ENERGY-AWARE FAULT-TOLERANT AND
REAL-TIME WIRELESS SENSOR NETWORK FOR
CONTROL SYSTEM

by

Wenchen Wang

Bachelor of Engineering
Northeastern University, China
2013

Submitted to the Graduate Faculty of
the Kenneth P. Dietrich School of
Arts and Sciences in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh
2017
UNIVERSITY OF PITTSBURGH

DEPARTMENT OF COMPUTER SCIENCE

This proposal was presented

by

Wenchen Wang

Dr. Daniel Mosse, Department of Computer Science, University of Pittsburgh
Dr. Rami Melhem, Department of Computer Science, University of Pittsburgh
Dr. Youtao Zhang, Department of Computer Science, University of Pittsburgh
Dr. Daniel Cole, Department of Mechanical Engineering and Materials Science, University of Pittsburgh
Dissertation Director: Dr. Daniel Mosse, Department of Computer Science, University of Pittsburgh
ENERGY-AWARE FAULT-TOLERANT AND REAL-TIME WIRELESS SENSOR NETWORK FOR CONTROL SYSTEM

Wenchen Wang, PhD candidate

University of Pittsburgh, 2017

Wireless control systems enable several advantages over traditional wired industrial monitoring and control systems, including self-organization, flexibility, rapid deployment, and lower maintenance. However, wireless network delay and packet loss can result in two main challenges for the control system: instability and performance degradation.

This dissertation aims at solving these two major challenges. We have developed a fault-tolerant network design and a novel computational model to quantify the control system stability requirement based on the network design. We have explored a hybrid offline online network reconfiguration framework with time-correlated link failures to improve control system performance. Accordingly, a precise analytical model and different reconfiguration algorithms have been developed to quantify and improve the performance, respectively.

I propose to continue the research in two aspects. First, I propose to extend the network reconfiguration scheme to consider space-correlated link failures. I will conduct a case study to evaluate the reconfiguration scheme. Then, I propose to develop a cross-layer network scheduling algorithm with known application demands. I will study the impact of real-time scheduling on control system performance via another case study.
TABLE OF CONTENTS

1.0 INTRODUCTION .................................................. 1
   1.1 Problem Statement .......................................... 1
   1.2 Solutions and Contributions ............................... 2
   1.3 Outline ..................................................... 4

2.0 RELATED WORK AND BACKGROUND ............................. 5
   2.1 Related Work ............................................... 5
   2.2 Background ............................................... 7

3.0 FAULT-TOLERANT NETWORK DESIGN (COMPLETED) .......... 9
   3.1 Problem Statement ......................................... 9
   3.2 Proposed Solution ......................................... 10
   3.3 Performance Evaluation ................................... 12
   3.4 Summary .................................................. 14

4.0 NETWORK RECONFIGURATION: TIME-CORRELATED FAULTS
   (COMPLETED) .................................................. 15
   4.1 Problem Statement ......................................... 15
   4.2 Proposed Solution ......................................... 16
      4.2.1 Offline computation .................................. 17
      4.2.2 Online reconfiguration ............................... 17
   4.3 Performance Evaluation ................................... 19
   4.4 Summary .................................................. 22

5.0 NETWORK RECONFIGURATION: SPACE-CORRELATED FAULTS
   (FUTURE) .......................................................... 24
5.1 Motivation ................................................................. 24
5.2 Problem Statement and Proposed Work .......................... 24

6.0 REAL-TIME NETWORK SCHEDULING (FUTURE) ............ 27
   6.1 Motivation ............................................................ 27
   6.2 Problem Statement and Proposed Work ....................... 29

7.0 SUMMARY ............................................................. 31

8.0 BIBLIOGRAPHY ..................................................... 33
1.0 INTRODUCTION

Wireless control systems (WCSs) are composed of controllers, sensors, and actuators connected via a wireless network. WCSs controlled over multi-hop wireless sensor networks (WSNs) have received significant attention in recent years [Han et al., 2011; Li et al., 2015, 2016; Pajic et al., 2011b,a; Wang et al., 2016; Kim and Kumar, 2010]. WCSs enable several advantages over traditional wired industrial monitoring and control systems, including self-organization, rapid deployment, and flexibility [Gungor and Hancke, 2009]. While early success of WSNs has been recognized, significant potential remains in exploring WSNs as energy efficient fault tolerance and real-time networks for industrial plants.

1.1 PROBLEM STATEMENT

Even though WSNs are good for place-and-play deployment, wireless network communications are imperfect in terms of packet loss and network delay. Most WCSs are deployed and applied in industrial environments, such as smart grids [Gungor et al., 2010], water tanks [Li et al., 2015] and even nuclear power plants (NPPs) [Wang et al., 2016]. Harsh and complex electric-power-system environments pose great challenges in the reliability of WSN communications. Interference is the main factor of packet losses in WSNs [Li, 2015], where wireless links exhibit widely varying characteristics over time and space due to moving people/obstacles and electromagnetic and radio frequency interference (EMI/RFI) [Baccour et al., 2012; Kar and Moura, 2009; Baccour et al., 2012; Gungor et al., 2010; Ganesan et al., 2001]. These interferences can make some links/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). Moreover, time delay is another issue of WSNs due to interference and multi-hop characteristic. Real-time scheduling has been studied in WSNs to constraint network delays in [Saifullah et al., 2010; Gobriel et al., 2009b; Stankovic et al., 2003].

A WCS is a system with two subsystems, wireless sensor network and physical system (PS). The performance of one subsystem will affect the other. Network-induced imperfections [Zhang et al., 2013; Gupta and Chow, 2010], packet loss and time delay can result the control system in two main problems: instability [Zhang et al., 2001; Zhang and Yu, 2008; Jusuf and Joelianto] and performance degradation [Li et al., 2016; Pant et al., 2015]. When the physical system is unstable, the plant or part thereof can be damaged and leads to serious safety issues and financial loss. On the other hand, network can introduce unreliable/non-deterministic levels of service in terms of delays and losses and induce undesirable additional errors, that is, network-induced error. In addition, during WSN design, besides reducing network-induced imperfections, low energy consumption of WSN is desired, since sensors are typically battery supported [Yick et al., 2008].

This thesis breaks the two control problems described above into three sub-problems and studies different fault tolerance and real-time techniques in WSNs to solve them:

- **P1**: control system stability guarantee;
- **P2**: network-induced error reduction for a single control system;
- **P3**: total network-induced error reduction for multiple control systems.

### 1.2 SOLUTIONS AND CONTRIBUTIONS

To solve the above three sub-problems, we present four corresponding WSN solutions in-depth, including our accomplished work and our proposed work in the following chapters. Figure 1 shows the relationship between the problems and solutions.

- **S1**: Fault-tolerant Network Design (Completed). Given a control system stability requirement in terms of network delay and packet losses, we proposed a fault-tolerant network
Figure 1: Thesis problems and solutions relationship

node placement design and a computation model to estimate the minimum number of nodes placing in the network to meet the control system stability requirement with minimum energy consumption [Wang et al., 2016]. See Chapter 3.

- **S2: Network Reconfiguration for Time-correlated Link Failures (Completed).** To tolerate time-correlated link failures, we explored a network reconfiguration framework with offline and online parts. We studied the trade-off between the network delay and packet losses and presented a network-imperfection model in the offline part. An average link quality estimation mechanism and six centralized network reconfiguration algorithms were developed in the online part [Wang et al., 2017b,a]. See Chapter 4.

- **S3: Network Reconfiguration for Space-correlated Link Failures (Future).** To tolerate space-correlated link failures, we will study the fault-tolerant network reconfiguration technique. We propose a space-correlated fault model with one moving interference source and a new distributed online network reconfiguration algorithm, based on bitvector protocol [Wang et al., 2015]. We will conduct a case study to analyze the interaction between network reconfiguration and control. See Chapter 5.

- **S4: Real-time Network Scheduling (Future).** We observed that network delay significantly influences the control system performance, whose applications have dynamic demands [Wang et al., 2017c]. This observation motivates us to study the cross-layer dy-
namic real-time network scheduling with dynamic application demands of different PSs, in order to reduce the total network-induced errors for multiple control systems. We propose a network scheduling algorithm and will experimentally evaluate our algorithm that takes application demands into account via a case study. See Chapter 6.

1.3 OUTLINE

The rest of this proposal is organized as follows: Chapter 2 reviews existing fault tolerance and real-time techniques in WCNs and introduces the background of this thesis, and Chapter 3 introduces the energy-aware fault-tolerant network design approach and results. In Chapter 4, we build a network reconfiguration framework for time-correlated link failures in WSN. A space-correlated link failure model and a new network reconfiguration algorithm for space-correlated link failures are discussed in Chapter 5. In Chapter 6, a dynamic real-time network scheduling motivation and future works are discussed. Chapter 7 concludes the proposal and provides a timeline of the proposed works.
2.0 RELATED WORK AND BACKGROUND

2.1 RELATED WORK

The solutions for network delay and packet losses in WCSs are typically divided into three categories: control only, network only, and control and network co-design solutions. In this thesis, we only review the last two categories of fault tolerance and real-time techniques.

Fault Tolerance Technique in WSNs: Link failures are categorized as independent and correlated failures.

For independent link failures, fault tolerance protocols in WSNs have been presented in [Gobriel et al., 2006, 2008, 2009a]. Reliable routing algorithms for WCS have been explored in [Han et al., 2011]. Energy-aware routing for real-time and reliable wireless industrial sensor networks is introduced in [Heo et al., 2009]. Fault-tolerant node placement algorithms for link failures, k-edge disjoint algorithms have been investigated in [Frank and Tardos, 1989; Han et al., 2010]. However, all aforementioned works focus solely on the network without considering control aspect of a WCN. We solve control system instability issue by a flexible fault-tolerant node placement design and network imperfection estimation (see Chapter 3).

For correlated link failures, the link quality fluctuates over time [Cerpa et al., 2005; Srinivasan et al., 2010] and space [Zhou et al., 2006; Zhao and Govindan, 2003; Reijers et al., 2004; Cerpa et al., 2003]. Several research works focus on the statistical characterization of wireless links through estimation theory, link quality estimation (LQE). PRR (packet reception ratio)-based passive LQE algorithms are presented in [Woo and Culler, 2003; Cerpa et al., 2005]. Based on these LQE algorithms, network reconfiguration schemes are studied in routing [Zhang et al., 2015] and topology control [Ramanathan and Rosales-Hain, 2000; Narayanaswamy et al., 2002; Kawadia and Kumar, 2003]. Unfortunately, these works do not
consider control system performance. We design a network reconfiguration framework with a network imperfection model, indicating control system performance (see Chapter 4).

Interference source (IS) is one of the main factors of space-correlated link failures [Bac-cour et al., 2012], which are normally solved with two steps: IS detection and interference defense. IS detection usually achieves by statistical analysis of PRR or RSSI (received signal strength indicator) [Xu et al., 2006]. There are two typical approaches for interference defense: channel surfing or spatial evasion [Xu et al., 2006; Xu, 2007], and adjusting resources [Li et al., 2007; Bhavathankar et al., 2017]. However, the ISs in these works are all stationary. A distributed mobile jammer tracking scheme for a mobile IS is proposed in [Wei et al., 2016]. However, the authors only track the IS without solving the interference problem and do not consider control system performance. We will propose a space-correlated link failure model with a mobile IS and will propose a reconfiguration algorithm to defense the interference considering control system performance (see Chapter 5).

Fault-tolerant co-design of network and control system are effective for WCSs. A co-design of network topology conditions and control system stability is explored in [Pajic et al., 2011a]. A case study is conducted to see the interaction between model predictive control and network routing schemes in [Li et al., 2016]. The authors propose an algorithm of data link layer TDMA scheduling to achieve higher delivery ratio for emergency packets than regular packets in [Li et al., 2015]. However, none of these works address the tradeoff between network delay and packet loss imperfection in WCSs, nor present the interaction between network reconfiguration and control. We conduct case studies to show how the network reconfiguration affects control system performance (see 4 and Chapter 5).

Real-time scheduling in WSNs: Dynamic real-time network scheduling is an effective solution to constrain network delays. Although real-time TDMA scheduling algorithms in WSN are studied in [Stankovic et al., 2003; Saifullah et al., 2010; Gu et al., 2009; Liu et al., 2006], the authors do not consider control system application demands. Online data link layer scheduling is studied based on a rhythmic task model [Kim et al., 2012] in [Hong et al., 2015]. While the impact of network dynamics on existing network flows is minimized, overall control system performance is not considered and there is no case study for real-world applications. A popular cross-layer optimization is the one for wireless multimedia systems,
where videos and large files have to abide by a certain QoS (quality of service) [van der Schaar and Turaga, 2007; Zhang and Zhang, 2008; Andreopoulos et al., 2006], which is different from our approach of small data packets over small multi-hop sensor and relay nodes. For co-design of real-time network and control system, [Li et al., 2015] explores a real-time communication scheme with slot stealing algorithm [Gobriel et al., 2009b] to reserve time slots for emergency packets. However, neither network-induced error minimization is considered, nor multiple control systems are involved. In [Gatsis et al., 2016], the authors design a control-aware random access communication policy of a shared wireless medium. However, the authors do not consider the demand of application layer. To the best of our knowledge, cross-layer real-time scheduling has not been studied in the wireless control system, which is our proposed work (see Chapter 6).

2.2 BACKGROUND

The control system studied in this thesis is a nonlinear primary heat exchanger system (PHX) in an NPP [Greene et al., 2010], as shown in Figure 2. This thesis focuses on the wireless network design (marked as yellow area) for the control system. 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. We discuss wireless control for one PHX in Chapter 3, Chapter 4 and Chapter 5, and wireless control for multiple PHXs in Chapter 6. There are three measurements for each PHX periodically sent to the remote controller, namely outlet hot leg temperature, inlet hot leg temperature, and mass flow rate. Similar to [Li et al., 2015], when the measurements get lost during transmission, the controller uses the last received measurements to do the control.

The network protocols we use in this thesis are based on a TDMA protocol, ridesharing [Gobriel et al., 2006] to tolerate link failures. In ridesharing (shown in Figure 3), A node has one primary parent and one or more backup parents. If the backup parent finds out (while overhearing) that the primary did not send out the values, the backup parent will
compensate for it. For example, in Figure 3, $C_1$ has one primary parent $P_1$ and a backup parent $P_2$ and the link between $C_1$ and $P_1$ fails. When $C_1$ broadcasts its message at time slot 0, $P_2$ receives the message. When $P_1$ broadcasts its message at time slot 1, $P_2$ overhears $P_1$’s message and knows that $P_1$ does not receive $C_1$’s message. $P_2$ then sends $C_1$’s message at time slot 2. In this way, the backup parent $P_2$ tolerates the link failure and improves the network reliability. The more backup parents, the more reliable the network.

In this thesis, we make some assumptions as follows. We assume that links fail with a certain probability independently or correlatively 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 use network delivery ratio (DR) as the network reliability indicator, that is, the ratio of arrived messages ($DR \in [0, 1]$).
3.0 FAULT-TOLERANT NETWORK DESIGN (COMPLETED)

Control system stability is critical for physical plants, since system instability can result in plant damage and serious safety issues [Zhang et al., 2001; Zhang and Yu, 2008; Jusuf and Joelianto]. In WCS, network delay and packet loss are the potential threats to control system stability. Given a control system stability requirement in terms of network delay and packet loss, we first propose a flexible fault-tolerant node placement design. We then develop a computation model to meet the requirement, and to determine the initial network topology with the minimum number of nodes for a given average LSR [Wang et al., 2016].

3.1 PROBLEM STATEMENT

In PSs, the stability of the plant depends on the control system receiving sensor data in a timely fashion. We define network health (NH), $NH$, as the general control system stability requirement in terms of network delay $\Delta_{\text{network}}$ and delivery ratio (DR), $dr$,

$$NH = p_1 \Delta_{\text{network}}^2 + p_2 \Delta_{\text{network}} + p_3 - (1 - DR)$$  \hspace{1cm} (3.1)$$

where $p_1$, $p_2$, $p_3$ are characteristic constants for a specific controller. When $NH \geq 0$, the control system is stable.

We assume only network links fail independently with a certain probability. In this chapter, we focus on designing a fault-tolerance wireless network to meet the control system stability requirement with minimum network energy consumption. Our design input is the control system stability requirement, $NH$. 
3.2 PROPOSED SOLUTION

Since the control system is far from the sensors and actuators, the data and control signals will be transmitted through relay nodes. To improve reliability, we deploy relay nodes in the PS in two regions: a *k*-connected region and a *relay region* (connected by *virtual roots*) as shown in Figure 4. In the *k*-connected region, there are *k*-edge disjoint paths with the minimum node placement from measurement sensors to *virtual roots* [Han et al., 2010; Frank and Tardos, 1989]. We also add (*k*−1) backup nodes to the virtual root to improve reliability. In the relay region, primary relay nodes are placed in a “straight line” between the virtual roots and the remote controller (RC), called the *line of primary relay nodes*. The distance between two consecutive primary nodes is the same. In addition, there may be one or more *lines of backup nodes* from virtual roots to the RC. For each hop, one primary node and its backup nodes are called *level*. Note that we place backup nodes as close as possible to achieve minimum node placement in relay region (please refer to [Wang et al., 2016] for theorem proof details). Each node in level *l* is able to listen to all the nodes in level *l*−1 and level *l*+1.

![Figure 4: Fault tolerant relay nodes placement for single control system](image)

Figure 4: Fault tolerant relay nodes placement for single control system
After placing the relay nodes in the network area, backup nodes can be put to sleep to save energy when the LSR is high, and some of the backup nodes need to be activated when the LSR is low. We propose a computation model to quantify the network delay and delivery ratio in terms of average LSR and number of nodes. A fixed number of nodes can yield different $k$-connected regions and different number of backup nodes in each level. We modified the ridesharing protocol [Gobriel et al., 2006] to concatenate messages, instead of aggregating the measurements. Based on ridesharing protocol [Gobriel et al., 2006], the worst-case network delay is $\Delta_{\text{network}} = n\Delta_{\text{slot}}$, where $n$ is the number of the current active nodes and $\Delta_{\text{slot}}$ is the time slot of TDMA scheduling ($\Delta_{\text{slot}} = 10\text{ms}$). We estimate the DR at the RC as the expected number of messages received at the RC: $dr = \sum_{i=1}^{m} (p_{RC}(i) \times i)$, where $m$ is the total number of measurements sent from all the sensors, and $p_{RC}(i)$ is the probability that RC receives exactly $i$ messages.

We design a dynamic programming algorithm to calculate $p_{RC}(i)$ $(1 \leq i \leq m)$ for different number of nodes in the network with different average LSR values. We introduce the concept of state to represent message-receiving situation of a level. A state $i$ of level $l$ is $[[m_0, m_1, ... m_n], p_i]$, where $m_i$ is the number of messages received by one of the nodes in level $l$, $[m_0, m_1, ... m_n]$ is a sorted array, and $p_i$ is the probability of state $i$. Each level has multiple states. To calculate the probability of all possible number of messages received by the RC, we need to enumerate all possible states each level could have. For each level, we carry out the calculation with two phases, namely, a states-generating phase and states-combining phase. For the former, states are generated from each of the states of the previous level. For the states-combining phase, the probabilities of states with the same message receiving arrays are summed up and combined into one state. Since we compute each state’s probability level by level from the measurement sensors to the RC, all possible number of messages will be accounted for at the RC.

Therefore, we can estimate $NH$ for different LSR values and different node placements in the network, by estimating $\Delta_{\text{network}}$ and $dr$. Then, we can estimate the minimum number of active nodes in the network to meet the constraint $NH \geq 0$ for different LSR values.
3.3 PERFORMANCE EVALUATION

We evaluate a 12-hop (5 hops in the $k$-connected region and 7 levels in the backup region) wireless network of both the computation model results and simulation results. For the computation model, we analyse up to 3 lines of backup nodes to avoid long-running computation. For the simulation, we simulate up to 7 lines of backup nodes (49 backup nodes) for analysis purposes. We also assume the maximum connectivity degree of $k$-connected region is 4 (i.e., $k \leq 4$). Starting with one line of primary nodes and fixed $k$-connected region, we add lines of backup nodes from virtual roots to RC (i.e., add one node at a time in the first line of backup nodes from virtual roots to RC, add one node at a time in the second line of backup nodes, etc.). We evaluate three metrics: $DR$, $NH$ and the minimum number of nodes in the network for different LSRs with $NH \geq 0$.

The calculated DR at the RC is shown in Figure 5(a) for average LSR 0.8 (other values of LSR show the same trend). For a fixed $k$-connected region. We have three observations: (1) the inflection points happen when all primary relay nodes have the same number of backups (7 nodes or a complete line in this case); (2) while adding the first line of backup nodes, the DR exponentially increases due to the probability of sending messages from virtual roots to RC is $P_{\text{virtual} \rightarrow \text{RC}} = ((1 - p) \times p + p)^b \times p^{7-b}$, where $b$ is the number of backup nodes added in the first line of backup nodes. As $b$ increases, the $P_{\text{virtual} \rightarrow \text{RC}}$ increases exponentially; (3) the slope decreases when adding more lines of backups. This is because that the probability of using the last node in one level handling messages decreases as the number of backup nodes in each level increases. Figure 5(b) shows the calculated average network health for different LSRs. We can estimate the minimum number of nodes needed in the network, while meeting the constraint $NH \geq 0$.

We use the TOSSIM network simulator [Levis et al., 2003] with wireless traces from a 21-node subset of the WUSTL Testbed [tes, 2017] to do simulation analysis. Similar to [Li et al., 2015], we use controlled Received Signal Strength (RSSI) with uniform gaps to simulate various LSR values (Table 1 shows the average LSR of each RSSI value). DR is shown in Figure 5(c) for different RSSI values. The DR increases when the number of nodes increases, showing network gains in reliability. Obviously, the higher the RSSI, the higher the DR;
Figure 5: Computation model results and simulation results

however, the difference decreases as a function of the number of nodes and RSSI becomes irrelevant for networks with many nodes. \( NH \) is shown in Figure 5(d). \( NH \) increases at first, because DR increases faster than the network delay. But \( NH \) then decreases, because the network delay increases faster than DR.

We compare the computation results (CMR) with the simulation results (SR). Table 1 is the comparison of the minimum number of nodes of CMR (MinCMR) and the minimum number of nodes of SR (MinSR) when satisfying PHX stability requirements \((NH \geq 0)\) for various values of RSSI. \( \text{Diff} \) is defined as the percentage difference between MinCMR and

(a) Computation model results: delivery ratio at remote controller; average LSR = 0.8

(b) Computation model results: network health with link success ratio (LSR) 0.7, 0.8 and 0.9.

(c) Simulation results: delivery ratio for different number of nodes.

(d) Simulation results: network health distribution for different number of nodes.
Table 1: Comparison of Model and Simulation results

<table>
<thead>
<tr>
<th>rss (dBm)</th>
<th>average LSR</th>
<th>LSR stdv</th>
<th>MinCMR</th>
<th>MinSR</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>-64</td>
<td>0.93</td>
<td>0.020</td>
<td>26</td>
<td>26</td>
<td>0%</td>
</tr>
<tr>
<td>-70</td>
<td>0.88</td>
<td>0.024</td>
<td>29</td>
<td>30</td>
<td>-3.4%</td>
</tr>
<tr>
<td>-76</td>
<td>0.82</td>
<td>0.031</td>
<td>33</td>
<td>32</td>
<td>3.0%</td>
</tr>
<tr>
<td>-82</td>
<td>0.77</td>
<td>0.035</td>
<td>37</td>
<td>39</td>
<td>-5.4%</td>
</tr>
<tr>
<td>-84</td>
<td>0.71</td>
<td>0.037</td>
<td>46</td>
<td>42</td>
<td>8.7%</td>
</tr>
</tbody>
</table>

MinSR, $\text{Diff} = (\text{MinCMR} - \text{MinSR})/\text{MinCMR}$, indicating how much difference between our computation model and the realistic network simulation. The simulation result verifies our computation model with average 4.1% difference from computation model result.

### 3.4 SUMMARY

In this work, we focus on designing a fault-tolerant wireless network to meet control system requirement with minimum energy consumption. We first propose a fault-tolerant network node placement design. We then present our computation model to estimate the control system constraint, $NH$ and estimate the minimum number of active nodes in the network to meet $NH \geq 0$. Finally, we simulate a 12-hop wireless network on TOSSIM simulator. The simulation result verifies our computation model with average 4.1% difference from computation model result.
4.0 NETWORK RECONFIGURATION: TIME-CORRELATED FAULTS (COMPLETED)

Network imperfections, network delays, and packet losses in wireless networks can degrade control system performance, which motivates us to find the optimal network configuration to minimize that impact. Another main difficulty of having wireless networks for the control systems is caused by interference and noise that produce time-varying fault patterns [Cerpa et al., 2005; Srinivasan et al., 2010], which motivates us to find a fast and effective way to carry out network reconfiguration at run time. In [Wang et al., 2017b,a], we design and implement a new framework with offline and online components to do network reconfiguration for the control system with time-varying network link failures. To see the interaction between network reconfiguration and the control system, we conduct a systematic case study with a WSN for a single PHX in NPP.

4.1 PROBLEM STATEMENT

In practice [Han et al., 2011], the control sampling period (i.e., the interval where the control loop makes decisions) in WCS is $2^n$ seconds, where $-2 \leq n \leq 9$ (250 ms to 8 minutes). For the network delay, there are two cases. (1) Network worst-case delay is less than the control sampling period: packet loss is the key effect on control system performance; The lower the packet loss, the better the control system performance. (2) The worst-case delay is more than the control sampling period: there is a trade-off between network delay and packet losses for control system performance. In this chapter, we focus on the latter case and assume that the network delay is greater than the control sampling period with time-correlated link failures.
Figure 6: Network reconfiguration framework for the control System with dynamic Interference

occur in the network. Our objective is to reduce the network-induced error into the control system (i.e., improve the control system performance) as the LSR changes over time.

4.2 PROPOSED SOLUTION

We propose a network reconfiguration framework that has as input a network configuration set (a network topology set in our case). Our network configuration set is generated by activating the different number of nodes in the network shown in Figure 4: activate paths from left to right in the $k$-connected region, and activate backup nodes from highest level to lowest level in relay region. We assume different topologies correspond to the different number of nodes in the network. Our framework contains two parts, offline and online, as shown in Figure 6. An optimal network configuration table is computed offline indexed by LSR values, which is stored in the remote controller side. 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 the configuration.
4.2.1 Offline computation

For the offline part, we quantify the trade-off between network delay and DR. We propose a model describing the network-induced imperfection impact on the control system performance. We then present an algorithm to find an optimal network configuration set by using this model.

We define , the total delay induced into the control system by the wireless network as

\[
\Delta = \left\lceil \frac{\Delta_{\text{network}} + n_{\text{loss}}\Delta_{\text{ssp}}}{\Delta_{\text{esp}}} \right\rceil \Delta_{\text{esp}}
\]  

(4.1)

where \(\Delta_{\text{network}}\) is the network end-to-end delay, \(\Delta_{\text{esp}}\) is the control sampling period, \(n_{\text{loss}}\) is the number of consecutive packet losses, \(\Delta_{\text{ssp}}\) is the sensing sampling period. \(n_{\text{loss}}\) is estimated by the expected value of the network loss ratio \((1 - DR)\) as \(n_{\text{loss}} = \sum_{i=1}^{n} i(1 - DR)^i, ((1 - DR)^i \geq t)\), where \((1 - DR)^i\) is the probability of \(i\) consecutive losses. When the probability is less than a threshold, \(t\), we assume that the probability can be ignored to avoid long-running computations. By applying the computation model proposed in Chapter 3 (\(\Delta_{\text{network}}\) and \(DR\) estimation), we are able to estimate the total induced delay \(\Delta\) for each network configuration given a certain LSR value.

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, \(\Delta\). Note that each network configuration has different \(\Delta\) values for different LSR values. We devise an algorithm that iterates through the given set of network configurations to find a set of estimated optimal network configurations for each average LSR value and stores them in a look-up table \(T\) indexed by LSR values.

4.2.2 Online reconfiguration

For the online part, network reconfiguration is based on the offline look-up table given the current LSR. We first present an algorithm to estimate LSR at run time. We then propose six centralized network reconfiguration algorithms running in the remote controller, that is, three original algorithms combined with and without considering consecutive packet losses.
The idea of LSR estimation algorithm is that during the LSR estimation interval (LSRI), every node calculates its average receiving LSR (the ratio of the number of messages it receives and the number of children it has). At the end of every LSRI, each node averages the LSRs from its children and its own receiving LSR and sends out the message with the calculated average LSR to its parents (one or more parent nodes). Eventually, the remote controller computes the final overall network average LSR during the LSRI.

The online algorithm finds 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 frequency of changing network topology in the above three algorithms is LSRI. When considering consecutive losses (CL), we have another three algorithms, CL-DO, CL-MICD and CL-AC. All these algorithms have access to the offline look-up table $T$.

For DO, 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. Given that a topology corresponds to different number of relay nodes, we propose MICD, inspired by [Sankarasubramaniam et al., 2003], 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. 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). Inspired by adaptive control theory [Hovakimyan and Cao, 2010], in AC, $\text{curr}_{node} = \alpha \times \text{curr}_{node} + (1 - \alpha) \times \text{est}_{node}$, where $\alpha$ is a parameter that guides the speed of addition and reduction of nodes in the network ($0 < \alpha < 1$), $\text{curr}_{node}$ is the current number of nodes in the network and $\text{est}_{node}$ is the estimated number of optimal nodes from offline analysis. In essence, the smaller the $\alpha$, the higher the speed to change the current number of nodes. From Equation 4.1, the total induced delay is proportional to the number of consecutive losses $n_{loss}$. 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 can degrade the control system.
performance. As a first experimental step, whenever there are more than three consecutive message losses, we add \( k (k = 3) \) more nodes in the network. When considering consecutive losses, we devise three more online algorithms: CL-DO, CL-MICD and CL-AC, which are orthogonal to the three algorithms above.

4.3 PERFORMANCE EVALUATION

We combine the implementations of the framework with a state-of-the-art cyber-physical system simulator (WCPS 2.0 [Li et al., 2015]) and allow the wireless network to run together with the PHX Simulink model. We deploy a 12-hop and up to 50 nodes (9 sensors and 41 relay nodes) in the NPP, as shown in Figure 4. We evaluate five metrics: average RMSE (MW)\(^1\), network lifetime (days), the number of nodes over time, total induced delay (s) over time and RMSE (MW) over time.

![Graphs showing performance metrics](image)

(a) Offline optimal network configuration for different LSR values. (b) Total induced delay result for different \( \text{rssi} \) values. (c) Power output RMSE (in MW) for different \( \text{rssi} \) values.

Figure 7: Offline results

For the offline results, we plot the look-up table containing the estimated optimal configuration set for each LSR value in Figure 7(a). The higher the LSR, the more robust the network, and therefore the fewer sensor nodes needed. We run the simulation with static

\(^1\)The metric measures the network-induced error, RMS error between the closed-loop responses using wired control (i.e., we assume there are no packet drops and no network delay in wired control) and wireless control.
Figure 8: Comparison of estimated and real LSR values (average RSSI = -82 dBm)

RSSI values, to correlate the network imperfection model (shown in Figure 7(b)) and the control system performance (shown in Figure 7(c)). Comparing Figures 7(b) and 7(c), 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, which shows that our model is accurate.

For the online results, to simulate time-correlated faults, we adjust each relay node’s RSSI value [Lee et al., 2007] to change LSR. We use real-world noise traces [Li et al., 2015] and change the range of link quality over time. $rssi$ duration is defined as the time interval at which the RSSI value is fixed. We randomly choose RSSI value from $rssi$ range (-60dBm, -85dBm) and randomly choose $rssi$ duration from $time$ range (0, 20s) both with a uniform distribution. We present the sensitivity analysis of LSRI and AC algorithm performance with different $\alpha$ values.

**Sensitivity Analysis of LSRI** Since LSR is estimated periodically, the length of LSRI will affect the control system performance. Figure 9(a) shows the average power output RMSE for different LSRI values. “static” is when the number of nodes in the network is fixed. We test the number of nodes 20 to 50 and choose the static scheme with minimum RMSE among these tests. Even the best static scheme 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 perform better. When the LSRI increases, the RMSEs of the schemes DO, MICD and AC increase because
Figure 9: Sensitivity analysis of LSRI (the average RSSI value is -82 dBm; AC and CL-AC schemes with $\alpha = 0.1$)

the estimation is less accurate at high LSRI values. In Figure 8, the yellow line is the real LSR; the black line (LSRI of 2s) tracks the real LSR better than the LSRI of 8s and 16s. Therefore, the control system performance shows less error when the LSR estimation is accurate. But RMSE metrics for the CL-* schemes (CL-DO, CL-MICD and CL-AC) are not affected by the LSRI values because even though the LSR estimation is not accurate, CL-* schemes add additional nodes to make the network robust. However, the side-effect is that CL-* schemes consume more energy. Figure 9(b) shows the network lifetime, which is directly influenced by energy consumption.

