Design of a Reliable Communication System for Grid-Style Traffic Control Networks

Junghoon Lee
Dept. of Computer Science and Statistics
Jeju National University
Jeju-Do, Rep. of Korea, 690-756
jhlee@jejunu.ac.kr

Song Han, Aloysius K. Mok
Dept. of Computer Science
University of Texas
Austin, TX 78712, USA
{shan, mok}@cs.utexas.edu

Abstract

This paper designs and analyzes the performance of a reliable communication scheme for the traffic control system built upon a wireless process control protocol, aiming at enhancing the robustness and timeliness of the safety-critical control applications. Focusing on the slot-based predictable access and the grid topology of urban area road networks, the proposed scheme establishes one primary and secondary route from the controller to each other node and allocates the time slots accordingly. A split-merge operation makes a sender node sense the primary channel (channel to the primary receiver) and take the secondary channel if the first one is not clear in a single slot, while making the receiver first listen to the primary sender and switches to the secondary sender. In addition, for the sake of finding the path having the lowest error rate, the split-merge operation is modeled as a single virtual link to exploit the shortest path algorithm. We evaluate the performance of our scheme by simulation using a discrete event scheduler. The experimental results show that the proposed scheme enhances the transmission success ratio for the 4 × 4 grid by up to 7.9 %, much more improving the ratio for long-hop nodes. In addition, the routing scheme can find the path that can improve the delivery ratio of control messages by up to 6.69 % for the given performance parameters, compared with the normal grid routing scheme.

1 Introduction

[1] The industry process control system can be abstractly modeled as a feedback loop consisting of a controlled process, a controller, sensors, and actuators[23]. Sensors collect the current state of the controlled process and report to the controller, while the controller responds to the state change captured by the sensors after computing the actions required to reduce the disparity between the current and target states of the system. Actuators perform the control action delivered by the controller. For a timely reaction, the control decision must be computed precisely in time and the communication between each component must be completed within the given time bound. Many communication systems capable of meeting such time constraints have been developed and deployed into diverse field systems. Among these, HART (Highway Addressable Remote Transducer) is one of the most outstanding protocols for process control[2]. Its master/slave mechanism provides timely and predictable communication.

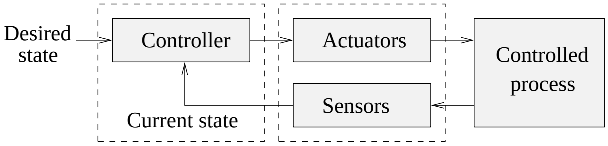


Figure 1. Process control architecture

In addition to the wireline counterpart, wireless process control has been of great interest due to its potential for cost reduction, easy maintenance, and enhanced flexibility. Based on the proven and widely accepted (wireline) HART technology, the WirelessHART standard provides a robust wireless protocol for distributed process control applications[3]. Its key capabilities lie in reliability, security, and efficient power management. Especially, to the end of overcoming the transient instability of wireless channels, it puts a special emphasis on reliability by mesh networking, channel hopping, and time-synchronized messaging. This protocol has been implemented and is about to be released to the market[21]. However, it is impossible to completely

---

[1] This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency. (NIPA-2009-C1090-0902-0040))

remove communication losses in wireless networks, while the loss leads to the delay of control activation and increases the reaction cost. Therefore, the reliability enhancement scheme must be continuously investigated and integrated, considering the given system goal and the underlying network architecture.

In process control, the control loop is executed periodically at a rate fast enough to react to external stimulus. Accordingly, messages have their own deadlines so as not to delay or jeopardize the reaction decision procedure. Generally, the error control procedure brings additional messages, extending the execution of a control loop. This leads to a conflict between the two system goals, namely, reliability and timeliness. According to the WirelessHART standard, before the transmission of a message in each time slot, the sender performs CCA (Clear Channel Assessment) to detect the existence of other transmission or noise[19]. If the channel is not clear, the transmission is just cancelled, unavoidably resulting in the waste of a slot. However, a modification of this operation can save the slot waste in some cases. Namely, a sender can attempt another channel to take an alternative route in a single time slot, not extending the control loop, if the receiver is able to select one of two possible channels also in a single slot.

This idea cannot work for all wireless network topologies[16]. For the idea to be better applicable, both alternative and primary paths have to have the sufficient number of nodes in common and also have the same length. If their lengths are different, some time slots must be wasted to wait for the possible packet transmission from the longer path. Grid networks have many unique features including the fact that many equal-length paths can be found very easily, irrespective of whether they are link-disjoint or not. Even though this topology may look very rare in real life, we can find in some situations. For example, consider a road network in urban areas and each crossing has a traffic light. Each traffic light can play the role of a wireless node when it is equipped with sensor devices and actuators along with a wireless communication interface. Each node can provide global network access to the moving vehicles, while the network can take advantage of a wireless process control system and exploit the above-mentioned idea. In this regard, this paper is to design and measure the performance of a reliable traffic control system that can mask channel error without extending the control loop based on the wireless process control protocol, specifically, WirelessHART.

This paper is organized as follows: After outlining the problem in Section 1, Section 2 describes the background of this paper and the related works. Section 3 proposes a channel switch scheme on the grid-style traffic control system, while Section 4 describes how to take into account the channel switch in deciding the route from and to the controller. After the performance measurement results are demonstrated in Section 5, Section 6 summarizes and concludes this paper with a discussion of future work.

