and allows the application packets with larger weights to choose shorter paths for low delay. Topology control is another active research area to dynamically tolerant node [18], [20], [22] and link failures [28], [23], [10]. To tolerate node failure, some heuristics algorithms, like CORP [14] in a distributed way and SpiderWeb [31] in a centralized way, are proposed to federate the disjoint network segments with least relay nodes and eventually improve the network reliability. To tolerate link failure, several algorithms [21], [19] mitigate the impact of lossy links by maintaining K-connectivity of the network. However, all aforementioned works do not consider control system performance during reconfiguration (only from the network perspective) and do not conduct systematic case studies, which is an important evaluation step in WCSs.

Integration and co-design of network and control system are effective for WCSs, as described next. A co-design of network topology conditions and control system stability is explored in [25]. The integration of fault-tolerant wireless network and control in NPPs are studied in [33]. In [16], the author proposes an algorithm on data link layer TDMA scheduling to achieve higher delivery ratio for emergency packets than regular packets. A case study of a wireless-controlled water tank is conducted. However, the author does not address the network delay imperfection in WCSs, since they assume the worst-case network delay is less than the control sampling period. In [3], the author derives a sufficient condition for the random access communication policy of shared wireless medium and design a control-aware random access communication policy. However, all of these works assume the network environment is stable or the network interference is at a given, fixed level. None of them present the interaction between network reconfiguration and control, which is our focus in this paper. We show how network reconfiguration affects control system performance, comparing with a state-of-the-art static network configuration.

III. Offline optimal network configuration

We first introduce a model describing the network-induced imperfection impact on the control system performance as caused by the packet loss and network delay. We then show how to find optimal network configuration set (there might be more than one optimal configuration) by using this model.

Similar to [16], we have two assumptions: (1) when a message is not received by the controller, the previously-

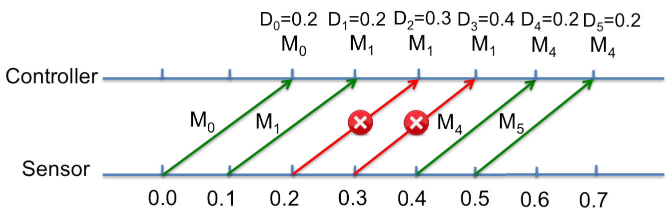

Figure 2: Network delay and delivery ratio tradeoff illustration, when network delay is greater than control sampling period.

received value will be used. (2) sensors do not fail, and produce real values (i.e., no noise). Under these assumptions, the model and algorithm to determine the optimal network configuration set are general and can be used in any WCS.

A. Network Imperfection Model

We define the delay induced into the control system by the wireless network as $T_{used} - T_{sensed}$, where T_{used} is the time the measurement signal is used by the controller and T_{sensed} is the time the sensor sends out the sensor measurement. The delay induced into the control system is

$$D = \lceil \frac{D_{network} + n_{loss}\Delta_{ssp}}{\Delta_{csp}} \rceil \Delta_{csp} \tag{1}$$

where $D_{network}$ is the network end-to-end delay, Δ_{csp} is the control sampling period, n_{loss} is the number of consecutive packet losses, Δ_{ssp} is the sensing sampling period. For example, as shown in Figure 2, the control sampling period and sensing sampling period are both 0.1s, but when the network delay is 0.2s and measurement M_2 gets lost, the induced delay D_2 is 0.3s and the controller will use measurement M_1 instead. When the measurement M_3 also gets lost, the induced delay D_3 is 0.4s and the controller will (re-)use measurement M_1.

D is related to both network delay and the number of consecutive packet losses, which is a function of the packet delivery ratio (delivery ratio estimation and end-to-end worst-case delay estimation have been studied elsewhere [33], [29]). n_{loss} is estimated by the expected value of the network loss ratio (1-DR). We assume message losses follow the uniform distribution, since DR can be viewed as the probability a message received by the controller. Thus, $n_{loss} = \sum_{i=1}^{n} i(1-DR)^i, ((1-DR)^i \geq thr)$, where $(1-DR)^i$ is the probability of i consecutive losses. When the probability is less than a threshold (thr), we assume that the probability can be ignored to avoid the computation running forever. For example, when $DR = 0.9$ and the $thr = 0.001$, the probability of getting 1, 2 and 3 consecutive losses are 0.1, 0.01, and 0.001, respectively. Since the probability of four consecutive losses is less than thr, we ignore the probability of more than three consecutive losses. Therefore, the expected number of consecutive losses is $1 \times (1-0.9) + 2 \times (1-0.9)^2 + 3 \times (1-0.9)^3 = 0.123$.

B. Optimal Network Configuration Determination Algorithm

Our offline algorithm discovers the set of network configurations that will be examined during the online portion. Since our goal is to minimize the network impact (network delay and package loss) to the control system, an optimal network configuration means configuration with minimum induced delay, D, considering the network imperfection model in Section III-A. It is important to note that the bigger the LSR value is, the bigger the delivery ratio is. Therefore, each LSR value corresponds to a delivery ratio. As shown in Algorithm 1, we find a set of estimated optimal network configurations for each average LSR value and store them in a look-up table T indexed by LSR value.

Initialization;
Configuration set C, All possible LSR set LSR, D_{min};
for $lsr_i \in LSR$ **do**
  reset D_{min} and $conf_{opt}$;
  for $conf_j \in C$ **do**
    calculate estimated induced delay D_i of $conf_j$
    if $D_j <= D_{min}$ **then**
      $D_{min} = D_j$;
      $conf_{opt} = conf_j$
    end
  end
  T.insert(lsr_i, $conf_{opt}$);
end
return T

Algorithm 1: The optimal network configuration set determination algorithm

IV. Online Network Reconfiguration

Due to the time-varying noise and interference in the environment, we devised an online dynamic network reconfiguration to improve the control system performance and minimize the total induced delay D.

The network reconfiguration is carried out by the controller, given that it has all the information needed to decide the optimal configuration for the current network status. In essence, we assume that the network conditions do not change as fast as the network reconfiguration algorithms execute and the nodes are time synchronized. When a reconfiguration is needed due to interference or noise, the controller broadcasts a new network configuration to all the nodes in the network. Note that even though our offline algorithm (Section III) is general, in this paper we restrict network configuration to refer to the number of sensor nodes in a certain area. To save network energy consumption to prolong the network lifetime, sleep nodes are activated when needed, or active nodes are put to sleep if not needed, creating different network topologies. For simplicity, we assume different topologies have different number of nodes.