**Sensitivity Analysis of $\alpha$ values** Recall that the AC scheme has a variable $\alpha$
Figure 10: Power output RMSE result comparison of AC and CL-AC for different $\alpha$ values. (The average RSSI value is -82 dBm and the LSR interval is 2s.)

$\alpha < 1$), which determines the speed to add or reduce nodes in the network (small $\alpha$, fast node adding). Figure 10 shows the RMSEs of AC and CL-AC schemes for different $\alpha$ 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 11 shows the reason more clearly (see AC ($\alpha=0.9$) and AC ($\alpha=0.1$)). From Figure 10, we also see that CL-AC always performs better than AC. Although the speed to add or reduce nodes is slow for $\alpha 0.9$, considering consecutive losses can compensate when the network has more interference. Figure 11 shows more details (see AC ($\alpha=0.9$) and CL-AC ($\alpha=0.9$)).

4.4 SUMMARY

In this work, we design and implement a framework with offline and online parts. A systematic case study is conducted to see the in-depth interaction between network reconfiguration and the control. The simulation results show that the network imperfection model is accurate with Pearson correlation 0.993, that network reconfiguration works better than the static scheme showing low error and longer network lifetime. We find that consecutive message losses can degrade the control system performance.
Figure 11: (a) Average number of nodes in the network, (b) average induced delay and (c) average RMSE over time for AC and CL-AC (LSRI: 2s; Average RSSI: -82 dBm).
5.0 NETWORK RECONFIGURATION: SPACE-CORRELATED FAULTS (FUTURE)

5.1 MOTIVATION

Interference is one of the major factors of link unreliability in WSN [Baccour et al., 2012]. Interference sources (ISs) are generally considered as cordless phones, WiFi, Bluetooth, microwave ovens, radio jammer, walkie-talkies and etc [Tang et al., 2007; Lapinsky and Easty, 2006]. In wireless control systems, ISs can also be equipment noise, electromagnetic interference, radio frequency interference (RFI) and fading [Low et al., 2005; Fadel et al., 2015; Chiweve et al., 2015]. Many empirical studies of low power wireless links have been conducted with various ISs [Lin et al., 2009; Tang et al., 2007; Hackmann et al., 2008; Guo et al., 2012; Srinivasan et al., 2010], showing temporal and spatial characteristics of wireless link qualities. The degree of interference is related to the distance to ISs: the longer distance, the less interference [Lapinsky and Easty, 2006; Xu, 2007; Wei et al., 2016; Xu et al., 2006]. Spatial link failures caused by ISs in WCS affect the packet delivery and can even make the network disconnected, thus severely degrading the control system performance. Therefore, the adaptive strategy is necessary for WCS to deal with space-correlated link failures, which has not been researched in WCS before.

5.2 PROBLEM STATEMENT AND PROPOSED WORK

We will focus on the space-correlated link failures with one mobile IS. The IS moves with a certain speed, as shown in Figure 12. The links that suffer interference by the IS (red links)
Figure 12: A moving object causes space-correlated link failures in WSN (red links has lower quality caused by the moving object)

will have lower quality to do data transmission than the other links (black links). This kind of space-correlated fault model has not been proposed before, but it is a common problem in WCSs [Baccour et al., 2012]. Our objective is to reduce the network-induced error to the control system by detecting the IS and tolerating dynamic link failures.

As shown in Figure 4, we place backup nodes in the relay region as close as possible to minimize the number of relay nodes [Wang et al., 2016]. However, with space-correlated link failures, that node placement strategy should be changed, because links clustering together are vulnerable, where IS is nearby. Instead, the backup nodes in each level should be placed further from the other nodes, but at a distance that they can still listen to each other. The assumption in Chapter 3 and Chapter 4 that side links never fail is no longer valid when considering space-correlated link failures. The bitvector protocol [Wang et al., 2015] deals with side link failures using bit vector, which contains 4 bits for each neighbor, to estimate the link quality of its neighbors. The bit vector is appended to the message sent by the node, in addition to the measurement value. Each node dynamically chooses its primary parent with highest link quality. But the protocol is used for environment monitoring in WSNs and has not been used to do data transmission in WCS. For our future works, we propose to study a network reconfiguration algorithm in a distributed way. If there is no parent with high link quality, extra backup nodes should be activated. Instead of the centralized network reconfiguration algorithms running in the remote controller in Chapter 4, the primary node in each level will decide how many nodes to be activated or deactivated to save network bandwidth while keeping robustness.
To complete this work, we propose three future tasks:

(1) Create a space-correlated fault model with one moving IS (Figure 12). The IS moves with a certain speed and determines at run time which links fail with what probability: links closer to the IS will fail with higher probability than the other links.

(2) Implement bitvector protocol on WCPS [Li et al., 2015] and extend it with a distributed network reconfiguration algorithm.

(3) Conduct a case study in NPP with a single PHX and evaluate the results, to show that the network reconfiguration can achieve significant reduction in power output RMSE compared to the baseline of the work in Chapter 4.
6.0 REAL-TIME NETWORK SCHEDULING (FUTURE)

6.1 MOTIVATION

A new wave of NPPs considers several Small Modular Reactors (SMRs) [Greene et al., 2010], instead of a single large reactor, due to the flexibility and cost-benefit of starting and stopping SMRs. Figure 13 shows an NPP with three PHXs (each PHX in one SMR), each of which transmits measurement data via a shared wireless network (in total, a maximum of 9 measurements are sent periodically). Figure 14 shows 8 different ideal scenarios (each is associated with a power reference function, decided by the plant operator) of a PHX. For example, ramp30 is supposed to reduce the power from 42MW to 32MW within 30s; we call the power change time (30s) the power change duration (PCD), and the total amount of power change (10 MW) as power change amount (PCA). 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 (ie, change both PCD and PCA), based on energy requirements and balancing the power required to achieve a certain level of power output.

