

# Chapter 3

## Schedulability Analysis of Mixed-Criticality Data Under EDF Scheduling



**Abstract** In this chapter, to improve the schedulability of high-criticality flows when the network is running, we present a supply/demand bound function analysis method based on earliest deadline first (EDF) scheduling. In addition, our method considers both source routing and graph routing. At the beginning, when the network is in low-criticality mode, source routing is applied. When errors or exceptions occur, the network switches to high-criticality mode, and network routing turns to graph routing to guarantee that critical flows can be scheduled. By estimating the demand bound for the mixed-criticality data model, we can determine the schedulability of industrial wireless networks.

### 3.1 Background

Graph routing [1] as an effective way to improve network reliability has been widely used in recent years. A network under graph routing allocates two dedicated time slots for each transmission; if the first transmission fails, a retransmission will be sent. Furthermore, the controller assigns a third shared slot on a separate path for another retransmission. Since graph routing is a reliable method to handle transmission failures, a few works have begun to focus on graph routing. The work in [2] presents the first worst-case end-to-end delay analysis for periodic real-time flows under reliable graph routing. The work in [3] studies the network lifetime maximization problem under graph routing. However, graph routing introduces great challenges for real-time analysis. Many conflicts are generated on a large number of transmission tasks. