Chapter 5
Mixed-Criticality Scheduling with Multiple Radio Interfaces

Abstract In this chapter, we introduce the nodes with multiple radio interfaces (MRI) into mixed-criticality industrial wireless networks. When an error occurs or transmission demand changes, the MRI nodes can switch their transmission mode, changing to a high-criticality configuration to meet the system's new demand. Hence, we first propose a heterogeneous MRI system model. Based on this model, we propose a Slot Analyzing Algorithm (SAA) that guarantees system schedulability by reallocating slots for each node after replacing conflict nodes with MRI nodes. By considering both system schedulability and cost, SAA also reduces the number of MRI nodes. Then, we propose a Priority Inversion Algorithm (PIA) that improves the schedulability by adjusting slot allocations before replacing conflict nodes with MRI nodes. By reducing the use of MRI nodes, PIA achieves better performance than SAA when the system is in the high-criticality mode.

5.1 Background

Due to the high real-time and reliability requirements of industrial systems, traditional cyber-physical systems cannot be applied directly [1–4]. To guarantee the requirements of ICPSs, we introduce multiple radio interface (MRI) nodes into ICPSs. A traditional network node is usually equipped with only one antenna, while an MRI node with two antennas, can both receive and send packets simultaneously [5–7]. By replacing conflict nodes with MRI nodes, multiple flows can transmit without delays. If we were to replace all the conflict nodes with MRI nodes, that would eliminate transmission conflicts in the system; consequently, when the number of MRI nodes is sufficient, we can guarantee system schedulability. However, considering the energy consumption and cost of MRI nodes, it is advantageous to be able to guarantee the network schedulability using fewer MRI nodes [8–12]. Hence, there are two main problems in mixed-criticality industrial wireless networks, (1) reducing the number of MRI nodes required to guarantee the network schedulability and (2) analyzing the network schedulability under different criticality modes. To address these problems, we propose two algorithms, a Slot Analyzing Algorithm (SAA) and a Priority Inversion Algorithm (PIA), to

X. Jin et al., *Mixed-criticality Industrial Wireless Networks*, Wireless Networks,
https://doi.org/10.1007/978-981-19-8922-3_5

improve the network schedulability. SAA first allocates slots without considering transmission conflicts and replaces conflict nodes with MRI nodes to guarantee future schedulability in low-criticality mode. In contrast, PIA first optimizes slot allocation to reduce the number of transmission conflicts; then, we replace conflict nodes with MRI nodes only when the system cannot be scheduled. The contents of this chapter are as follows:

1. We first propose an algorithm to obtain the candidate node set (CNSA), which indicates the key nodes that affect system performance, and then propose SAA to improve system schedulability. SAA can guarantee that a system can be scheduled when it satisfies Theorem 5.1. Furthermore, SAA also reduces the number of MRI nodes through slot analysis.
2. We propose PIA to reduce the number of transmission conflicts before replacing the overlap nodes with MRI nodes. When a flow cannot be scheduled, priority inversion occurs when it satisfies Theorem 5.3.
3. We analyze system schedulability in high-criticality mode and prove that PIA achieves better schedulability than SAA in Theorem 5.4.
4. Simulation results show that our algorithms can improve the schedulability of industrial wireless networks with MRI nodes and that PIA achieves a better performance than SAA in the high-criticality mode.

5.2 System Model

We consider a heterogeneous network consisting of field devices (both MRI nodes and traditional sensor nodes), one centralized controller and one gateway. In this section, firstly, we propose a network model, and then we introduce MRI nodes into mixed-criticality networks. Finally, we apply a fixed priority (FP) scheduling scheme in industrial wireless networks.

5.2.1 Network Model

The system model is based on WIA-PA [13]. There are several features as follows: (1) Time Division Multiple Access (TDMA); (2) Route and Spectrum Diversity and (3) Handling Internal Interference. The number of channels is denoted as m, and there are N fixed nodes in our system.

Industrial systems have higher requirements of real-time performance and reliability. Here, we introduce MRI nodes into industrial systems. MRI nodes are promising for wireless transmissions because they can improve the average user spectral efficiency [14]. By mounting multiple antennas on a single sensor node, the node can receive and transmit simultaneously (or improve the packet acceptance ratio when all the antennas work on the same channel).

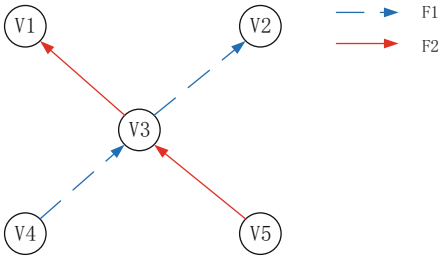

Fig. 5.1 An example

Hence, our model includes two types of sensor nodes: traditional nodes and MRI nodes. Each traditional node in our system is equipped with a half-duplex omni-directional radio transceiver whose status can alternate between transmitting and receiving. In contrast, an MRI node has two working modes: (1) when the antennas work on different channels, the node can both receive and transmit packets from several paths (the transmission capacity of an MRI node depends on the number of antennas it has); and (2) when the antennas work on the same channel, the node can reduce signal interference and improve the acceptance ratio because several antennas receive a packet from the same flow. Each packet in our system is transmitted through the network under source routing.

Figure 5.1 shows an example of an MRI node in the first mode (where antennas work on different channels) to address a transmission conflict in which F_1 and F_2 send packets to the same destination simultaneously. We introduce MRI to solve this issue. The flows can be scheduled by replacing $V3$ with an MRI node. MRI nodes can also be used as a method to increase the packet acceptance ratio when two or several antennas receive packets on the same channel. For the second mode of an MRI node (where the antennas work on the same channel), when the links around $V3$ have poor signal quality (such as from co-channel interference, intermodulation interference, spurious emissions or adjacent-channel interference), node $V3$ can switch its antenna receiving frequencies to the same channel to improve the acceptance ratio. However, considering that the power consumption and cost of MRI nodes is considerably higher than the cost of a normal node, we cannot deploy MRI nodes throughout the entire system.

