Work-in-Progress: Wireless Network Reconfiguration for Control Systems

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. To solve these two issues, we propose a network reconfiguration framework with offline and online components that considers time-correlated link failures. We are conducting a case study to wirelessly control a non-linear primary heat exchanger system in a small modular nuclear reactor of a nuclear power plant.**

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 [4], [6], [8], [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 from the control room to sensors/actuators, and packet losses can induce serious errors in the control system, which is very undesirable (e.g., in a nuclear power plant, there could be loss of efficiency or danger of a meltdown). These network impacts, such as packet loss and network delay, have been called *network-induced imperfections* [13] and categorized into five types: (a) time delays, (b) packet losses and disorder, (c) time-varying packet transmission/sampling periods, (d) competition of multiple nodes accessing networks and (e) data quantization. The first two types gain the most attentions, which are also our focus in this paper.

In practice [4], the *control sampling period* (i.e., the interval where the control loop makes decisions) in cyber-physical systems (CPS) is 2^n seconds, where $-2 \leq n \leq 9$ (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; (2) The worst-case delay is more than the control sampling period. For the former, the *delivery ratio* (DR, defined as the ratio of arrived messages to 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 induces more delay for messages to be delivered to the control room.

However, when the worst-case delay is larger than the control sampling period, there is a trade-off between network delay and DR for control system performance. Although the network-induced imperfection problem has been studied from the control perspective [7], [9], minimizing the network-induced imperfections has not been studied from the network perspective. The trade-off between losses (requiring more redundant nodes) and delays (calling for fewer nodes) can be achieved through an optimization to find the network configuration with minimum imperfections, solved off-line.

However, control system environments typically changes with time and space [1], [3], such as obstacles, moving people/objects, and electromagnetic and radio frequency interference (EMI/RFI). Some interference can make the network inaccessible and disconnected for a limited amount of time (e.g., a factory robot, blocks the wireless transmission), and the control system cannot get measurements during that time, which in turn severely impacts the control system performance. Therefore, we propose an online network reconfiguration to guarantee the control system performance.

Integration and co-design of network and control system are effective for wireless control system, as described next. A co-design of network topology conditions and control system stability is explored in [8]. The integration of fault-tolerant wireless network and control in nuclear power plants is studied in [12]. In [6], 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 wireless control system, since they assume the network delay is less than the control sampling period. In [2], the authors derive 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 and do not consider the interaction between network reconfiguration and control, which is our focus in this paper. We show how the network reconfiguration changes the control system

978-1-5090-5269-1/17/$31.00

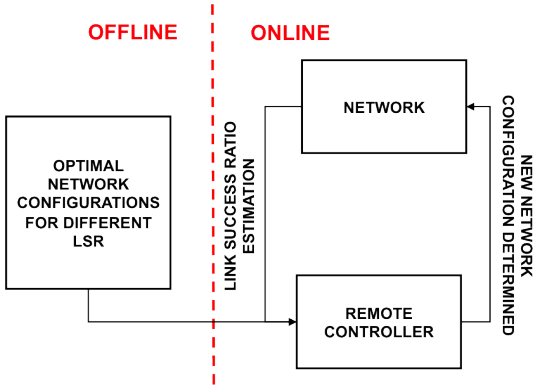

Figure 1: Framework: network reconfiguration for control System with dynamic Interference

performance, comparing with a state-of-the-art static network configuration.

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 system sampling period. We only consider link failures between network nodes. We define average *link success ratio* (LSR) as the probability a message can be sent out successfully on that link and use it 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 DR to the *total induced delay* on the control system. We estimate the total induced delay for a set of network configurations, including network topology. 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. The control node is also part of the colsed-loop control and sends control signals back via wireless network. We study the interaction between network reconfiguration and the control system with a simulator that controls the wireless sensor network and a non-linear primary heat exchanger system [12] in a small modular nuclear reactor (SMR) in a nuclear power plant. Note that safety issues are beyond the scope of this paper; we focus on feasibility first.

II. Offline Optimal Network Configuration

In this section, we propose the first model to quantify the network imperfections in terms of network delay and DR, considering the control system performance. We then show how to find an optimal network configuration set (there might be more than one optimal configuration) with this model.

Similar to [6], we have two assumptions: (1) when a message is not received by the controller, the previously-received value will be used. (2) sensors do not fail, and produce real values (i.e., no noise). Under these assumptions,

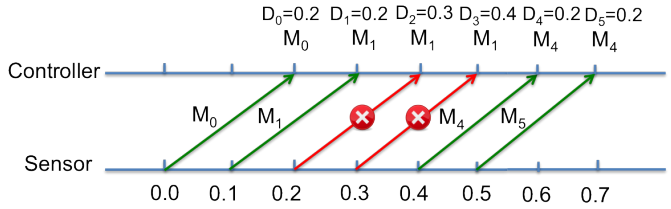

Figure 2: Network delay and delivery ratio trade off illustration example, when network delay is greater than control system sampling period.

the model and the algorithm to determine the optimal network configuration set are general and can be used in any WCS.