2 Background and related work

2.1 WirelessHART

Just like Zigbee[11] and Bluetooth[18], WirelessHART operates on the IEEE 802.15.4 GHz radioband[2]. It includes 16 frequency channels spaced by 5 MHz guard band. The data link layer provides time-division access on top of the time synchronization process, which is carried out continuously during the whole operation time[12]. Its time axis is divided into 10 ms time slots and a group of consecutive slots is defined to be a superframe. To meet the robustness requirement of industrial applications, WirelessHART makes a central network manager determine routing and communication schedules. This schedule can be propagated to the field devices in priori of system operation or updated via the channel reserved for this purpose. The manager also decides the path between source and destination, and assigns each (sender, receiver) pair to a time slot. The network operation of each node is time-driven, so a device wakes up by the timer interrupt on its sending and receiving slots[8].

Figure 2 shows the detailed description of a time slot, which has several time intervals including CCA (Clear Channel Assessment)[2]. Due to the half-duplex nature of current wireless systems, collision detection can't be applicable. Instead, automatic CCA before each transmission and channel blacklisting can be used to avoid a specific interference area and also to minimize interference to others. The 802.15.4 standard specifies that CCA may be performed using energy detection, preamble detection, or the combination of the two. In addition, there are reliable channel estimation methods available for the MAC layer to obviate erroneous communication over the unclear channel. If the channel is detected not to be clear, the sender does not proceed or try another recovery action, detailed action not being defined in the standard. Channel probing can be accomplished by CCA, RTS/CTS, and so on[14]. Besides, WirelessHART has many features useful for process control such as node join and leave operations in addition to time synchronization. For detailed description, readers can refer to [21].

2.2 Channel schedule

Communication on grid topology is researched in the wireless mesh network, and its main concern is routing the message inside the grid[13]. The network has multiple design goals to achieve, such as minimizing end-to-end delays, ensuring fairness among nodes, and minimizing power consumption[15]. The channel assignment problem

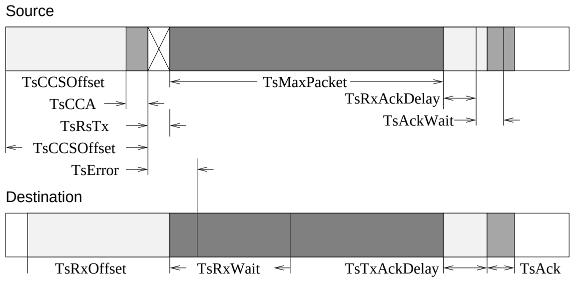


Figure 2. WirelessHART time slot

is NP-hard[20], so even for the case of no interference in the network, the optimal schedule cannot be computed in polynomial time. The greedy approach is most general for static channel assignment, exploiting the well-known coloring algorithm[6]. That is, the coloring or time slots must be assigned such that no two 2-hop nodes have the same color. For slot-based access, most schemes focus on how to assign a color on the diverse topology and how to cope with the topology change, some handling local route decision when just partial topology information is available. Anyway, existing schemes have not dealt with the dynamic channel selection within a single slot and the subsidiary slot allocation.

N. Chang considered transmission scheduling based on optimal channel probing in a multichannel system, where each channel state is associated with a probability of transmission success and the sender just has partial information[7]. To compensate for the overhead and resource consumption of the channel probing procedure, this work proposed a strategy which decided the channel to probe based on a statistical channel model. Using this estimation, the sender or scheduler can pick the channel to use for transmission. Even though the concept of channel probing is very prospective, this scheme focuses just on the dynamic operation of communication procedure, so it is not suitable for process control. Moreover, it lacks predictability, as it does not consider the route length but just the probabilistic estimation on successful transmissions in the selection of an available channel.

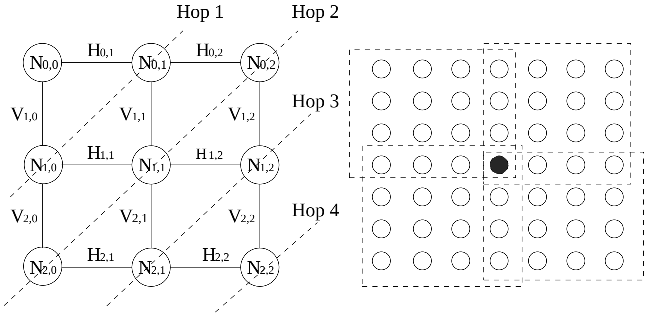
A slot management on WirelessHART was considered along with a mathematical framework in terms of modeling and analysis of multi-hop communication networks[5]. This model allows us to analyze the effect of scheduling, routing, control decision, and network topology. In this model, each node has at least two neighbor choices to route a packet for any destination nodes. Hence, the time slot schedule must explicitly have two paths for each source and destination pair. That is, regardless of whether the first path successfully transmits a message, the secondary route redundantly delivers the same message. This scheme can integrate any style of alternative route such as a node-disjoint or link-disjoint path, but bandwidth waste is unavoidable and slot allocation can get too complex.

Lee et al. proposed a channel selection scheme based on the CCA result in the preliminary version of this paper[16]. The main idea of this paper is inherited from [16]. However, it restrictively assumed that the slot rate of each link is the same and the split-merge operation, which will be described later, is simply placed to the rectangle closer to the destination. As contrast, this paper generalizes to the case that each channel has its own slot error rate and addresses how to, or even whether to, apply the channel switch operation in each control message path.

3 Channel scheduling scheme

3.1 Traffic control network

Downtowns of most modern cities have a Manhattan-like road network, at which some researches on vehicular networks are targeting[9]. Accordingly, the traffic control network, where wireless nodes are placed at each crossing of such road network, has a grid topology as shown in Figure 3 (a). Each node, generally having 4 neighbors, can exchange messages directly with its neighbors which reside in a vertical or horizontal position. Two nodes in the diagonal of a rectangle cannot generally reach each other directly, as obstacles like a tall building are highly likely to block the wireless transmission. In this network, one node can take the control of the network, and we assume that it is located at the fringe of a rectangle, for this architecture makes the communication schedule simple and systematic. In Figure 3 (a), $N_{0,0}$ is the controller node. When the controller node is located in the arbitrary point within the grid, 4 quadrants can be made as shown in Figure 3 (b), each of which having the controller node at one corner[16]. Then 4 quadrants can be mapped or transformed to the network shown in Figure 3 (a) by making the controller be placed at the left top corner.