5.2.2 *Mixed-Criticality System*

Due to the introduction of MRI nodes, our mixed-criticality system model differs from other models [15–19]. The flow is a periodic end-to-end communication between a source and its destination. There are n flows in our system, denoted by $F = \{F_1, F_2, \ldots, F_n\}$. F_i is characterized by $< t_i, d_i, \xi, c_i, p_i >$, a number between $1 \leq i \leq n$, where the period is t_i, the deadline is d_i, the criticality level is ξ (we focus on a dual-criticality system $\{LO, HI\}$) in which $\xi = 2$, meaning the system has two critical levels. The superframe as the lowest common multiple of

the flow periods is denoted as T. Initially, the system works at a low critical level, and the MRI nodes receive packets on different channels. The system switches to the high critical level to enhance system reliability, and the MRI nodes reconfigure their antennas to receive high critical flow packets on the same channels when the accident occurs. Here, c_i is the number of hops from a source to a destination, and the routing path is p_i. When the system critical level switches from low to high, the MRI node's work pattern changes from receiving on different channels to receiving on the same channels to improve the schedulability of high critical flows. In addition, the characteristics of high-criticality flows switch to high-criticality mode. In this chapter, the system reduces the sampling period of a high-criticality flow to $\aleph_i t_i$ to improve the sampling rate of high critical flow, where $\aleph_i$ is a value that satisfies $\frac{c_i}{t_i} \leq \aleph_i < 1$. To simplify the calculation we assume that $d_i = t_i$. Hence, $\aleph_i$ satisfies $\frac{c_i}{d_i} \leq \aleph_i < 1$. We model the duration of the mode switch as γ, which is used to calculate the delay of packets delivered during the mode change.

In the beginning, packets are transmitted in low critical mode. To provide more resources, each MRI node's antennas work on different channels. The system switches when the accident occurs. To enhance system reliability in high-criticality mode, each MRI node's antennas are reconfigured to work on the same channel.

5.2.3 Fixed Priority Scheduling

We provide a summary of fixed priority scheduling to analyze the schedulability of systems in this subsection. FP scheduling is a commonly adopted scheme in practice for cyber-physical systems and real-time CPU scheduling [20]. Each job priority is pre-allocated by the network controller, and transmissions are scheduled based on this priority.

We assume that the priority of each flow is the same as its number. That is, F_1 has the highest priority, and F_n has the lowest priority in the system. There are two types of delays in industrial wireless sensor systems: (1) Channel contention and (2) transmission conflicts, the definitions can be obtained in [21].

We define a system as schedulable when all the flows in a system can be scheduled (reach the destination before its deadline). Then the definition of network schedulability is whether or not all flows in a network are schedulable. It is worth noting that when the system is in high-criticality mode, we no longer focus on the schedulability of low-criticality flows. When we repeat Z experiments, and only z experiments the emergency flow can be scheduled, then, the schedulability ratio is $\frac{z}{Z}$.

5.3 Problem Statement

Given the flow set F, a wireless network, and the FP scheduling policy, our objective is to use MRI nodes to improve the schedulability of mixed-criticality industrial networks. We first analyze the schedulability of mixed-criticality industrial wireless networks. Initially, the system is working in low-criticality mode. To improve system schedulability, we introduce MRI technology to provide more resources (the resources such as slots and channels are increased since MRI nodes can receive and send packets simultaneously). Hence, we replace nodes at several intersections with MRI nodes and propose a Priority Inversion Algorithm to guarantee that the system can be scheduled. However, when the accident occurs, we must guarantee high-criticality flows schedulability; consequently, the system switches to high-criticality mode, and the transmission mode of MRI nodes changes. We then analyze the schedulability of our method to evaluate the quality of PIA. The challenges in this situation are listed below.

1. When the system is deployed, we can easily determine flows that miss their deadlines. However, each flow's schedulability is interrelated with others. Thus, how should we decide which nodes should be replaced with MRI nodes?
2. As described in the previous section, the power consumption and cost of an MRI node are much higher than those of a normal node. Therefore, it is unreasonable and not cost-effective to deploy many MRI nodes when the system can be scheduled. Hence, the problem of determining the smallest number of MRI nodes that can meet the system requirements in low-criticality mode is another challenge.
3. The MRI nodes' transmission mode changes when the system switches to high-criticality mode. Therefore, the scheduling algorithm also needs to consider schedulability in high-criticality mode.
4. We need to analyze the system's schedulability under our proposed method.

5.4 Scheduling Algorithms

In this section, we first study the issue of how to improve system schedulability with a small number of MRI nodes. First, we identify the nodes that may be replaced with MRI nodes to define candidate nodes as follows:

Definition 5.1 (Candidate Node) We define a candidate node as a node at which transmission conflicts (intersection or overlap nodes) can occur. As Fig. 5.1 shows, the paths of F_1 and F_2 intersect at $V3$; consequently, a transmission conflict may occur at this node. Thus, $V3$ is a candidate node.

Flows may conflict when they have path overlaps. To facilitate candidate node identification, we assume that, at most, one part of a path overlaps between two flows. As Fig. 5.2 shows, two periodic flows transmit in a network that conflict at