Since we assume the worst-case network delay is greater than the control sampling period, packet re-ordering is possible. In this paper, the old packet is discarded, if the latest packet has already arrived at the remote controller.

Recall that online network reconfiguration is based on the offline look-up table given the current LSR. In this section, we first propose an algorithm to estimate LSR at run time. We then propose six online network reconfiguration algorithms, that is, three original algorithms combined with and without taking into account consecutive packet losses (see Section IV-C).

A. Network Average Link Success Ratio Estimation

Since Algorithm 1 generates a look-up table containing optimal network configurations for each LSR value, we need an estimation for the link success ratio when the reconfiguration algorithm is executed. We propose a jumping window in-network aggregation method to estimate the overall average network LSR. The idea is that each node calculates its own message receiving ratio, which is the average LSR for its receiving links, and adds it to its message. Then, each node will propagate its own average LSR to its parents, and parents will average their own LSR with their children's LSRs; this repeats until it gets to the controller.

Specifically, every node records the number of messages it receives during a certain amount of time, the LSR estimation interval (LSRI). Every node is able to calculate its average receiving LSR (calculated as the ratio of the number of messages it receives and the number of messages it should receive) from its receiving links during LSRI. At the end of an LSRI, each node concatenates its average receiving LSR with the message it needs to send, and sends the message to its parent nodes (one or more parent nodes). The parent node averages the received LSR from its children and its own LSR and sends out the message with the calculated average LSR over all its children (grandchildren, grand-grandchildren and etc.) and itself to its parents. Eventually, the remote controller will compute the final overall network average LSR.

B. Online Network Reconfiguration algorithms

The intuition behind the online algorithm is to find the optimal configuration according to the current estimated LSR, and then adjust the network topology to the optimal configuration. We explore three options to reach the optimal configuration, given that the optimality depends on the LSR, which cannot be computed instantaneously. The algorithms are DirectJump to Optimal (DO), Multiplicative Increase and Conservative Decrease (MICD), and Adaptive Control (AC). The input of these algorithms is the offline look-up table T.

1) DO: DirectJump to Optimum Algorithm: We adjust the network topology to have the exact number of nodes that correspond to the optimal network topology estimation, whenever the LSR value changes. See Algorithm 2 for details.

```
Initialization;
LSRCounter=0, curr_node = min_node;
while true do
    if LSRCounter == LSRI then
        do Link success ratio estimation, curr_LSR;
        get topology estimation from T, T(curr_LSR);
        curr_node=T(curr_LSR).node;
        change network topology to be curr_node;
        LSRCounter=0;
    end
    LSRCounter++;
end
```

Algorithm 2: Direct jump to optimum (DO)

2) MICD: Multiplicative Increase and Conservative Decrease Strategy: Given that a topology corresponds to different number of relay nodes, we were inspired by [30] and ensure the network is reliable, the number of nodes is multiplicatively (i.e., very quickly) increased when the current number of nodes is less than the estimated optimal number of nodes (similar to the TCP/IP protocols window reduction, but in a different context). When the current number of nodes is more than the estimated optimal number of nodes, the number of nodes is conservatively decreased (in our case, we reduce the current number of nodes by 1). Algorithm 3 shows more details.

```
Initialization;
LSRCounter=0, est_node=0, curr_node = min_node,
 increment = 1;
while true do
    if LSRCounter == LSRI then
        do Link success ratio estimation, curr_LSR;
        get topology estimation from T, T(curr_LSR);
        est_node=T(curr_LSR).node;
        if curr_node < est_node then
            curr_node = curr_node+increment;
            increment = increment×2;
        else if curr_node > est_node then
            curr_node = curr_node-1;
        else
            increment=0;
        end
        change network topology to be curr_node;
        LSRCounter=0;
    end
    LSRCounter++;
end
```

Algorithm 3: Multiplicative Increase and Conservative Decrease (MICD)

3) AC: Adaptive Control Algorithm: Inspired by adaptive control theory [9], in AC, the larger the difference between the estimated optimal number of nodes and the current number of nodes is, the faster we add or remove nodes. In Algorithm 4, α is a parameter that guides the speed of addition and reduction of nodes in the network ($0 < \alpha < 1$). When $\alpha = 0$, AC behaves like DO and the speed to add or reduce nodes is maximum. When $\alpha = 1$, the current number of nodes does not change, that is, it is a static network. In essence, the smaller the α is, the higher speed to change the current number of nodes. Algorithm 4 shows more details.

```
Initialization;
LSRCounter=0, est_node=0, curr_node = min_node;
while true do
    if LSRCounter == LSRI then
        do Link success ratio estimation, curr_LSR;
        get topology estimation from T, T(curr_LSR);
        est_node=T(curr_LSR).node;
        curr_node = α × curr_node + (1 − α) × est_node;
        change network topology to be curr_node;
        LSRCounter=0;
    end
    LSRCounter++;
end
```

Algorithm 4: Adaptive Control (AC)

C. Online Network Reconfiguration Algorithms Considering Consecutive Message Losses

From Equation 1, the total induced delay is proportional to the number of consecutive losses n_{loss}. Therefore, orthogonal to the algorithms in Section IV-B, we consider n_{loss}. Since the LSR estimation is inaccurate (we predict future average LSR based on the previous average LSR), there could be undetected consecutive message losses, which will degrade the control system performance. In other words, when there are consecutive message losses, we need to make the network more robust (we choose to add more nodes). As a first experimental step, whenever there are more than three consecutive message losses, we add k ($k = 3$) more nodes in the network. Considering consecutive losses, we devise three more online algorithms: CL-DO, CL-MICD, and CL-AC.

V. Primary Heat Exchanger Case Study

We conducted a case study to show and experiment with our wireless network reconfiguration for a nonlinear primary heat exchanger system (PHX) in a NPP [5]. Safety issues are beyond the scope of this paper; we focus on feasibility first.

The PHX has its main function the exchange of heat from inside of the reactor to the outside. The PHX is typically modeled as a nonlinear system, and a nuclear reactor model typically has three PHXs. For simplicity of presentation, we only discuss wireless control for one PHX, but when multiple PHXs are present, multiple streams could be isolated and/or combined. There are three measurements periodically sent to the remote controller, namely outlet hot leg temperature, inlet hot leg temperature, and mass flow rate. The deadline for each measurement to reach the controller is 0.586s, otherwise the system becomes unstable [33]. Another requirement is low power consumption to increase the network lifetime and reliability. We equate power with the number of active nodes.