In our prelim work, we noticed that network delay, PCD and PCA have a significant influence on the power RMSE of NPPs, as follows. We have three observations:

1) In Figure 15(a), we show the effect of network delays and PCD of reference functions on a single PHX system with $DR$ of 0.9 (similar results for other DRs). The control system performance is related to the PCDs of reference functions. When PCA is the same, the shorter the PCD, the higher the RMSE. For reference functions with shorter PCDs, the network delay becomes a more significant factor on the control system performance. For example, the control system performance of ramp15 with delay=0.2s is the same as the
performance of ramp45 with delay=0.4s. However, when PCD is greater than 60s, the control system performs similarly.

(2) Figure 15(b) shows the effect of network delays and DRs comparison (similar results for the other reference functions). We observe that the system is very delay-sensitive. Delay has more significant effect on control system performance than packet losses.

(3) Figure 15(c) shows the PCAs of different reference functions effect for control system performance with PCD of 30s (similar results for other PCDs). The control system performance is related to the PCA, besides the PCD (observation (1)). When PCD is the same, the higher PCA, the more RMSE.

Motivated by the above three observations, for the delay-sensitive system, we will impose a smaller network delay (in essence a deadline) for the more urgent application demand (e.g., ramp15, or more aggressive PCA) and a more laxed deadline for the less urgent applications (e.g., ramp45, or less aggressive PCA). We plan to dynamically schedule the network flows, based on the application layer demand. We call this kind of dynamic scheduling cross-layer real-time scheduling, given that it takes application behaviors into account and changes deadlines to influence packet scheduling at the network layer. To the best of our knowledge, cross-layer network scheduling has not been studied in time-constraint WCS before.
6.2 PROBLEM STATEMENT AND PROPOSED WORK

The inputs of our problem are the reference functions of the physical systems. In this thesis, we focus on the reference function with multiple ramps. Different PSs may have different reference functions in terms of PCAs and PCDs. Similar to [Saifullah et al., 2010], we define a set of $m$ end-to-end network flows as $F = \{F_1, F_2, \ldots, F_m\}$. Each network flow delivers one
measurement to the remote controller. Each flow $F_i$ associated with a source $s_i$, a destination $d_i$, a period $p_i$, and a deadline $D_i$. The network flow set is divided into two subsets, a critical flow set, and a non-critical flow set. In our case, for example (see Figure 15(a) and Figure 15(c)), the flows with PCD greater than 60s and PCA more than 4 MW are critical flows.

Our initial proposed objective is to reduce the total network-induced error, $error = \sum_{i=1}^{n} r_i$, where $n$ is the total number of physical systems controlled over one wireless network, and $r_i$ is power output RMSE for physical system $i$. Each physical system has its own reference function, depending on its application demand.

To achieve our objective, we propose three future tasks:

1. Define the deadline for each network flow, according to the offline control system analysis.

2. Devise a real-time scheduling algorithm to schedule network flows dynamically to reduce RMSE of the set of control systems.

3. Conduct a case study in NPP with three PHXs and evaluate the results on WCPS [Li et al., 2015].
7.0 SUMMARY

Wireless control systems are gaining rapid adoption in process industries because of its advantages in lowering deployment and maintenance cost in challenging environments. While early success of industrial WSN has been recognized, significant potentials remain in exploring WCS as a unified system to address control system stability and performance issues.

In this thesis, we first propose a fault-tolerant network design and a computation model to quantify the control system stability requirement with awareness of network energy consumption. We then propose a network reconfiguration framework to improve the control system performance for time-correlated link failures. We will explore the network reconfiguration techniques for space-correlated link failures and real-time scheduling technique to enhance the control system performance. Case studies will be used to test the viability of the proposed ideas. The proposed work will be completed following the timeline, shown in Table 2.
Table 2: Timeline of Proposed Work.

<table>
<thead>
<tr>
<th>Date</th>
<th>Content</th>
<th>Deliverable results</th>
</tr>
</thead>
<tbody>
<tr>
<td>May. - Aug. 2017</td>
<td>Dynamic network flow scheduling algorithm design and implementation on WCPS</td>
<td>Network deadline formulation and a WCS with the function of dynamic network flow scheduling</td>
</tr>
<tr>
<td>Sep. - Dec. 2017</td>
<td>Measure the performance of WCS with dynamic network flow scheduling</td>
<td>A paper for publication</td>
</tr>
<tr>
<td>Jan. - Feb. 2018</td>
<td>Finish the implementation of bitvector protocol and space-correlated fault model on WCPS</td>
<td>A WCS with a fault-tolerance protocol to deal with space correlated link failures</td>
</tr>
<tr>
<td>March. 2018</td>
<td>Come up with a network reconfiguration algorithm and implement it on WCPS</td>
<td>A WCS with the function of network reconfiguration for space-correlated link failures</td>
</tr>
<tr>
<td>April. 2018</td>
<td>Measure the performance of WCS with network reconfiguration mechanism</td>
<td>A paper for publication</td>
</tr>
</tbody>
</table>


Jusuf, Farid and Joelianto Endra. Stabilization of networked control system with time delay induced by network imperfections. In *ICCSII 2012*.


34


Pant, Yash Vardhan, Abbas Houssam, Mohta Kartik, Nghiem Truong X, Devietti Joseph, and Mangharam