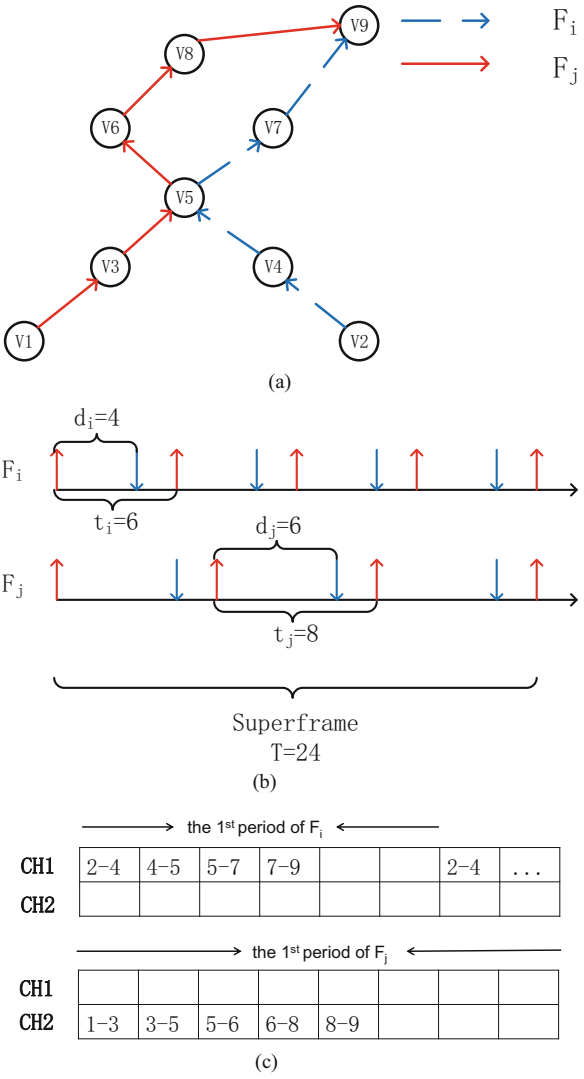

Fig. 5.2 The analysis of conflicts. (**a**) Routing. (**b**) Superframe. (**c**) Scheduling

the second and third slots in the first period. There are two methods for addressing transmission conflicts. The first method is to reallocate the slots for each node after replacing the conflict nodes with MRI nodes (as Fig. 5.2 show, the system can be scheduled when we replace node $V5$ with an MRI node, where CH is the abbreviation of Channel); the other method is to adjust slot allocation, as shown in Fig. 5.3. By rationally allocating each flow's transmission slots, we can reduce the number of MRI nodes and the signal interference caused by the coexistence of multiple channels, which can also improve the schedulability when the wireless network is in high-criticality mode. However, this is unsuitable for a system that

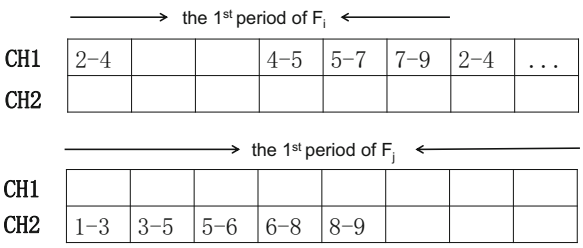

Fig. 5.3 An example for un-schedulable conflict

cannot be scheduled (in this example, the system cannot be scheduled if the deadline of F_i is 4).

Therefore, for a system in which each link's transmission slot has been allocated, we propose the slot analyzing algorithm to improve system schedulability under a traditional FP policy. When the system cannot be scheduled by MRI nodes, SAA re-allocates system resources to guarantee system schedulability (increase the transmission speed of unscheduled data flows). By considering the characteristics of mixed-criticality systems and MRI nodes, we then propose the PIA algorithm, which adjusts slot allocations before replacing intersection nodes with MRI nodes (the second method to address transmission conflict). By optimizing slot allocation, PIA can guarantee system schedulability with fewer MRI nodes. To guarantee the schedulability of the network, PIA will also re-allocate slots when the system cannot be scheduled. It is important to note that initially both the SAA and PIA algorithms work in low-criticality mode. We will analyze the schedulability of these two algorithms later.

5.4.1 Finding Candidate Nodes

A network consists of numerous nodes. In this subsection, we study how to select candidate nodes to guarantee the system can be scheduled. Transmission conflicts occur at the path overlaps of flows. As Fig. 5.4 shows, there are two types of overlaps (without considering the flow's direction).

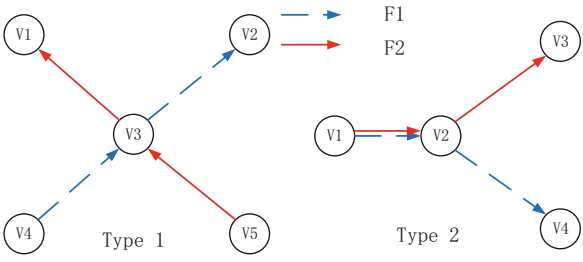

Fig. 5.4 Transmission conflict

Lemma 5.1 *When flow paths have an overlapping region, the overlap nodes are candidates for MRI nodes. We denote the candidate node set as* Λ*, and the node set on each flow's path can be denoted as* λ_i*.*

Proof Transmission conflict can obviously only occur at an overlapping region. We can solve this issue by MRI nodes. We can account for the conflicts caused by the first type as shown in Fig. 5.1. For the transmission conflict caused by the second type, as Fig. 5.4 shown, there are two nodes ($V1$ and $V2$) on both F_1 and F_2 involved in the second type of overlap. When this type of transmission conflict occurs, we can improve the schedulability by replacing both $V1$ and $V2$ with MRI nodes. Hence, when flow paths have an overlapping region, the overlap nodes are candidates for MRI nodes. □

Hence, we propose a Candidate Node Searching Algorithm (CNSA) to search for the set of candidate nodes as follows:

Algorithm 5.1 Candidate node searching algorithm

Input: F;
Output: the candidate node set $\Lambda = \{\lambda_i\}, i \in F$;
1: **for** each flow F_i **do**
2: **if** node n_k is the overlap node **then**
3: n_k join Λ;
4: **end if**
5: **end for**
6: **return** Λ;

We search the candidate nodes by traversing the flow paths of the entire system. Nodes on more than one flow path are added to the candidate node set Λ. The time complexity of CNSA is $O(F^2)$.

5.4.2 Slot Analyzing Algorithm

After obtaining the candidate node set, we reduce the number of nodes in this set to reduce the system's cost. Because not all the candidate nodes can experience transmission conflicts, we propose the slot analyzing algorithm to analyze the schedulability of the network under a traditional FP policy in one superframe (a superframe is the lowest common multiple of t_i, $i \in F$). SAA improves system schedulability by replacing some of the nodes in Λ with MRI nodes.

Since SAA is used after the CNSA, we do not consider the slot allocations for each flow; we just replace conflict nodes with MRI nodes when a transmission conflict occurs.