We deploy a 12-hop and up to 50 nodes (9 sensors and 41 relay nodes) in the NPP under consideration. We wake up some inactive nodes or put some active nodes to sleep, as reconfigurations are needed. As shown in Figure 3, the network contains two parts: a k-connected region and a relay region. We connect these two regions by virtual roots, special sensor nodes. Messages go from sensors to virtual roots, and then to the controller; the routing in each region is done differently. Note that all the nodes in one level of the relay region can reach all nodes in the next level in our topology design.

In the k-connected region, there are k edge-disjoint paths (here we consider $k \leq 4$) from the sensors to the virtual roots. The number of paths is set depending on the current network link quality. The more edge-disjoint paths are required, the more nodes need to be active in the network and the higher DR is. In this paper, we activate paths from left to right. Experimentally, adding paths from right to left or randomly results in no significant differences.

In the relay region, there is a line of primary nodes and at most three lines of backup nodes from the virtual roots to the remote controller. In this paper, we activate backup nodes from highest level to lowest level. We have experimented with adding nodes from lowest to highest level and random, without significant differences in the results (see Section VI-B5).

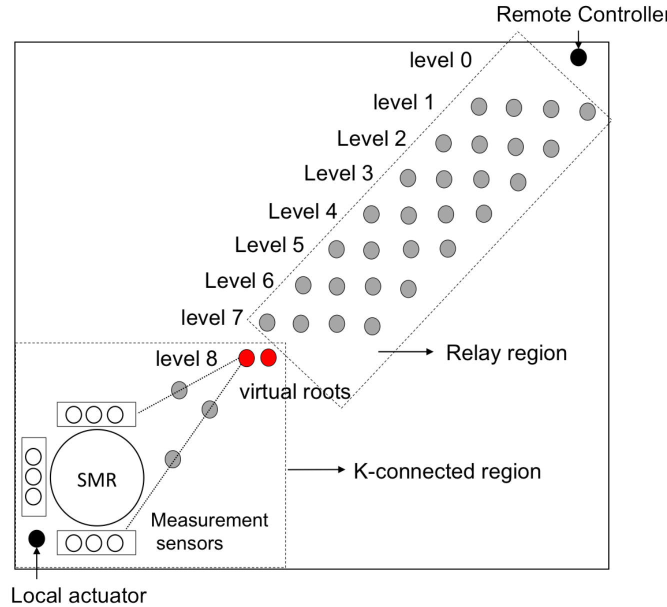

Figure 3: Network topology design

We use the bitvector protocol [34], which uses broadcast and TDMA scheduling to guarantee real-time transmission, as follows. Relay nodes broadcast messages level by level towards the controller. Within each level, primary node will broadcast first, then the first, second, and third backup nodes, in order. Therefore, the worst-case network delay is $n\Delta t$, where n is the number of the current active nodes and Δt is the time slot of TDMA scheduling (Δ = 10ms in the case study). If the backup parent finds out (while overhearing and checking the bit vector) that the primary did not send out the values, the backup parent compensates for it. Therefore, the more active nodes in the network, the higher DR and network delay are.

For testing, we combined the implementations of bitvector protocol with a state-of-the-art cyber-physical system simulator (WCPS 2.0 [16]) and allow wireless network to run together with the primary heat exchanger system simulink model. The online reconfiguration algorithms mentioned in Section IV are implemented on the controller.

To simulate time-correlated faults (see Section IV), we adjust each relay node's received signal strength (RSSI) [13] to change the quality of links (LSR) associated with the relay node to be in the range (0.5, 1.0). We use real-world noise traces [16] and change the range of link quality over time, depending on the following quantities. *rssi duration*: the time interval at which the RSSI is fixed (after that, the RSSI may be changed); *rssi range* and *time range*: the value and time range the RSSI duration is chosen from. We randomly choose RSSI from *rssi range* (-60dBm, -85dBm) with a uniform distribution and randomly choose *rssi duration* from *time range* $(0, 20s)$ also with a uniform distribution. Figure 4 shows an example of RSSI values; it's clear that the RSSI values vary a lot.

VI. Quantitative Results from Case Study

The quantitative results shown are from WCPS augmented with the algorithms developed herein; an operator changes the power output for the reactor. To evaluate the performance

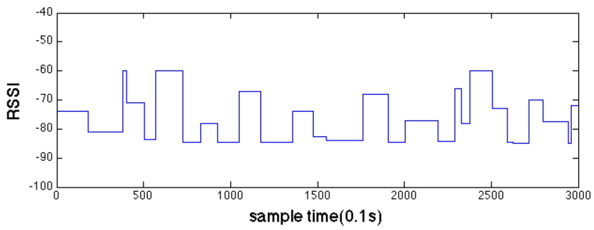

Figure 4: Time-correlated rssi variation example

Table I: Parameters and Values

Parameters	Values
Sensing sampling period	0.1s
Control sampling period	0.1s
Simulation time	300 seconds
LSRI Range	[2s, 20s]
Reference functions	[ramp30, ramp120]
α value	[0.1, 0.9]

of the control system (in this case the PHX), we adopted three metrics: Integral Absolute Error (IAE) [15], Maximum Absolute Error (MAE) [15] and RMS Error (RMSE), which is the RMS error measured between the closed-loop responses using wired control (i.e., no packet drops and no network delay) and wireless control under consideration. We only show RMSE in the paper, given that the IAE and MAE results are very similar because all these metrics describe the error between wireless and wired control (no packet loss and no delay). We also measure the induced delay (to analyze RMSE) of the network imperfection model, the number of nodes that are used in the network, and the network lifetime. Table I shows our simulation parameters and values.

A. Offline Optimal Network Configuration

By applying Algorithm 1, we can get the look-up table containing the optimal configuration set for each LSR value. Figure 5a shows the estimated optimal number of nodes for different LSR values. The higher the LSR, the higher the percentage of packets that get delivered, and the more robust the network will be, and therefore the fewer sensor nodes needed. The optimal number is always multiple of 10 and the worst-case delay is multiple of 0.1s (Δcsp) due to the network imperfection model calculation. For example, the network delay is taken as 0.3 if the actual values are above 0.2s and below 0.3s, because of the ceiling operation.

To correlate the network imperfection model (see Section III-A) and the control system performance, we run the simulation with static RSSI values. Figure 5b shows the induced delay for different number of nodes and different RSSI values. Note that when the number of nodes is 20, the network is not robust and has more consecutive message losses, thus has more induced delay although the network delay is the lowest (for the messages that are actually delivered). When the number of nodes is 50, the opposite behaviors occur.

Figure 5c shows the power output RMSE of the PHX. Comparing Figures 5b and 5c, we can see visually and statistically (Pearson correlation $r = 0.993$, $p < 0.001$) that our network imperfection model is significantly correlated to the power output RMSE.

B. Online Network Reconfiguration results

In order to simulate time-correlated link quality models, we fine-tuned the RSSI range and duration to get different representative network fault models with different average RSSI values. We simulate our system on five fault models with average RSSI values as -65 dBm, -70 dBm, -74 dBm, -78 dBm and -82 dBm. In this section, we present control system performance, network lifetime results for different online reconfiguration schemes.

1) Heat exchanger system power reference function: We change the power reference function (i.e., the required power output of a nuclear reactor), considering four ramp functions to reduce power from 42 MW to 37.8 MW as shown in Figure 6. Ramp30 means that it takes 30 seconds to reduce the power from 42 MW to 37.8 MW. As shown in Figure 7, 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 in much less time, and thus it will have more transient response, causing larger RMSE. We note that the online network reconfiguration schemes perform similarly for all reference functions. We only present the results for ramp30 to save space. Note that the RMSEs decrease with the decrease in slope, because the system has more time to adjust to the new output reference power. Thus, our scheme is most useful when there is little time for adjustment (e.g., quick correction of the power output) or when the difference in power is large (e.g., at startup of a new reactor when other reactors have to adjust their power output). Still, our scheme is also effective under all circumstances; our worst scenario is for ramp120, and the cumulative gains of our scheme (i.e., the difference in error we provide) is about 10MW over a 300-second interval (with RSSI = -82).

2) Comparison of Online Reconfiguration Schemes: Figure 8a shows the power output RMSE of the PHX for different average values of RSSI and different online network reconfiguration schemes proposed in Section IV; IAE and MAE results are similar to the RMSE results. In the figure, "static" is when the number of nodes in the network is fixed. We tested the number of nodes 20 to 50 and chose the static scheme with minimum RMSE among these tests. As expected, the more interference (lower RSSI) causes the dynamic network reconfiguration schemes perform better than the static scheme with minimum RMSE. This is because that when the average RSSI values of network fault models are -65dBm, -70dBm, and -74dBm, the LSR is mostly between 0.8 and 1.0. From Figure 5a, for those values of RSSI, the static scheme with 30 nodes can handle most of the network interference. However, when the network fault models are -78dBm and -82dBm, the LSR drops to between 0.85 and 0.5, and the system necessitates more frequent network reconfiguration.

3) Sensitivity Analysis of LSR Estimation Interval: For the network energy consumption, we calculate the average network energy consumption during one round of measurements of transmissions from sensor to the controller. We measure $\pi = \pi_{send} + \pi_{rec}$, where π is the sum of the sending and receiving energy for all active sensors (energy in sleep mode is ignored). Since we use a broadcast TDMA protocol, each active node broadcasts only once during each transmission round. We assume each active node consumes the same energy when broadcasting and thus: (a) $\pi_{send} = Ne_{send}$, where N

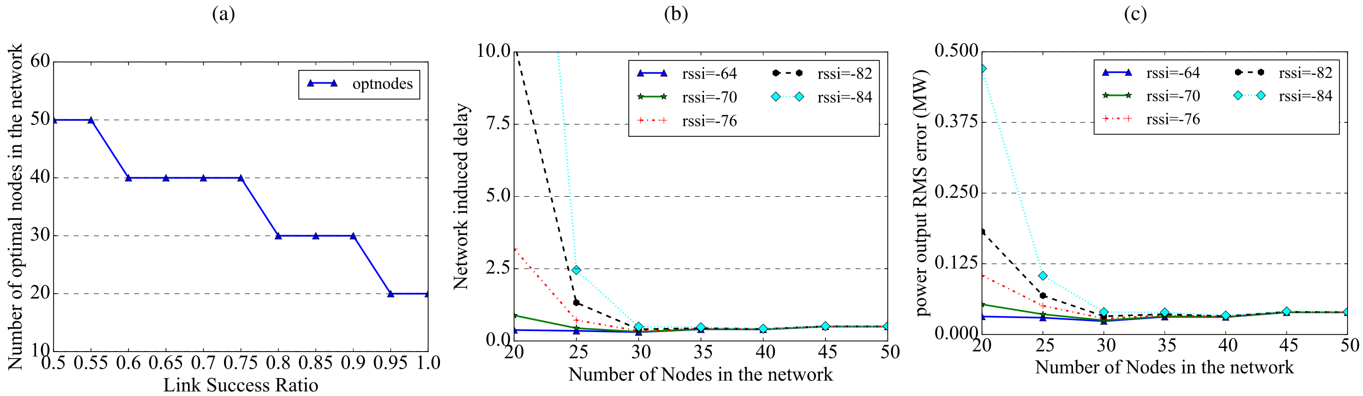

Figure 5: (a) Offline optimal network configuration determination for different LSR values (b) Total induced delay result with RSSI=-64 dBm (average LSR=0.93), RSSI=-70 dBm (average LSR=0.88), RSSI=-76 dBm (average LSR=0.82), RSSI=-82 dBm (average LSR=0.77) and RSSI=-84 dBm (average LSR=0.72) (c) power output RMSE (in MW) comparing with the network with no error no delay ($DR = 1.0$ and $\Delta_{network} = 0.0$) for different RSSI values, as a function of the nodes in the network