(a) 3 * 3 topology (b) grid partition

Figure 3. Traffic control network

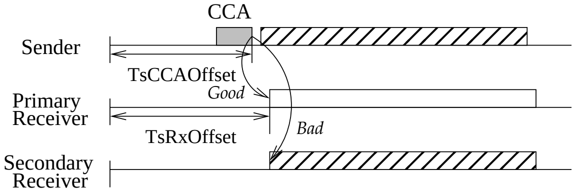
For process control, device-to-device communication is carried out in a pre-scheduled time window. This static schedule provides very reliable, power-efficient, and scalable message delivery, promoting interoperability and ease of use. A control loop consists of three phases, namely, reading state variables from sensors, deciding the control action, and sending the value of control variables to the actuators. The first and third steps involve the network and have their own communication schedules, which can be overlapped via the different channel hopping sequences. After all, in a control loop, the communication schedule includes the sequential transmissions from all the controller to all nodes and vice versa. As they are symmetric, we will just focus on the downlink case. In some WirelessHART implementation, the network manger traverses a communication graph or a grid by the breadth-first search and allocates slots according to this order, appropriately inserting the high priority management slots such as keep-alive, join, and other messages[21]. The schedule also contains the hop sequence by which the channel is selected in each slot boundary. This paper does not consider simultaneous transmissions of multiple nodes which have no interference, as schedule optimization belongs to another research domain[10].

Control decision is made both locally in each node and centrally by the controller node. Local operation enables each node to work autonomously to guarantee a ceaseless operation even in the case of network disconnection, sometimes tolerating performance degradation. The central controller decides an appropriate action for the traffic light and other possible actuators according to the embedded program logic and the value of process variables including wide area traffic condition, accident information, vehicle speed level, and so on. It should send messages containing the value of control variables to all nodes in a control loop. The system does not have to concern about power consumption, as it can share stable and sufficient power provisioned to the traffic light[1]. Moreover, the radios at each node are capable of fast switching between channels with a very small latency.

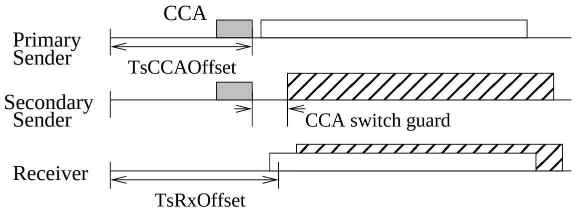
3.2 Main idea

In WirelessHART, if the channel status is not good, the sender discards the transmission, wasting the slot time and making the subsequent transmission schedule complex. In case the sender can attempt on the alternative route in the same slot, the slot waste can be avoided. To enhance reliability, an alternative path is indispensable. If the two paths have the same length, a further improvement can be expected as will be explained later and the slot assignment will be very efficient. The main idea can be described by an example network of Figure 3 (a). At now, we assume that every node has a common error rate, but this will be removed at the next section. In this figure, consider the transmission from $N_{0,0}$ to $N_{1,1}$ which has two 2-hop paths, namely, $N_{0,0} \rightarrow N_{1,0} \rightarrow N_{1,1}$, and $N_{0,0} \rightarrow N_{0,1} \rightarrow N_{1,1}$. If the network manager selects the first as primary route and allocates the slots, the schedule will include ($N_{0,0} \rightarrow N_{1,0}$) and ($N_{1,0} \rightarrow N_{1,1}$) at time slots, say, t and $t+1$, respectively. For this communication to be successful, $V_{1,0}$ at slot t and $H_{1,1}$ at slot $t+1$ should be both clear. Otherwise, the transmission will fail. Integrating an alternative path needs additional slots in the original WirelessHART.

We can modify the operation of senders and receivers as follows: The sender, $N_{0,0}$, senses a channel status for $V_{1,0}$ at slot t. If it is clear, it sends to $N_{1,0}$ according to the original schedule. Otherwise, instead of discarding the transmission, it sends to $N_{0,1}$ after switching channels, this time without CCA as another probing may violate time slot boundary ($Split$ operation). Hence, this step may lead to a useless transmission on an unclear channel. After all, in slot t, $N_{0,1}$ (the receiver on the alternative path) as well as $N_{1,0}$ (the receiver on the primary path) must listen to the channel simultaneously, and this is the overhead cost for enhanced reliability. At slot $t+1$, either $N_{1,0}$ or $N_{0,1}$ can possibly send a packet to $N_{1,1}$. $N_{1,1}$ first tries to receive via $N_{1,0}$, as it is the primary sender. If the packet arrives, it receives as scheduled. Otherwise, namely, the message doesn't arrive until the first $TsRxOffset$, the receiver immediately switches to $H_{1,1}$ and receives the packet ($Merge$ operation). For this to work, $N_{0,1}$ in the secondary route must send after a small delay as large as one channel switch time. In this way, the path is split and merged over each rectangle. This operation is summarized in Figure 4.



(a) Split operation



(b) merger operation

Figure 4. The split-merge operation

This procedure can significantly improve the transmission success ratio just with a small overhead of additional listening, wasteful communication, and channel probing. It is possible as CCA and the channel switch takes just tens of bit time, much smaller than the whole slot time. In particular, in the traffic control system, neither power nor bandwidth is a restriction, while we can safely disregard a node failure as almost every traffic light works very reliably in our everyday life, namely, the traffic light does hardly fail[24]. In addition, channel probing is a key component and it is true that a channel experiences multiple transmission rates. However, for simplicity, if the rate is higher than the given bound, the channel is considered to be clear. Moreover, it must be mentioned that false alarms may result in performance degradation, but we assume that there is no false alarm, as the correctness of channel probing is not our concern.

3.3 Slot schedule

Figure 5 shows the slot allocation table for the 3 × 3 grid shown in Figure 3(a). In a control loop, $N_{0,0}$, the controller, sends control values to all nodes. The order of appearance in the $Dest$ column follows the hop count from $N_{0,0}$, so $N_{0,1}$ comes first and $N_{2,2}$ last. To send a message to $N_{1,0}$ or $N_{0,1}$, just one slot is needed, so slot 1 and 2 are allocated, respectively. For the transmission to $N_{2,0}$, two slots are needed, as it is 2 hops away from the controller, so slot 3 via link $V_{1,0}$ and slot 4 via $V_{2,0}$ are assigned sequentially. In slot 5, $N_{0,0}$ sends a message either to $N_{1,0}$ or $N_{0,1}$ according to the CCA result of $V_{1,0}$. In slot 6, $N_{1,1}$ receives either from $N_{1,0}$ or $N_{1,1}$. So, the rows for slot 5 and slot 6 have two entries, respectively. Slot 5 corresponds to a split operation and slot 6 to a merge operation. The interval between these two operations must be as short as possible to remove the lengthy hopping sequence.

The message for $N_{2,1}$ is assigned to slots from 9 to 11. In slot 9, $N_{0,0}$ sends to $N_{1,0}$ according to the basic WirelessHART operation. Then, from $N_{1,0}$ to $N_{2,1}$, the split-merge operation is applied as specified in slot 10 and 11. It is also possible to apply the split-merge operation in the transmission from $N_{0,0}$ to $N_{1,1}$. We put this operation at the rectangular loop closer to the destination, as the early discard may have more chance for a recovery action. However, when the average error rate of all links is the same, the placement has no effect on the overall delivery ratio. Slots from 15 to 18 include the allocation for the transmission to $N_{2,2}$. Just for the primary schedule, the path is allocated $N_{0,0} \rightarrow N_{1,0} \rightarrow N_{1,1} \rightarrow N_{2,1} \rightarrow N_{2,2}$. Here, we can apply twice the split-merge procedure, as the path has two rectangles. After all, the schedule for the 3 × 3 grid includes 10 additional listening operations, so the secondary hopping sequence has 10 entries[16]. In the meanwhile, when the destination is $N_{1,0}$, $N_{2,0}$, $N_{0,1}$, or $N_{0,2}$, we cannot expect any improvement, as they have no alternative path whose length is equal to the primary one. In case the destination is $N_{2,2}$, improvement will be doubled, as we can place two split-merge operations, namely, one from $N_{0,0}$ to $N_{1,1}$, and the other from $N_{1,1}$ to $N_{2,2}$.

Slot	From	To	Dest	Link
1	$N_{0,0}$	$N_{1,0}$	$N_{1,0}$	$V_{1,0}$
2	$N_{0,0}$	$N_{0,1}$	$N_{0,1}$	$H_{0,1}$
3	$N_{0,0}$	$N_{1,0}$	$N_{2,0}$	$V_{1,0}$
4	$N_{1,0}$	$N_{2,0}$	$N_{2,0}$	$V_{2,0}$
5−1	$N_{0,0}$	$N_{1,0}$	$N_{1,1}$	$V_{1,0}$
5−2	$N_{0,0}$	$N_{0,1}$	$N_{1,1}$	$H_{0,1}$
6−1	$N_{1,0}$	$N_{1,1}$	$N_{1,1}$	$H_{1,1}$
6−2	$N_{0,1}$	$N_{1,1}$	$N_{1,1}$	$V_{1,1}$
7	$N_{0,0}$	$N_{0,1}$	$N_{0,2}$	$H_{0,1}$
8	$N_{0,1}$	$N_{0,2}$	$N_{0,2}$	$H_{0,2}$
9	$N_{0,0}$	$N_{1,0}$	$N_{2,1}$	$V_{1,0}$
10−1	$N_{1,0}$	$N_{2,0}$	$N_{2,1}$	$V_{2,0}$
10−2	$N_{1,0}$	$N_{1,1}$	$N_{2,1}$	$H_{1,1}$
11−1	$N_{2,0}$	$N_{2,1}$	$N_{2,1}$	$H_{2,1}$
11−2	$N_{1,1}$	$N_{2,1}$	$N_{2,1}$	$V_{2,1}$
12	$N_{0,0}$	$N_{0,1}$	$N_{1,2}$	$H_{0,1}$
13−1	$N_{0,1}$	$N_{1,1}$	$N_{1,2}$	$V_{1,1}$
13−2	$N_{0,1}$	$N_{0,2}$	$N_{1,2}$	$H_{0,2}$
14−1	$N_{1,1}$	$N_{1,2}$	$N_{1,2}$	$H_{1,2}$
14−2	$N_{0,2}$	$N_{1,2}$	$N_{1,2}$	$V_{1,2}$
15−1	$N_{0,0}$	$N_{1,0}$	$N_{2,2}$	$V_{1,0}$
15−2	$N_{0,0}$	$N_{0,1}$	$N_{2,2}$	$H_{0,1}$
16−1	$N_{1,0}$	$N_{1,1}$	$N_{2,2}$	$H_{1,1}$
16−2	$N_{0,1}$	$N_{1,1}$	$N_{2,2}$	$V_{1,1}$
17−1	$N_{1,1}$	$N_{2,1}$	$N_{2,2}$	$V_{2,1}$
17−2	$N_{1,1}$	$N_{2,1}$	$N_{2,2}$	$H_{1,2}$
18−1	$N_{2,1}$	$N_{2,2}$	$N_{2,2}$	$H_{2,2}$
18−2	$N_{1,2}$	$N_{2,2}$	$N_{2,2}$	$V_{2,2}$