When the system cannot be scheduled, we need to improve the schedulability using MRI nodes. Because MRI nodes can both receive and send packets in a single time slot, we first allocate slots for each node without considering transmission

conflicts (the network controller allocates the channel for each transmission using the FP scheduling policy). Then, we replace conflict nodes with MRI nodes to guarantee system schedulability. Obviously, it is both unnecessary and not cost-effective to replace all the nodes in the candidate node set with MRI nodes. Therefore, SAA reduces the number of MRI nodes in Λ through slot analysis. The pseudo code of the SAA algorithm is as follows:

Algorithm 5.2 Slot analyzing algorithm

Input: the characters for each flow F_i; the candidate node set $\Lambda = \{\lambda_i\}$, $i \in F$;
Output: the schedulability of the network;
1: reallocate the slot for each node.
2: **for** each flow i **do**
3: **if** the flow cannot be scheduled **then**
4: find the intersection nodes and reallocate slots for F_i without considering transmission conflicts.
5: **else**
6: retain the original allocation.
7: **end if**
8: **end for**
9: **for** each node $\lambda_i \in \Lambda$ **do**
10: **if** λ_i has two or more than two transmissions in the same time slot in one superframe **then**
11: λ_i needs to be replaced;
12: **end if**
13: **end for**
14: **if** the system also cannot be scheduled **then**
15: **for** each unscheduled flow **do**
16: enhance the unscheduled flow's transmission speed;
17: **if** λ_i has two or more than two transmissions in the same time slot in one superframe **then**
18: λ_i needs to be replaced;
19: **end if**
20: **end for**
21: **end if**
22: update λ_i and $\Lambda = \{\lambda_i\}$;
23: **return** Λ;

We reallocate the transmission time slots for each node by the schedulability of each flow (lines 1–8). If the flow cannot be scheduled, we find the intersection nodes and reallocate the slots for this flow without considering transmission conflicts. Otherwise, we retain the original allocation. Then, we analyze the transmission slot for each node in Λ (lines 9–23). When a node in Λ has more than one transmission at the same time slot, that means a transmission conflict occurs at this node, and we need to replace the node with an MRI node (lines 9–13). When the system also cannot be scheduled, SAA then re-allocates slots to accelerate the transmission speed of unscheduled flows until the flows can be scheduled (lines 14–21). Finally,

SAA updates λ_i and returns Λ—the set of nodes that need to be replaced (lines 22–23). The theorem is as follows:

Theorem 5.1 *A network can be scheduled with SAA when the number of channels is no less than the number of flows ($m \geq n$).*

Proof When k flows conflict at node A, we denote the flow with the highest priority as F_1, and the flow with the lowest priority as F_k. F_1 transmits first and cannot be delayed at node A. The other flows must wait on F_1,consequently, this generates delay. When the number of channels is no less than the number of flows, no delay is caused by transmission conflicts. Hence, flow $F_i, i \in k$ will miss its deadline and cannot be scheduled when $c_i + del_i > d_i$, where del is the delay slots. Using SAA, we can eliminate the delays caused by transmission conflicts. k flows can transmit simultaneously when $m > n$ because the number of hops in each flow's transmission c is no larger than its deadline d. Thus, we can guarantee all the flows can be scheduled and, therefore, the system can be scheduled using SAA when the number of channels is no less than the number of flows. □

5.4.3 Priority Inversion Algorithm

In this subsection, we study how to reduce the number of MRI nodes by the second method, the PIA scheduling algorithm, which performs optimal allocation of resources before the system runs. Through optimal allocation, PIA can decrease the number of transmission conflicts at intersection nodes and further reduce the number of required MRI nodes. As Fig. 5.5 shows, $V5$ does not need to be replaced by an MRI node when $d_i = 6$ in Fig. 5.2a.

Initially, PIA allocates slots based on the traditional FP policy. If the system cannot be scheduled, the network controller obtains each flow's arrival time at its destination, which can be denoted as r_i. Then, we can obtain the following theorem,

Theorem 5.2 *When F_i and F_j conflict at node set $V = V_1, V_2 \ldots, V_h$ ($i < j$), we denote the length of the overlap as $Len(ij)$, and its conflict delay is $D(ij)$. The schedulability of F_i is unaffected by the priority inversion of F_i and F_j when it satisfies:(1) $r_i + D(ij) \leq d_i$;(2) V is the last part of the overlapping nodes of F_i.*

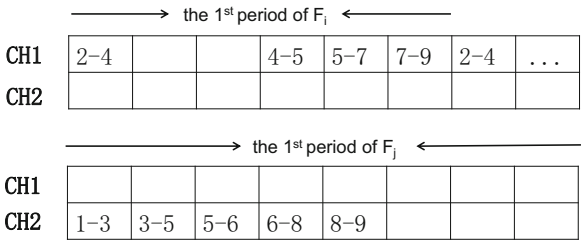

Fig. 5.5 Transmission conflict

Proof The priority of F_i is higher than that of F_j based on the assumption that $i < j$. When transmission conflict occurs at node set $V = V_1, V_2 \ldots, V_h$, F_j must be delayed by F_i. We can accelerate the transmission speed by transmitting F_j first when the schedulability of F_i does not be changed. We introduce the result proposed by Saifullah in [22], in which the upper bound $\Delta(i, j)$ of the delay that F_j can experience from an instance of F_i is

$$\Delta(ij) = \sum_{k=1}^{\delta(ij)} Len_k(ij) - \sum_{k'=1}^{\delta'(ij)} (Len_{k'}(ij) - 3), \tag{5.1}$$

where $\delta(ij)$ is the number of path overlaps, $Len_k(ij)$ is length of the kth overlap on F_i and F_j, $\delta'(ij)$ is the number of path overlaps larger than 3. Hence, the problem is transformed into one of calculating the delay F_i must bear while waiting for F_j to transmit. By simplifying Eq. (5.2), we can obtain the delay in V as follows:

$$D(ij) = \begin{cases} Len(ij), & Len(ij) < 4, \\ Len(ij) - 3, & Len(ij) \geq 4. \end{cases} \tag{5.2}$$

When V is the last part of the overlap nodes of F_i, no other factors can cause additional delay to F_i. Hence, the schedulability of F_i is unaffected by the priority inversion of F_i and F_j when it satisfies:(1) $r_i + D(ij) \leq d_i$; and (2) V is the last part of the overlap nodes of F_i. □

Based on Theorem 5.2, PIA accelerates the unscheduled flow until it can be scheduled or no longer influences the other flow's schedulability. However, Theorem 5.2 is not suitable for all overlaps. When F_i involves several parts of overlaps, we can only guarantee the acceleration of the low priority flow on the last overlap. For the flow's intersections with F_i in the other parts of the overlaps, we define an Effective Overlap Region(EOR) as follows:

Definition 5.2 (Effective Overlap Region) When F_i and F_j conflict in node set $V = V_1, V_2 \ldots, V_h$ $(i < j)$, the overall node set V is the effective overlap region. If F_i and F_j have an overlap but do not conflict, then V is not an effective overlap region.