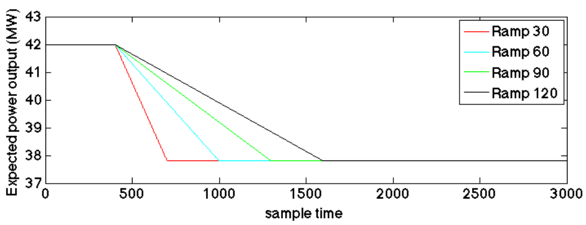

Figure 6: Control system power reference functions

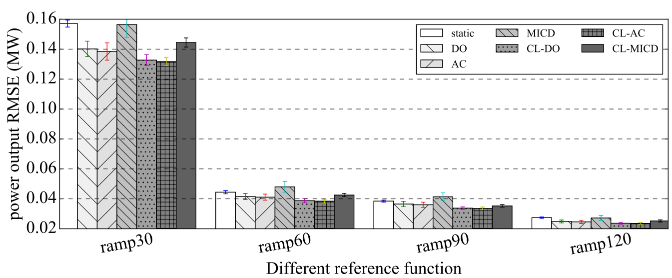

Figure 7: Power output RMSE for different reference functions (average RSSI=-82 dBm and LSRI = 2s (20 samples))

is the number of active nodes and e_{send} is the energy for one node of sending one message; (b) $\pi_{rec} = Ee_{rec}$, where e_{rec} is the energy consumption for one node of receiving a message and E is the total number of transmission links; and (c) $E = \sum_{l=1}^{L} n_{l-1}n_l$, where L is the total network levels, n_l is the number of nodes of level l (assuming all the nodes in one level can reach all nodes in the next level, as in Section V). For simplicity, we assume a general battery capacity is 8640J (two AA batteries), $e_{rec} = e_{send}$ =20 mA [1], transmission and receiving duration is 5ms and voltage is 1.5V. Figure 8b shows the network lifetime for different reconfiguration schemes. Schemes considering consecutive message losses (CL-DO, CL-AC and CL-MICD) consume more energy than their counterparts not considering consecutive message losses (DO, AC and MICD). This is because CL-* schemes are more aggressive adding additional nodes when there are consecutive losses. In

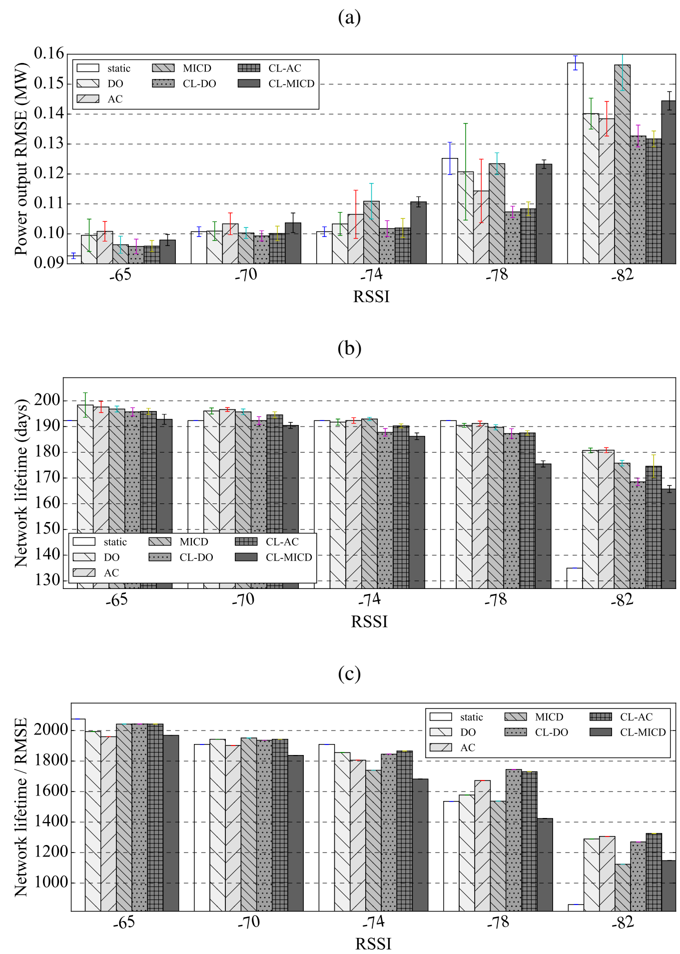

Figure 8: (a) Power output RMSE results, (b) network lifetime results and (c) network lifetime / RMSE results; (LSRI: 2s; AC and CL-AC schemes with α: 0.1)

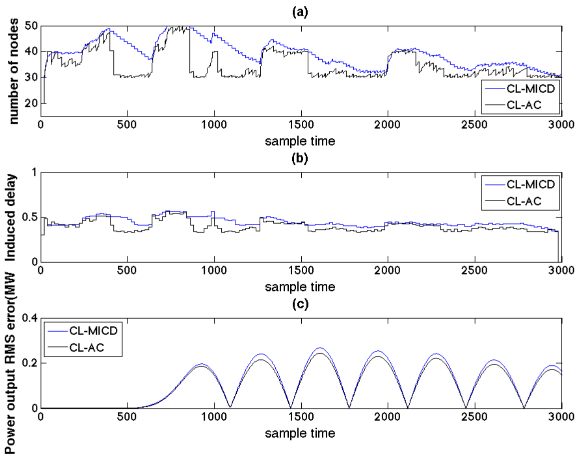

Figure 9: (a) Average number of nodes in the network, (b) average induced delay and (c) average RMSE over time (LSRI: 2s; Average RSSI: -82 dBm; CL-AC schemes with α: 0.1)

addition, from the Figure 8b, we found that when there is more interference in the network, the network consumes more energy, since the network needs more backup nodes to handle link failures. For network performance results, see Figure 16.

To consider both control system performance and network energy consumption together, we normalize network lifetime by RMSE in Figure 8c for different average RSSI values. The static scheme in fault model with average RSSI = -82dBm is significantly worse than the dynamic schemes, because it consumes the most network energy consumption, and it has the most RMSE, demonstrating that our reconfiguration schemes are necessary and work well when the network has more interference. Note that we selected the best (minimum RMSE) static scheme to be conservative in our evaluation, but in reality it would be hard to select a good static configuration *a priori*, since the network interference is unpredictable. In addition, CL-MICD always performs the worst among the dynamic reconfiguration schemes. Figure 9 shows the comparison over 3,000 samples of CL-MICD and CL-AC on 20 experiments. The reason CL-MICD always performs worse is because the speed of reducing the number of nodes is slow (reduce one at a time) and the speed of adding nodes is fast (exponential increase), which produces more induced delay (induced delay of CL-MICD is always higher than CL-AC) and degrades the control system performance.