A. Network Imperfection Model

We define D, 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

$$D = \left\lceil \frac{D_{network} + n_{loss}\Delta_{ssp}}{\Delta_{csp}} \right\rceil \Delta_{csp} \quad (1)$$

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 system sampling period and sensor 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 $D_{network}$ and n_{loss}, which is a function of the packet DR (DR estimation and end-to-end worst-case delay estimation have been studied elsewhere [12], [10]). n_{loss} is estimated by the expected value of the network loss ratio (1-DR). We assume message loss follows the uniform distribution, since DR can be viewed as the probability a message received by the controller. Note that we only estimate n_{loss} as uniform distribution and will address bursty losses problem in Section III-C. 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 Algorithm

Our offline algorithm discovers the set of network configurations for the online portion. Since our goal is to minimize the network impact (network delay and packet loss) to the control system, an optimal network configuration means configuration with minimum induced delay, D, considering the network imperfection model in Section II-A. Note that the bigger the LSR value is, the bigger the DR is. Therefore, each LSR value corresponds to a DR. 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 (See Algorithm 1).

Initialization: Configuration Set C; all possible LSR set LSR; Look-up table T; 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**
      $D_{min} = D_j$;
      $conf_{opt} = conf_j$
  T.insert(lsr_i, $conf_{opt}$)

Algorithm 1: The optimal network configuration set determination algorithm

III. 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, the nodes are time synchronized and messages are sent periodically every sensing sampling period. 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 is general, in this paper we restrict network configuration to refer to the number of sensor nodes. 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 system 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. We first propose an algorithm to estimate LSR at run time. We then propose six online network reconfiguration algorithms, that is, three original algorithms (Section III-B) combined with and without taking into account consecutive packet losses (Section III-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 LSR 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, and parents will average their own LSR with their children's LSRs; this repeats until it gets to the controller.

Specifically, 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 an LSRI, each node concatenates its average receiving LSR with its own message, and sends the message to its parent nodes (one or more parent nodes) and itself to its parents. Eventually, the remote controller will compute the final overall network average LSR.

B. Online Network Reconfiguration algorithms

We explore three options to reach the optimal configuration, given that the optimality depends on the LSR, which cannot be computed instantaneously (estimated from the last average LSR during LSRI). The algorithms are DirectJump to Optimal (DO), Multiplicative Increase and Conservative Decrease (MICD), and Adaptive Control (AC). These algorithms have access to the offline look-up table T.

1) DO: DirectJump to Optimum Algorithm: In 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 (see Algorithm 2).

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;
    LSRCounter++;

Algorithm 2: Direct jump to optimum (DO)

2) MICD: Multiplicative Increase and Conservative Decrease Algorithm: Given that a topology corresponds to different number of relay nodes, we were inspired by [11] 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).

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;
    change network topology to be $curr_{node}$;
    LSRCounter=0;
  LSRCounter++;

Algorithm 3: Multiplicative Increase and Conservative Decrease (MICD)

3) AC: Adaptive Control Algorithm: Inspired by adaptive control theory [5], AC attempts to adjust the number of nodes to the network conditions: 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.

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}=$T(curr_{LSR}).node$;
    $curr_{node} = \alpha \times curr_{node} + (1 - \alpha) \times est_{node}$;
    change network topology to be $curr_{node}$;
    LSRCounter=0;
  LSRCounter++;

Algorithm 4: Adaptive Control (AC)

C. Reconfiguration Considering Consecutive Message Losses

From Equation 1, the total induced delay are proportional to number of consecutive losses n_{loss}. Therefore, orthogonal to the algorithms in Section III-B, we consider n_{loss}. Since the LSR estimation is inaccurate and message loss is assumed as uniform distribution, 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. By considering consecutive losses, we end up with three more online algorithms: CL-DO, CL-MICD, and and CL-AC.

IV. Conclusion and Future Work

We explore the interaction between online network reconfiguration and control, when the worst-case network delay is bigger than the control system sampling period. We propose a network imperfection model and network reliability estimation offline; we propose six online network topology control algorithms, with and without considering consecutive loss (three each). Next we will study a mechanism for tolerating loss of the control messages and create a case study for nonlinear heat exchanger system in nuclear reactors controlled by wireless network.

References

[1] 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 Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.

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

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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*. ACM, 2003.

[12] 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)*. IEEE, 2016.

[13] 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.