Hence, Theorem 5.2 can be extended as follows:

Theorem 5.3 *When F_i and F_j conflict at node set $V = V_1, V_2 \ldots, V_h$ $(i < j)$, the schedulability of F_i is unaffected by the priority inversion of F_i and F_j when it satisfies:(1) $r_i + D(ij) \leq d_i$; (2) it does not change the character of non-EORs after V; and (3) V is the last EOR of F_i.*

Proof Based on Theorem 5.2 we extend priority inversion to the last part of the EOR of F_i. When the operation satisfies (1) $r_i + D(ij) \leq d_i$; (2) it does not change the character of non-EORs after V; and (3) V is the last part of the EOR of F_i. Then, the schedulability of F_i and its intersection flows are unaffected by the wait for F_j

to transmit. Hence, when F_i and F_j conflict at node set $V = V_1, V_2 \ldots, V_h$ $(i < j)$, the schedulability of F_i is unaffected by the priority inversion of F_i and F_j when it satisfies:(1) $r_i + D(ij) \leq d_i$; (2) it does not change the character of non-EORs after V; and (3) V is the last part of the EOR of F_i. □

To guarantee the second constraint condition in Theorem 5.3, we denote the number of overlaps after the last EOR in F_i as o_i. For any flow k, when F_k and F_i have a path overlap after the last EOR in V, we denote this overlap path as o_{ik}. Hence, the fact that the second condition in Theorem 5.3 can be translated into the priority inversion operation does not affect the schedulability of F_i and F_k. Obviously, F_k and F_i cannot conflict when the packets generated by the two flows arrive such that o_{ik} satisfies

$$\|r_i^{o_{ik}} - r_k^{o_{ik}}\| > Dij, \tag{5.3}$$

where $r_i^{o_{ik}}$ is the time at which the packet generated by F_i reaches o_{ik} (it is easy to obtain o_{ik} from the network controller). When the arrival time difference of F_i and F_k is larger than Dij, o_{ik} is not an EOR. By updating $r_i^{o_{ik}}$, $i \in F$, we can determine whether to invert the priority of F_i and F_j.

Based on the above discussion, PIA can reduce the number of transmission conflicts before adding MRI nodes to the system. Furthermore, PIA can also improve the schedulability of networks under a high-criticality mode. When the system cannot be scheduled under PIA, the nodes at flow intersections will be replaced by MRI nodes to guarantee the schedulability of the network. The pseudo code of PIA is as follows,

Algorithm 5.3 Priority inversion algorithm

Input: the characters for each flow F_i; the candidate node set $\Lambda = \{\lambda_i\}$, $i \in F$.
Output: reduce the number of transmission conflicts by optimal slot allocations;

```
reallocation slots for each node by FP.
the number of MRI nodes τ = 0.
while the system cannot be scheduled do
    for each flow i do
        obtain each flow's o_i, Len and the corresponding o_ik, i, k ∈ F;
    end for
    for each unscheduled flow j do
        if flow i satisfies Theorem 5.3 then
            invert the priority between F_i and F_j;
        else
            replace the overlap nods with MRI nodes;
            τ++;
        end if
    end for
end while
return τ;
```

PIA first allocates slots under the FP policy (line 1) and initializes the number of MRI nodes (line 2). Then, it judges whether the system can be scheduled or not. If the system can be scheduled, it returns τ, otherwise, PIA accelerates the transmission speed of unscheduled flows by Theorem 5.3. When the flows do not satisfy the conditions, PIA replaces the overlap nodes with MRI nodes, similar to SAA, and increments τ by one (lines 3–16).

When the system performs the priority inversion operation at the last EOR, the conflict is removed. Hence, that part of the overlap is no longer an EOR. The system repeats this process until either the system can be scheduled or no flow satisfies Theorem 5.3.

5.4.4 Algorithm Analysis in High-Criticality Mode

In this subsection, we analyze the schedulability of SAA and PIA. Since both SAA and PIA can be scheduled by adding MRI nodes in low-criticality mode, we analyze only the schedulability in high-criticality mode.

Theorem 5.4 *PIA has a higher schedulability than SAA when the system is in high-criticality mode.*

Proof When the system switches to high-criticality mode, low-criticality flows are abandoned and the high criticality flow period changes to $\aleph_i t_i$, $\frac{c_i}{t_i} \leq \aleph_i < 1$. Because $t_i = d_i$, the flow must arrive at its destination before $\aleph_i d_i$. We analyze the schedulability by a discussion of classification. The flows are classified into two categories: (1) those in which the flow does not overlap other high-criticality flows and (2) those in which the flow overlaps other high-criticality flows. For F_i in the first category, the schedulability of both SAA and PIA are identical. When a flow is not blocked by other flows it can be scheduled when

$$r_i \leq \aleph_i d_i. \tag{5.4}$$

For F_j in the second category, F_j is delayed by other flows in $D(j)$ slots. Then, F_j can be scheduled when

$$r_j + D(j) \leq \aleph_j d_j. \tag{5.5}$$

We denote the number of MRI nodes in SAA and PIA as τ_j^{SAA} and τ_j^{PIA}, respectively. Because PIA reduces the number of transmission conflicts before replacing intersection nodes with MRI nodes, $\tau_j^{SAA} \geq \tau_j^{PIA}$. By Theorem 5.2, we can obtain

$$D(j) = \sum_{k=1}^{\tau_j} D(ij), \tag{5.6}$$

when F_i and F_j conflict, $D(ij) = \begin{cases} Len(ij), & Len(ij) < 4 \\ (Len(ij) - 3), & Len(ij) \geq 4 \end{cases}$, which is determined only by the length of the overlap. Then, we can obtain the relationship between SAA and PIA as follows:

$$r_j^{PIA} + D(j)^{PIA} \leq r_j^{SAA} + D(j)^{SAA}. \tag{5.7}$$

Hence, PIA has a higher schedulability than SAA when the system is in high-criticality mode. □