Since LSR is estimated periodically, the length of the LSR estimation interval (LSRI) will affect the control system performance. Figure 10a shows the results of the power output RMSE for different LSRI values. When the LSRI increases, the RMSE of the schemes DO, MICD and AC increases because estimation is less accurate at high LSRI values. In Figure 11, the yellow line is the real LSR; the black line (LSRI of 2s) tracks the real LSR better than the LSRIs of 8s and 16s. Therefore, the control system performance gains less error, when the LSR estimation is accurate. Figure 12 shows the comparison of the DO with LSRI of 2s and DO with LSRI of 20s. From sample 600 to 800, the DO with LSRI of 20s runs with 30 nodes in the network because the LSR is estimated high averaged over the last 200 samples (from 400

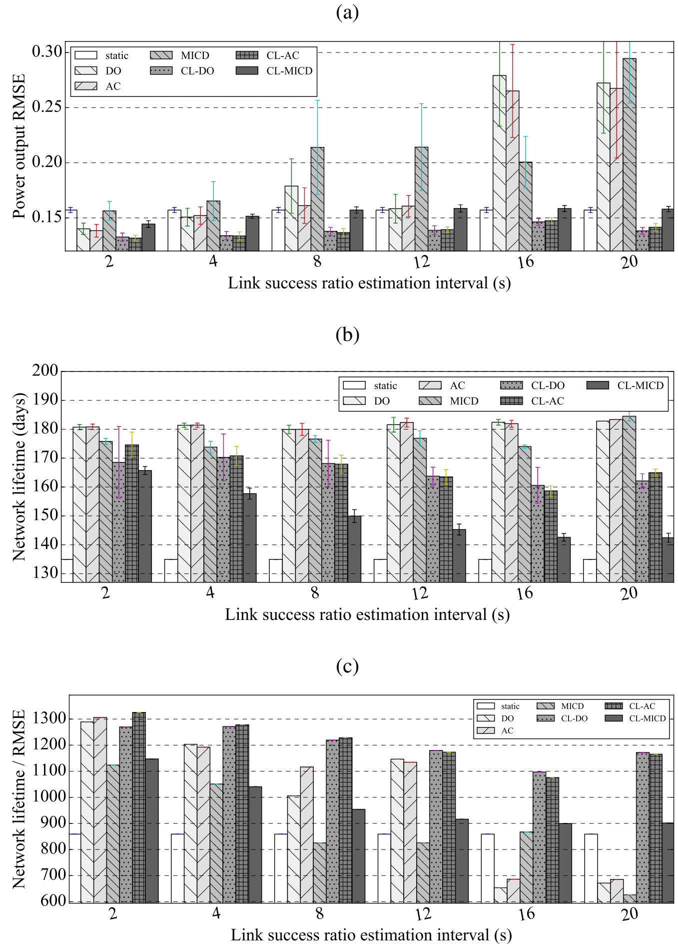

Figure 10: (a) Power output RMSE result, (b) network lifetime results and (c) network lifetime / RMSE results for different LSRIs (RSSI: -82 dBm; AC and CL-AC with α: 0.1)

to 600). However, from sample 600 to 800, the LSR is low (network has more interference) and 30 nodes cannot handle the link failures, which makes the consecutive message losses happen (induced delay D is high) and negatively affects the control system performance. Note that the RMSEs of the CL-* schemes are not affected by the LSRI values (similar results for the other fault models) because, even though the LSR estimation may not be accurate, CL-* schemes add additional nodes to compensate to make the network robust. However, the side-affect is that CL-* schemes consume more energy. Figure 10b and 10c show the network lifetime and the network lifetime normalized by RMSE. For network performance (network delay and DR), see Figure 17.

4) Adaptive control algorithm with different alpha values: Recall that the AC scheme has a variable alpha ($0.1 \leq \alpha \leq 0.9$), which determines the speed to add or reduce nodes in the network (small α, fast node adding). The α value can also affect the control system performance. Figure 13 shows the RMSE of AC and CL-AC schemes for different α values. When $\alpha > 0.5$, the control system performs worse. This is because the speed of adding or removing nodes is so slow that it cannot react to the LSR variation in time. Figure 14 shows the reason more clearly. From sample 600 to 800, when

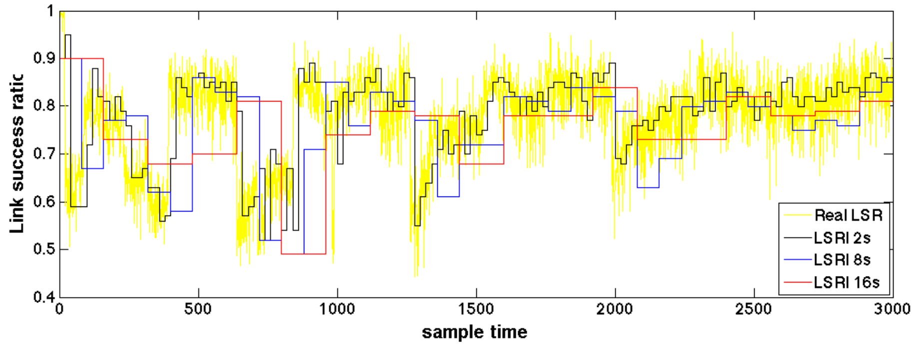

Figure 11: Comparison of estimated and real LSRs (average RSSI = -82dBm)

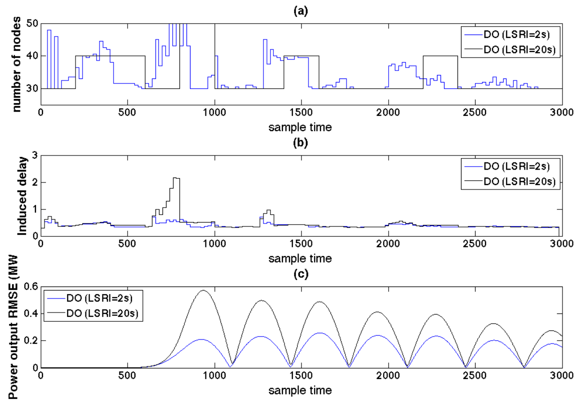

Figure 12: (a) Average number of nodes in the network, (b) average induced delay and (c) average RMSE over time (LSRI: 2s; Average RSSI value: -82 dBm;)

the network has more interference, the speed of AC (α=0.9) of adding nodes is slower than the AC (α=0.1), causing consecutive message losses and more induced delay. From sample 800 to 1300, when the network has less interference, the speed of AC (α=0.9) of reducing nodes is also slow and induce more delay (network delay is high) into the control system. From Figure 13, we also find that CL-AC always performs better than AC. Although the speed to add or reduce nodes is slow for α 0.9, considering consecutive losses can compensate somewhat when the network has more interference. Figure 14 shows more details. From sample 600 to 800, when the network has more interference (consecutive message losses happen), CL-AC (α=0.9) adds more nodes in the network than AC (α=0.9), which improves control system performance. But CL-AC may add more nodes than needed. From sample 1300 to 1600, CL-AC (α=0.9) adds too many nodes and causes more induced delay (see Figure 14 (b)) and thus consumes more network energy consumption comparing with AC (α=0.9).