Figure 5. Slot schedule for the 3 × 3 grid

4 Routing scheme

As mentioned before, the communication scheduler decides the channel schedule. In the control loop scenario, traffic goes from and to $N_{0,0}$. Namely, each node sends and receives a message to and from the coordinator node once in the period specified by the system requirement. Even if it is desirable to take the route which has the minimum number of hops to the destination, another detour can have advantage in terms of delivery ratio and transmission delay. A path is to be updated by the scheduler periodically according to the change in the error characteristics of each link. Each link has its own error characteristics due to different power level, obstacle distribution, and so on, while the change of link error characteristics can be estimated in many different ways[7]. We assume that the probing result is always correct, as the correctness of channel estimation is not our concern.

To begin with, we denote the error rate of link L by $E(L)$. Intuitively, L may be either a horizontal or vertical link. For example, $E(V_{2,1})$ denotes the error rate of the link from $N_{2,1}$ to $N_{1,1}$ and it has the same meaning with $E(N_{2,1} \rightarrow N_{1,1})$. However, L can be extended to represent a virtual link which consists of a single split-merge pair. That is, even though there is no direct link between the two nodes in a diagonal line, the two-hop link between them can be considered to be a single link, taking into account the split-merge operation. The split-merge operation works for the downlink from $N_{i,j}$ to $N_{i+1,j+1}$. For a downlink path, we have two path, namely, $N_{i,j} \rightarrow N_{i,j+1} \rightarrow N_{i+1,j+1}$ and $N_{i,j} \rightarrow N_{i+1,j} \rightarrow N_{i+1,j+1}$, each of which corresponds to ($H_{i,j+1}, V_{i+1,j+1}$) and ($V_{i+1,j}, H_{i+1,j+1}$), respectively.

The first step is to decide which one is the primary path when every link has different error rates. The path having the lower error rate will be the primary path. If we let the error rate of the first one be $F_1(N_{i,j}, N_{i+1,j+1})$ and that of the second one $F_2(N_{i,j}, N_{i+1,j+1})$, they can be calculated as follows:

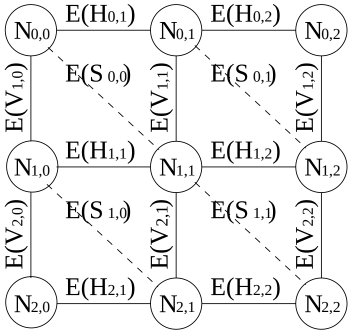
$$\begin{aligned} F_1(N_{i,j}, N_{i+1,j+1}) = \\ &(1 - E(H_{i,j+1}))(1 - E(V_{i+1,j+1})) + \\ &E(H_{i,j+1})(1 - E(V_{i+1,j})(1 - E(H_{i+1,j+1})) \end{aligned}$$

$$\begin{aligned} F_2(N_{i,j}, N_{i+1,j+1}) = \\ &(1 - E(V_{i+1,j}))(1 - E(H_{i+1,j+1})) + \\ &E(V_{i+1,j})(1 - E(H_{i,j+1})(1 - E(V_{i+1,j+1})) \end{aligned}$$

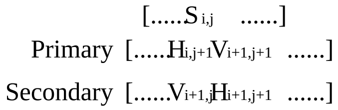
Recall that the sender of a slot decides which route to take according to the CCA result of the first link on the primary route, and the above relations are straightforward. If F_1 is greater than F_2, $(H_{i,j+1}, V_{i+1,j+1})$ is set to the primary path. Otherwise, $(V_{i+1,j}, H_{i+1,j+1})$ will be the primary path. The smaller of the two, namely, F_1 or F_2, will be the expected error probability of a virtual link from $N_{i,j}$ to $N_{i+1,j+1}$. Based on this estimation, we can build a communication graph including the error rate as the link cost as shown in Figure 6 (a). Here, the error rate can be replaced by the success probability by subtracting the former from 1.0. Now, Dijkstra's (one-to-many) shortest path algorithm can find the best route having the lowest error rate for both graphs, after substituting the product of probabilities for the sum of link costs in each node expansion. In each node scan, the node having the highest success ratio will be selected. Finally, the instance of virtual link, say $S_{i,j}$, is to be expanded by the split-merge schedule as shown in Figure 6 (b) to finalize a complete slot assignment. In this example, $S_{i,j}$ is replaced by $(H_{i,j+1}, V_{i+1,j+1})$ on the primary schedule and $(V_{i+1,j}, H_{i+1,j+1})$ for the secondary schedule, provided that $F_1(N_{i,j}, N_{i+1,j+1})$ is smaller than $F_2(N_{i,j}, N_{i+1,j+1})$. Additionally, uplink case is also shown in Figure 6 (c).

5 Performance measurement

This section measures and assesses the performance of the proposed process control scheme via simulation using SMPL which provides a lot of functions and libraries for the discrete event scheduler, easily combined with the commonly used compilers such as *Visual C++* and *gcc*[17]. Two cases are considered in the measurement. For the first case, it is assumed that every node is set to have the same average error rate, focusing on the reliability enhancement achieved by the proposed scheme. In the second case, we remove this restriction and each link has a different error rate, to assess the efficiency of our routing scheme. We considered a 4 × 4 grid and it needs 48 slots for each control loop. Nodes and channels are numbered by the same way as in Figure 3. According to the proposed schedule, 28 out of 48 slots participate in the split-merge operation. The first experiment measures the ratio of successful transmissions according to the slot error rate for all end-to-end transmissions generated in this system during the simulation time. The slot error rate depends on the data size and the channel error rate distribution. Here, we employ Guilbert-Elliot error model, which is quite simple, but can easily set the average error rate we want for the experiment.