5.5 Performance Evaluations

We evaluate the performance of our proposed methods by experiments. We compare our approaches with the traditional FP algorithm without MRI nodes. We compare both the acceptance rate and the number of MRI nodes for each criticality mode. We use the acceptance rate to represent the schedulability of a network. When all flows can be scheduled, the acceptance rate is 1; otherwise, it is 0. To control the workload of the entire system, the simulations use the UUniFast algorithm, which can make the flows neither pessimistic nor optimistic for the analysis [23]. All algorithms are implemented in C language. These programs run on a Windows machine with 3.2 GHz CPU and 8 GB memory. Some simulation parameters are summarized in Table 5.1.

5.5.1 Low-Criticality Mode

We first compare the performances of the algorithms in low-critical mode. As Fig. 5.6(a) shown, the system acceptance rate is decreased by the FP scheduling policy (n=15, N=50, m=16), which occurs because the idle resources decrease as the system utilization increases. The latency tolerance of a packet is reduced along

Table 5.1 Simulation parameters

Parameter	Description
n	The number of flows
N	The number of nodes
m	The number of channels
U	Network utilization
u_i	Flow i's utilization
t	The period of flow
d	The deadline of flow
c	The number of transmission hops

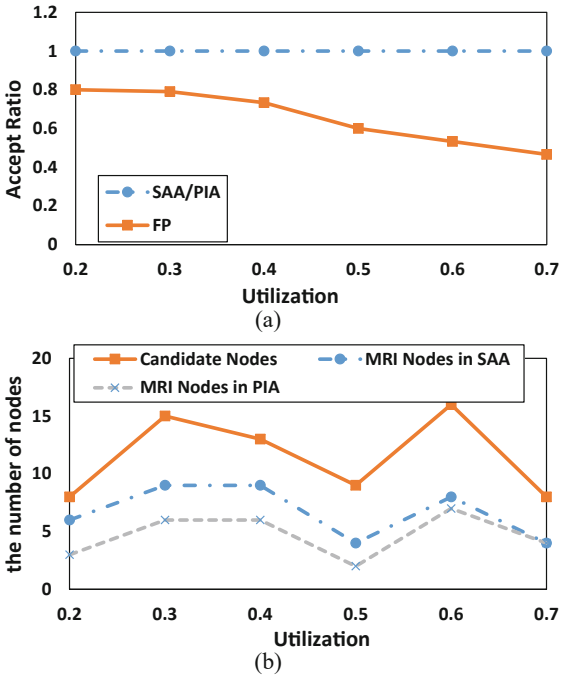

Fig. 5.6 Relationship between the acceptance rate/the number of MRI nodes and system utilization. (**a**) Acceptance rate. (**b**) The number of nodes

with the idle resources. The network can be scheduled under both SAA and PIA in any situation by increasing the number of MRI nodes when $m \geq n$. Figure 5.6b shows the relationship between the number of MRI nodes and utilization. Obviously, both SAA and PIA can reduce the number of MRI nodes under the premise that the system is schedulable. Because PIA optimizes slot allocation before the node replacement operation, PIA uses fewer MRI nodes than SAA. However, none of the curves have an obvious tendency because system utilization involves not only the flow period (t) but also the number of transmission hops (c). We need to regenerate the transmission path for each flow to satisfy the system utilization requirements. Hence, the number of candidate nodes goes up and down. Because an MRI node is selected from the candidate node set, the number of MRI nodes is always less than the number of candidate nodes.

We repeat this simulation for the situation in which all flows can be scheduled as shown in Fig. 5.7. The number of candidate nodes is fixed when we increase system utilization by only adjusting the flow period (because there is only one test in this simulation, and the flow transmission paths do not vary). Initially, no MRI nodes exist in the system because they can be scheduled without any MRI nodes. However, when we increase the system utilization, transmission conflict occurs. To guarantee the schedulability of the system, we need to add MRI nodes to the system. In addition, the number of MRI nodes required by PIA is always less than the number required by SAA. This occurs because by optimizing slot allocation,

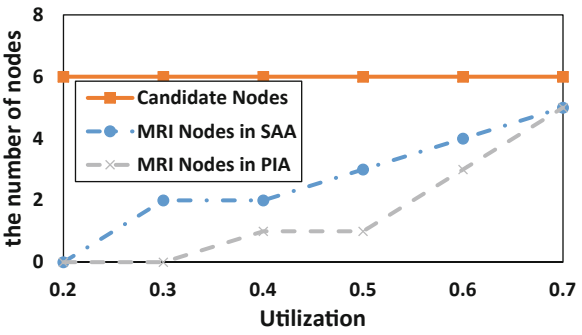

Fig. 5.7 Relationship between the number of MRI nodes and system utilization

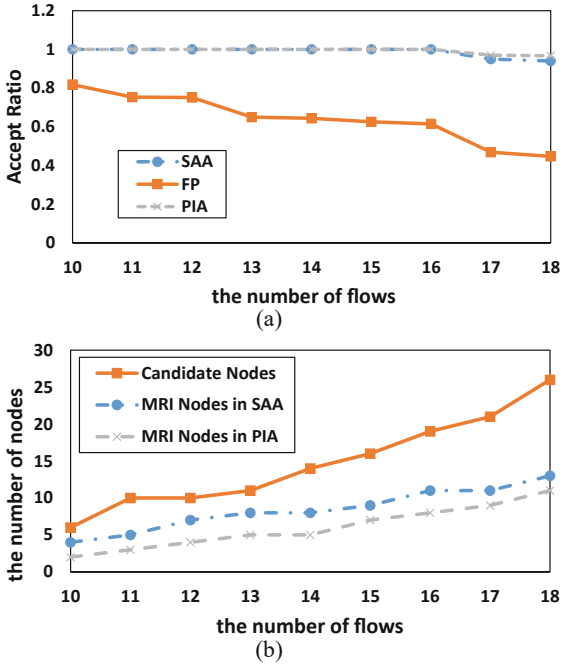

Fig. 5.8 Relationship between the acceptance rate/number of MRI nodes and the number of flows. (**a**) Acceptance rate. (**b**) The number of nodes

PIA can reduce the number of transmission conflicts. Hence, the number of MRI nodes required by PIA is always less than the number required by SAA.