5) Comparison of Nodes Activation Methods: Our experiments show that adding nodes from the highest level to the lowest level (deactivating nodes in the opposite direction: from lowest to highest level), from the lowest level to the highest level, or randomly. For lack of space, we do not show all the results, but the three methods behave very similarly for DR and network delay. In the same vein, the results for AC and CL-AC values are also similar, because that our fault model generates

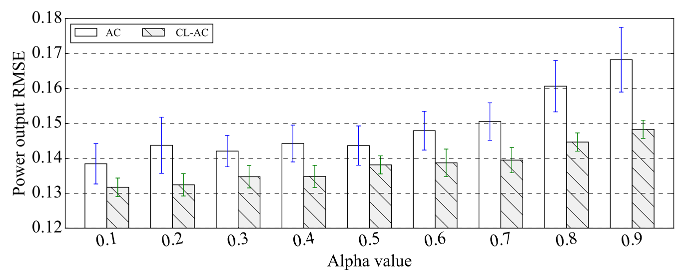

Figure 13: Power output RMSE result comparison of AC and CL-AC for different alpha values (average RSSI value: -82dBm; LSRI: 2s)

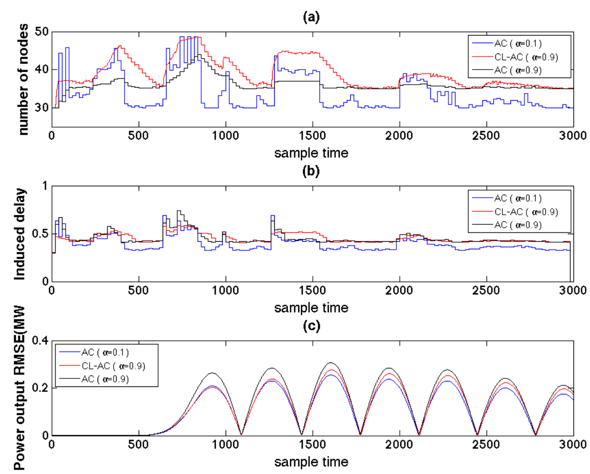

Figure 14: (a) Average number of nodes in the network, (b) average induced delay and (c) average RMSE over time for AC (α=0.1), CL-AC (α=0.9) and AC (α=0.9) (average RSSI value: -82dBm; LSRI: 2s;)

time-correlated faults, instead of space-correlated faults.

VII. Conclusion and Future Work

In recent years, the WCS controlled over multi-hop wireless sensor network has been explored. However, the interaction between network reconfiguration and control, when the worst-case network delay is bigger than the control sampling period has not been researched. In this paper, we propose and implement a framework with offline and online parts. In

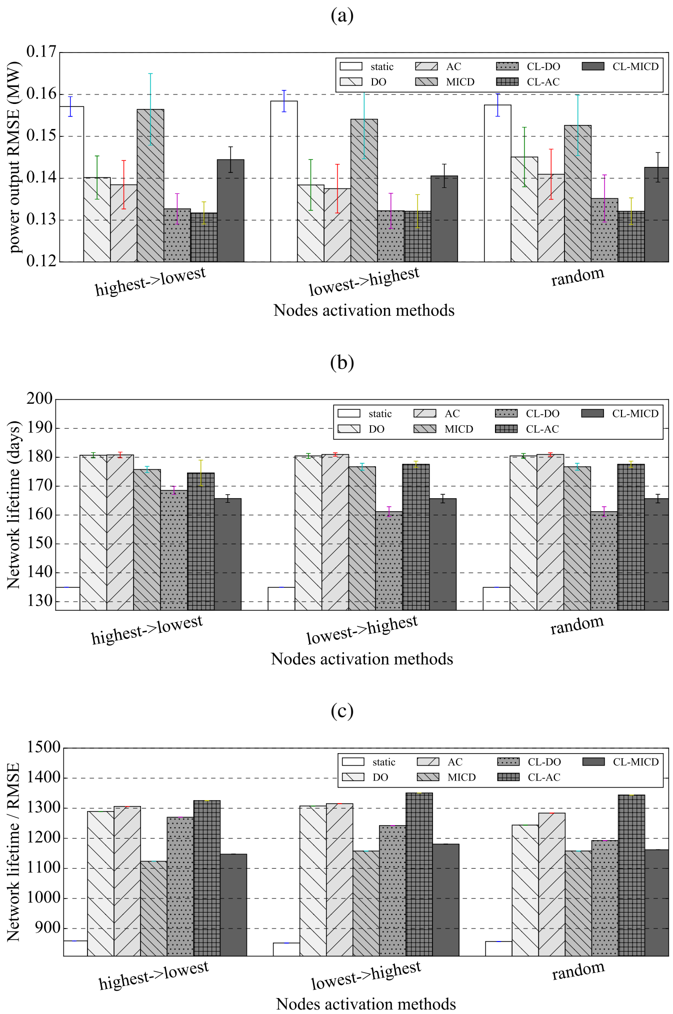

Figure 15: (a) Power output RMSE result, (b) network lifetime results and (c) Network lifetime / RMSE results for different adding/reducing nodes methods (the average RSSI value: -82 dBm; AC and CL-AC schemes with α: 0.1; LSRI: 2s)

detail, we propose a network imperfection model and network configuration determination estimation offline; we propose six online network topology control algorithms with and without considering consecutive losses. We also conduct a systematic case study to see the in-depth interaction between network reconfiguration and the control. Our simulation result shows that our network imperfection model is accurate with Pearson correlation 0.993, that network reconfiguration works better than the static scheme showing low error (RMSE, MAE, IAE), and showing longer network lifetime. We find that consecutive message losses can degrade the control system performance due to more total induced delay.