(a) cost graph



(b) downlink substitution

Figure 6. Cost graph and substitutions

As shown in Figure 7, the performance gap between the grid and proposed schedules gets significantly larger along with the increase of slot error rate, reaching 7.9 % when the slot error rate is 0.21. Here, the grid schedule takes a route without split-merge operations and it is generated also by means of the Dijkstra's shortest path algorithm, not adding any virtual link. As the error rate is the same for all links, the minimum hop route corresponds to the grid schedule. The probability of successful transmissions drops exponentially at each hop. So, the node many hops away from the controller has significantly low success ratio. On the contrary, the split-merge procedure can reduce the delivery failure almost by half for each 2-hop path. As this term is propagated by multiplication, the performance gap gets magnified on the node far away from the controller. Actually, the larger the grid, the more slots get involved in the split-merge operation and the average hop length between the controller and other nodes also increases. So, we can expect more improvement in the wireless communication for larger grids.

Figure 8 plots success ratios for the two schemes accord-

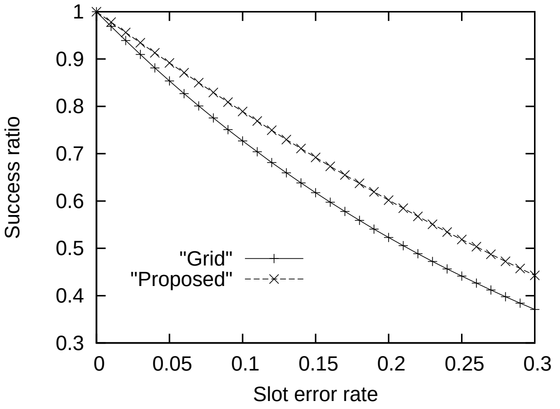


Figure 7. Effect of slot error rate

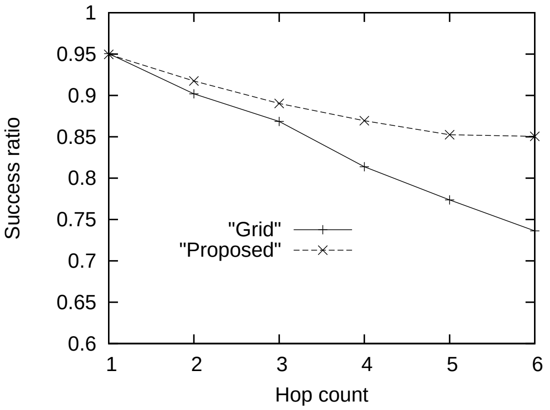


Figure 8. Success ratio vs. hop count

ing to the hop count, with the slot error rate fixed to 0.05. There is just one node which has hop count of 6, namely, $N_{3,3}$. The transmission from the controller to this node has 4 split-merge operations in $N_{0,0} \rightarrow N_{1,1}$, $N_{1,1} \rightarrow N_{2,2}$, $N_{1,1} \rightarrow N_{2,2}$, and $N_{2,2} \rightarrow N_{3,3}$, so many channel errors can be recovered. No 1-hop node can benefit from our scheme, so two curves meet 0.95. The majority of 2-hop and 3-hops nodes have just the straight path, namely, they are on the same row or column with the controller, so their success ratios are not so high. On the contrary, 4-hop and 5-hop nodes have at least one rectangular path in the 4 × 4 grid, so we can see the significant reliability enhancement. After all, the success ratios for the 4, 5, and 6-hop nodes reach 86.9 %, 85.2 %, and 85.0 %, obtaining 5.5 %, 7.8 %, and 11.4 % enhancement, respectively.

In the second set of experiments, we let each link have different error rates. We add virtual links after estimating their error rates. Again, only the downlink graph was considered, so the controller transmits messages to each node one by one according to the slot assignment as described in the previous section. The experiment measures the delivery ratio of end-to-end messages for the proposed routing scheme and compares with the case that decides a route and allocates the slot based on the general shortest path algorithm on the grid topology. Figure 9 exhibits the performance improvement according to the average slot error rate. This experiment makes each link have a different average error rate according to the exponential distribution. Namely, the experiment generates 500 sets for each average slot error rate ranging from 0.0 to 0.3, and measures the message delivery ratio from the controller to all other nodes. As shown in this figure, the proposed routing scheme can enhance the delivery ratio by up to 6.69 % when the slot error rate is 0.2. The performance gap tends to increase according to the increase of slot error rate.

Figure 10 plots the success ratio according to the dimension of square grids. Here, n in the x-axis means the $n \times n$ grid, and the slot error rate is set to 0.1. A larger grid makes the average hop count increase, so the success ratio drops according to the increase of the grid dimension. However, the proposed scheme always shows the better success ratio and is less affected by the grid dimension, as the slit-merge operation is also more likely to be applied. For the 10 × 10 grid, the performance gap reaches 9.0 %.