The relationship between the acceptance rate/number of MRI nodes and the number of flows is shown in Fig. 5.8 (U=0.3, N=50, m=16). The acceptance rate is reduced along with the number of flows under the FP scheduling policy. The acceptance rates of SAA and PIA are reduced when the number of flows becomes larger than the number of channels. This result occurs because when $m < n$, delays are caused by channel contention. PIA achieves a better performance than SAA when $n > 16$ because it uses fewer MRI nodes. In addition, both the number of candidate nodes and the number of MRI nodes increase as the number of flows increases. This result occurs because the number of intersections increases when

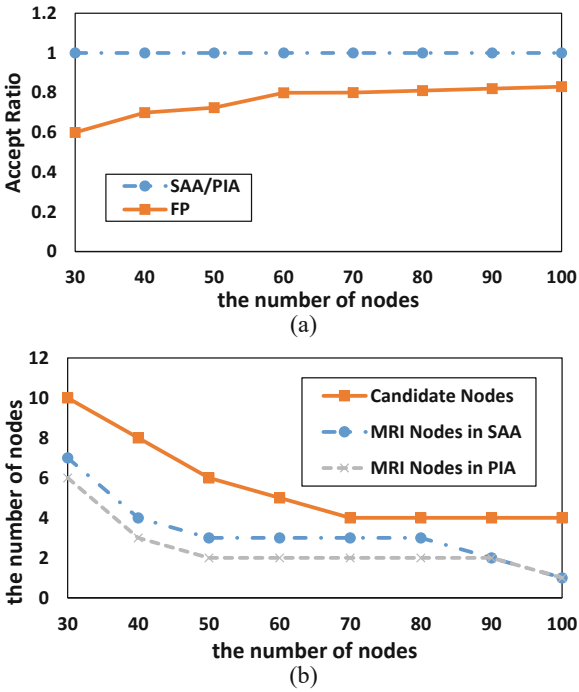

Fig. 5.9 Relationship between the acceptance rate/number of MRI nodes and the number of flows. (**a**) Acceptance rate. (**b**) The number of nodes

the number of flows increases, causing more transmission conflicts in the system. Thus, the system requires more MRI nodes to guarantee its schedulability, and the number of MRI nodes always satisfies $\tau^{FP} > \tau^{SAA} > \tau^{PIA}$.

Figure 5.9 shows the relationship between the acceptance rate/number of MRI nodes and the number of nodes (U=0.3, n=15, m=16). All these results indicate that SAA and PIA can guarantee the schedulability of the system. The number of MRI nodes in both SAA and PIA is no larger than the number of candidate nodes regardless of the conditions. In addition, when the number of nodes increases to 90, the number of MRI nodes in PIA and SAA is identical because PIA transmutes into SAA when it cannot resolve the conflict.

5.5.2 *High-Criticality Mode*

When the system switches to high-critical mode, the number of MRI nodes is no longer our main concern (the number of MRI nodes is the same as in low-critical mode). Instead, we are more concerned with the performances of SAA and PIA in high-critical mode.

Figure 5.10 shows the relationship between the system acceptance rate and system parameters such as the number of nodes and the number of flows. In these

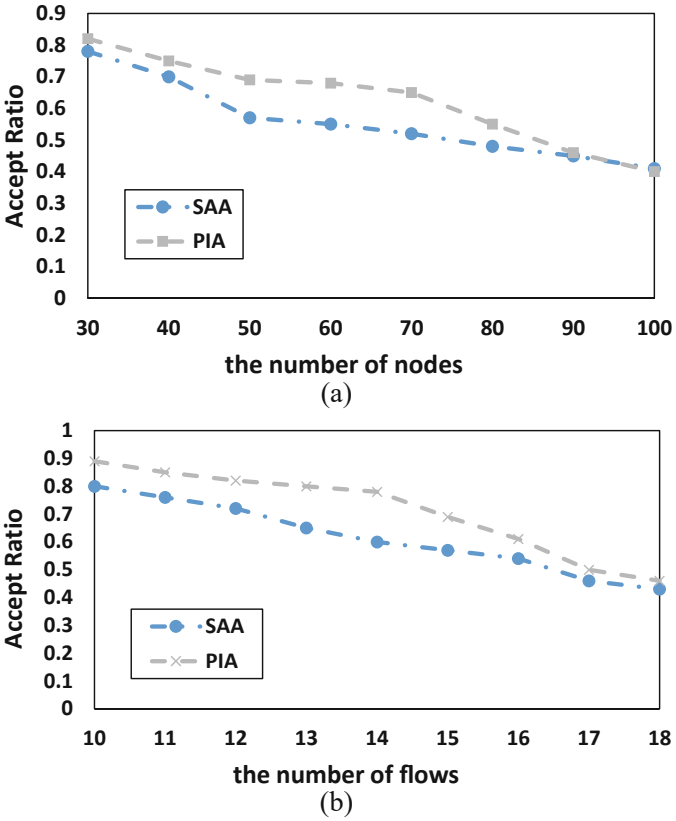

Fig. 5.10 Relationship between the acceptance rate and system parameters in high-critical mode. (**a**) Acceptance rate. (**b**) The number of nodes

conditions, we set $\aleph = 0.9 > \max\{\frac{c_i}{t_i}, i \in F\}$. If $\max\{\frac{c_i}{t_i}, i \in F\} \geq 0.9$, we will regeneration the system. Figure 5.10 illustrates (as Theorem 5.4 proves) that the acceptance rate in PIA is always better than in SAA.