In the future, aside from reconfiguring the number of nodes, we will explore other network aspects, such as routing layer reconfiguration and data link layer reconfiguration.

Appendix

A. *Network delay and network delivery ratio*

Figures 16 and 17 show that network delay and DR have similar tendency. The higher the network delay is, the higher the DR is. This is because that when the network delay is high, the network has more backup nodes, which increases the DR.

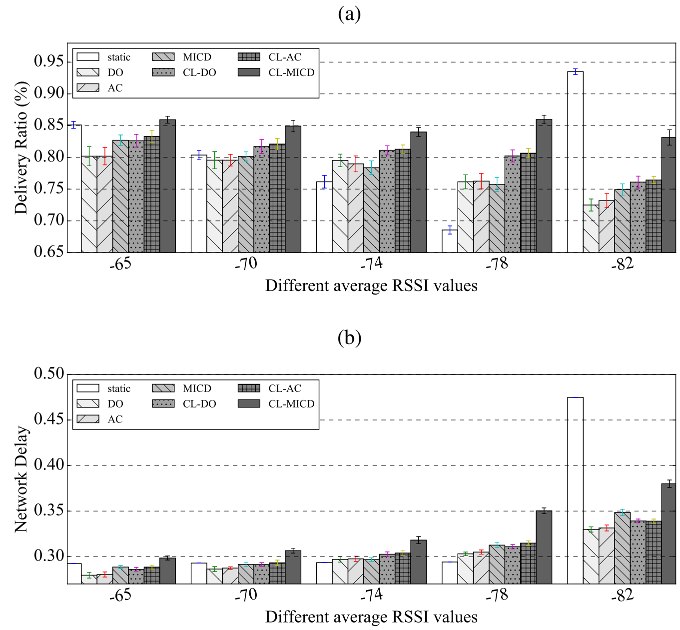

Figure 16: (a) Network delivery ratio and (b) network delay for different average RSSI values

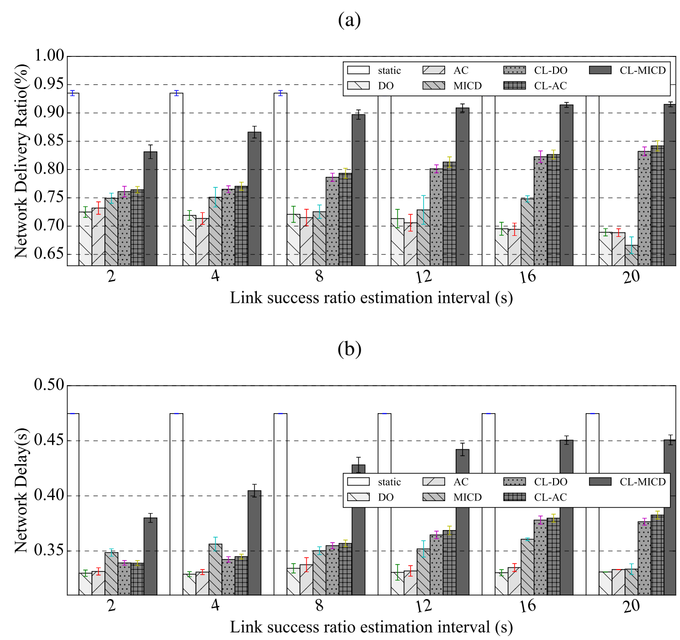

Figure 17: (a) Network delivery ratio and (b) Network delay for different LSRIs(the average RSSI value: -82 dBm; AC and CL-AC schemes with α=0.1)

References

[1] Micaz mote: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

[2] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Trans on Sensor Networks*, 8(4):34, 2012.

[3] 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.

[4] S. Gobriel, D. Mosse, and R. Cleric. Tdma-asap: Sensor network tdma scheduling with adaptive slot-stealing and parallelism. In *Int'l Conf on Distributed Computing Systems (ICDCS) 2009.*, pages 458–465, 2009.

[5] 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.

[6] V. C. Gungor, B. Lu, and G. P. Hancke. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE transactions on industrial electronics*, 57(10):3557–3564, 2010.

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

[8] 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.

[9] N. Hovakimyan and C. Cao. *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. SIAM, 2010.

[10] V. Kawadia and P. Kumar. Power control and clustering in ad hoc networks. In *INFOCOM 2003*, volume 1, pages 459–469. IEEE, 2003.

[11] J. Kim, K. Lakshmanan, and R. R. Rajkumar. Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems. In *Int't Conference on Cyber-Physical Systems*, pages 55–64, 2012.

[12] K.-D. Kim and P. Kumar. The importance, design and implementation of a middleware for networked control systems. In *Networked Control Systems*, pages 1–29. Springer, 2010.

[13] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In *2007 6th International Symposium on Information Processing in Sensor Networks*, pages 21–30. IEEE, 2007.

[14] S. Lee and M. Younis. Optimized relay placement to federate segments in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 28(5):742–752, 2010.

[15] 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.

[16] B. Li, L. Nie, C. Wu, H. Gonzalez, and C. Lu. Incorporating emergency alarms in reliable wireless process control. In *Int'l Conference on Cyber-Physical Systems*, pages 218–227. ACM, 2015.

[17] 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.

[18] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *ACM symposium on Principles of distributed computing*, pages 264–273. ACM, 2001.

[19] N. Li and J. C. Hou. Flss: a fault-tolerant topology control algorithm for wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 275–286. ACM, 2004.

[20] N. Li, J. C. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, 2005.

[21] X.-Y. Li, P.-J. Wan, Y. Wang, and C.-W. Yi. Fault tolerant deployment and topology control in wireless networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 117–128. ACM, 2003.

[22] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan, and S. Ravi. Algorithmic aspects of topology control problems for ad hoc networks. *Mobile Networks and applications*, 10(1-2):19–34, 2005.

[23] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *European Wireless Conference*, pages 156–162, 2002.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 404–413. IEEE, 2000.

[29] 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.

[30] Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz. Esrt: event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 177–188. ACM, 2003.

[31] F. Senel, M. F. Younis, and K. Akkaya. Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 60(4):1835–1848, 2011.

[32] 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.

[33] 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, 2016.

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

[35] J. Zhang, F. Ren, S. Gao, H. Yang, and C. Lin. Dynamic routing for data integrity and delay differentiated services in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 14(2):328–343, 2015.

[36] 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.