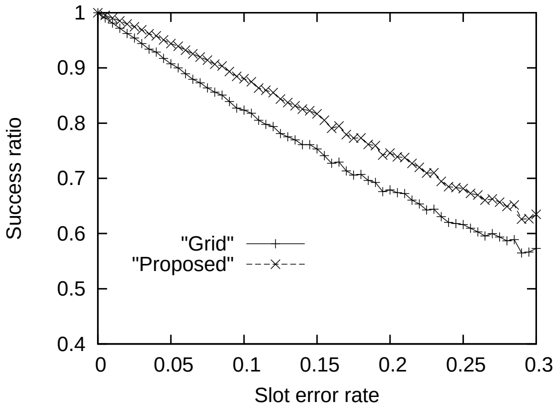


Figure 9. Effect of the routing scheme

In the control message exchange, some nodes must wake up and listen to the channel on the secondary path to receive the possible transmission. This additional receive operation is the overhead brought by the split-merge operation and leads to power consumption. The receive operation consumes much less power than the transmit operation, and as mentioned previously, the power consumption has little effect on the traffic light network. However, in the general wireless mesh network, power management may matter. For the $n \times n$ grid, the number of total transmissions for a control round is calculated as in (1).

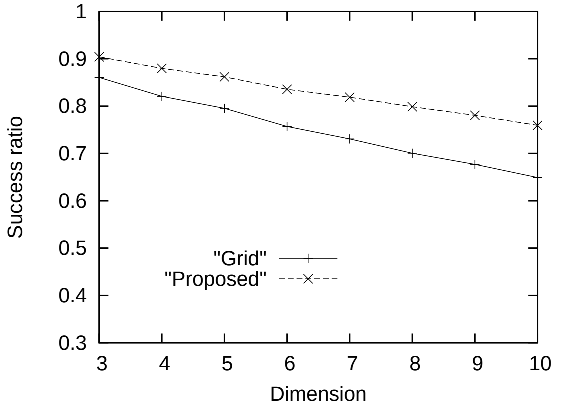


Figure 10. Effect of the grid dimension

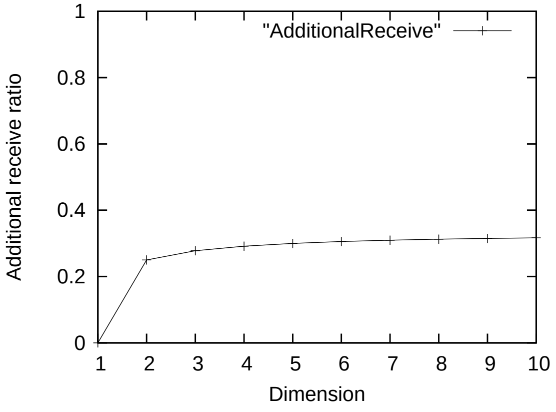


Figure 11. Additional receive ratio

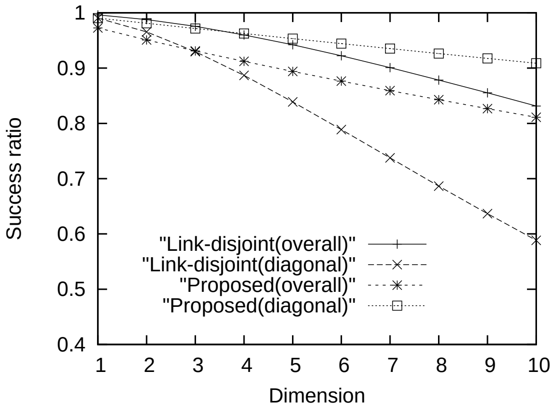


Figure 12. Comparison with multipath routing

$$\sum_{i=1}^{2(n-1)} (n - |n - i - 1|) \cdot i \tag{1}$$

Each iteration indexed by i corresponds to the hop count from the controller. Nodes having the same hop counts need the same number of slots. The number of nodes having i hops increases by one until i reaches (n-1), and then decreases by one until i reaches 2(n-1). In addition, the number of additional receive operations for the $n \times n$ grid is calculated as in (2).

$$\sum_{i=1}^{n-1} \{2(n - i) - 1\} \tag{2}$$

For node $N_{i,i}$, on the diagonal, there can be split-merge operations i times, so it needs i additional receive operations. The number of applicable split-merge operations is equal to that of $N_{i,i}$ for all nodes $N_{i,j}$, where $j \geq i$ and $N_{j,i}$, where $j \geq i$. After all, the additional receive operation ratio can be calculated by diving (1) by (2) and Figure 11 plots the ratio according to the grid dimension. Here, we can see that the additional receive operation remains below 33 % and does not increase uncontrollably even for larger n.

It is difficult to directly compare the proposed scheme with multipath routing, which tries to find a node-disjoint or link-disjoint path to enhance the transmission reliability by reserving an alternative path. Link-disjoint paths share no common link, and multipath transmissions via those two links need two explicit transmit and receive operations, respectively[4]. In link-disjoint path routing, if every transmission and reception is duplicated, the additional receive ratio is 100 %. In grid topology, a multipath routing scheme finds an alternative path very easily from the controller to all nodes. As contrast, our scheme takes an alternative path when necessary, and this action takes place in a single time slot. However, every node cannot benefit from the split-merge operation. No enhancement can be expected for the nodes on the same row or column of the controller, while those on the diagonal can fully take advantage of redundant paths.

Figure 12 plots the success ratio for diagonal nodes and for overall grid nodes to compare with the link-disjoint path, assuming ideally that there are always two link-disjoint paths having the equal length for all nodes. The error ratio is set to 0.05. For two curves marked as *diagonal*, the node number i means $N_{i,i}$. The success ratio can be improved by up to 32 % for $N_{10,10}$. In addition, for two curves marked as *overall*, the success ratio means the average success ratio of every node in the $n \times n$ grid. The link-disjoint case shows just a little bit better performance, but it needs doubled network bandwidth and extended control loop. Our scheme requires much less transmissions and works with a shorter control loop, but performance comparable to the multipath routing scheme.