Figure 5.11 shows the relationship between the system acceptance rate and system utilization. To study the relationship between these two elements, we increase system utilization by adjusting only the flow period. Subsequently, we obtain the same result as Fig. 5.10, which also verifies the correctness of Theorem 5.4.

5.6 Summary

In this chapter, we first introduce MRI nodes into mixed-criticality networks. Then, we analyze the transmission paths and obtain the candidate node set. Next, based on the characteristics of MRI nodes, we propose SAA and PIA to guarantee

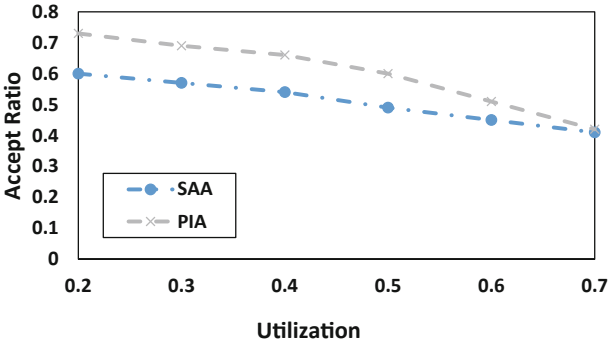

Fig. 5.11 Relationship between the number of MRI nodes and system utilization in high-critical mode

the network schedulability in low-criticality mode. By considering system cost, these two algorithms help to reduce the number of MRI nodes used. Finally, we analyze the schedulability of these two algorithms when the system switches to high-criticality mode. The simulation results show that our scheduling algorithms perform better than existing scheduling policies.

References

1. Lu CY, Saifullah A, Li B, Sha M, Gonzalez H, Gunatilaka D, Wu CJ, Nie LS, Chen YX (2016) Real-time wireless sensor-actuator networks for industrial cyber-physical systems. P IEEE 104(5):1013–1024
2. Lee EA, Seshia SA (2016) Introduction to embedded systems: a cyber-physical systems approach. MIT Press, Cambridge
3. Xiao HQ, Kong LS, Yuan CL, Xiao SP (2016) Stochastic optimization model and solution algorithm for blending procedure of process industrial. Inf Control 45(1):40–44
4. Saifullah A, Xu Y, Lu CY, Chen YX (2015) End-to-end communication delay analysis in industrial wireless networks. Trans Comput 64(5):1361–1374
5. Gesbert D, Shafi M, Shiu D, Smith PJ, Naguib A (2003) From theory to practice: an overview of MIMO space-time coded wireless systems. J Sel Area Commun 21(3):281–302
6. Hoydis J, Ten BS, Debbal M (2013) Massive MIMO in the UL/DL of cellular networks: how many antennas do we need? J Sel Area Commun 31(2):160–171
7. Lu L, Li G, Swindlehurst A, Ashikhmin A, Zhang R (2014) An overview of massive MIMO: benefits and challenges. J Sel Top Signal Process 8(5):742–758
8. Kong LH, Liu X (2015) mZig: enabling multi packet reception in zigBee. In: The annual international conference on mobile computing and networking. ACM, New York, pp 552–565
9. Hithnawi A, Li S, Shafagh H, Gross J, Duquennoy S (2016) Crosszig: combating cross-technology interference in low-power wireless networks. In: The international conference oninformation processing in sensor networks (IPSN). IEEE, Piscataway, pp 1–12
10. Cheng L, Gu Y, Niu JW, Zhu T, Liu C, Zhang Q, He T (2016) Taming collisions for delay reduction in low-duty-cycle wireless sensor networks. In: The IEEE international conference on computer communications. IEEE, Piscataway, pp 1–9

11. Yuan Y, He ZH, Chen M (2006) Virtual MIMO-based cross-layer design for wireless sensor networks. IEEE Trans Veh Technol 55(3):856–864
12. Zhao M, Yang YY, Wang C (2015) Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks. IEEE Trans Mobile Comput 14(4):770–785
13. Liang W, Zhang XL, Xiao Y, Wang FQ, Zeng P, Yu HB (2011) Survey and experiments of WIA-PA specification of industrial wireless network. Wireless Commun Mobile Comput 11(8):1197–1212
14. Zu KK, DeLamare B, Haardt M (2014) Multi-branch Tomlinson-Harashima precoding design for MU-MIMO systems: theory and algorithms. Trans Commun 62(3):939–951
15. Jin X, Wang JT, Zeng P (2015) End-to-end delay analysis for mixed-criticality wirelesshart networks. IEEE/CAA J Autom Sin 2(3):282–289
16. Baruah S, Li HH, Stougie L (2010) Towards the design of certifiable mixed-criticality systems. In: Real-time and embedded technology and applications symposium (RTAS). IEEE, Piscataway, pp 13–22
17. Baruah S, Bonifaci V, D'Angelo G, Li HH, Marchetti-Spaccamela A, Megow N, Stougie L (2012) Scheduling real-time mixed-criticality jobs. Trans Comput 61(8):1140–1152
18. Ekberg P, Wang Y (2012) Bounding and shaping the demand of mixed-criticality sporadic tasks. In: Euromicro conference on real-time systems (ECRTS). IEEE, Piscataway, pp 135–144
19. Huang PC, Yang H, Thiele L (2014) On the scheduling of fault-tolerant mixed-criticality systems. In: The 51st annual design automation conference. ACM, New York, pp 1–6
20. Huang WH, Chen JJ (2016) Self-suspension real-time tasks under fixed-relative-deadline fixed-priority scheduling. In: The conference on design, automation & test in Europe, pp 1078–1083
21. Xia CQ, Jin X, Kong LH, Zeng P (2017) Bounding the demand of mixed-criticality industrial wireless sensor networks. IEEE Access 5:7505–7516
22. Saifullah A, Xu Y, Lu CY, Chen YX (2015) End-to-end communication delay analysis in industrial wireless networks. Trans Comput 64(5):1361–1374
23. Bini E, Buttazzo CC (2005) Measuring the performance of schedulability tests. Real-Time Syst 30(1):129–154