6 Conclusion

This paper designs and analyzes the performance of a reliable communication scheme for the traffic control system based on WirelessHART, a promising protocol for wireless process control. Based on the grid topology of this network and the CCA technique that senses before transmission, the proposed scheme can generate a robust slot assignment schedule that can efficiently cope with the wireless channel error. When the sensing result is not good, a sender takes an alternative route having the same length with the primary route in a single slot. Hence, the path is split and merged at each branch point of the grid. In addition, the two joint links can be considered to be a single virtual link which has a much lower error rate. This paper also deals with a routing scheme that finds the route that has the lowest error rate considering such a virtual link.

The performance measurement result obtained by simulation using a discrete event tracer shows that the proposed scheme can improve the transmission success ratio for the 4 $\times$ 4 grid by up to 7.9 %, compared with the general allocation method based on the shortest path or the minimum error probability. In addition, our routing scheme can enhance the delivery ratio by 6.69 % when each link has its own slot error rate and the rate distributes exponentially. The proposed scheme can keep the additional receive ratio less than 33 % up to the dimension of 10. Particularly, the split-merge operation makes it possible for the message transmission carried out with halved network bandwidth and a shorter control loop, but achieving performance comparable to the multipath routing scheme.

As future work, we are first planning to improve this scheme to efficiently cope with node failure to apply to the general wireless mesh networks. In addition, a fault-tolerant flooding scheme is expected to be very useful in wireless process control[22]. So, we will design a slot assignment scheme combining the split-merge operation to overcome node or link failure for the control message broadcast.

References

[1] *Technical Report HS 810 591:Vehicle safety communication project-final report*. US Depart of Transportation, 2006.

[2] *IEC/PAS 62591: Industrial communication networks - Fieldbus specifications - WirelessHART communication network and communication profile*. 2008.

[3] *Why WirelessHART^{TM}? The Right Standard at the Right Time*. Hart Communication Foundation, 2008.

[4] S. Adibi and S.Erfanu. A multipath routing survey for mobile ad-hoc networks. *IEEE Consumer Communications and Networking Conference*, pages 984–988, 2006.

[5] R. Alur, A. D'Innocenzo, G. Pappas, and G. Weiss. Modeling and analysis of multi-hop control networks. *The 15th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 223–232, 2009.

[6] J. Cain, T. Billhartz, L. Foore, E. Althouse, and J. Schlorff. A link scheduling and ad hoc networking approach using directional antennas. military communications conference. *Military Communications Conference*, pages 643–648, 2003.

[7] N. Chang and M. Liu. Optimal channel probing and transmission scheduling for opportunistic spectrum access. *ACM international conference on Mobile computing and networking*, pages 27–38, 2007.

[8] D. Chen, M. Nixon, T. Aneweer, R. Shepard, and A. Mok. Middleware for wireless process control systems. *WACERT*, 2004.

[9] J. Chen, C. Chen, R. Jan, and R. Li. Expected link life time analysis in manet under manhattan grid mobility model. *ACM MSWIM*, pages 162–168, 2008.

[10] K. Chowdhury, N. Nandiraju, P. Chanda, P. Agrawal, and Q. Zeng. Channel allocation and medium access control for wireless sensor networks. *Ad Hoc Networks*, 7:307–321, 2009.

[11] D. Gislason. *ZIGBEE Wireless Networking*. Newnes, 2008.

[12] S. Han, J. Song, X. Zhu, A. Mok, D. Chen, M. Nixon, W. Pratt, and V. Gondhalekar. Wi-htest: Compliance test suite for diagnosing devices in real-time wirelesshart network. *The 15th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 327–336, 2009.

[13] S. Jaap, M. Bechler, and L. Wolf. Evaluation of routing protocols for vehicular ad hoc networks in city traffic scenarios. *The 5th International Conference on Intelligent Transportation Systems Telecommunications*, 2005.

[14] A. Khattab and K. Elsayed. Channel-quality dependent earliest deadline due fair scheduling schemes for wireless multimedia networks. *ACM MSWIM*, pages 31–38, 2004.

[15] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. *ACM MobiCom*, pages 73–87, 2005.

[16] J. Lee, I. Shin, and C. Kim. Design of a reliable traffic control system on city area based on a wireless network. *Lecture Note on Computer Science*, 5592:831–840, June 2009.

[17] M. MacDougall. *Simulating Computer Systems: Techniques and Tools*. MIT Press, 1987.

[18] N. Mullter. *Bluetooth Demystified*. McGraw-Hill TELECOM, 2000.

[19] I. Ramchandran and S. Roy. Clear channel assessment in energy-constrained wideband wireless networks. *IEEE Wireless Magazine*, pages 70–78, 2007.

[20] A. Raniwala, K. Gopalan, and T. Chieuh. Centralized algorithms for multi-channel wireless mesh networks. *ACM Mobiel Computing and Communication Review*, 2004.

[21] J. Song, S. Han, A. Mok, D. Chen, M. Nixon, M. Lucas, and W. Pratt. Wirelesshart: Applying wireless technology in real-time industrial process control. *The 14th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 377–386, 2008.

[22] P. Wan, S. Huang, L. Wang, Z. Wan, and X. Jia. Minimum latency aggregation scheduling in multihop wireless networks. *ACM MobiHoc*, pages 185–193, 2009.

[23] H. Woo and A. Mok. Real-time monitoring of uncertain data streams using probabilistic similarity. *Proc. of IEEE Real-Time Systems Symposium*, pages 288–300, 2007.

[24] W. Xiuchao and A. Ananda. Link characteristics estimation for ieee 802.11 dcf based wlan. *IEEE Int'l Conference in Local Computer Networks*, 2004.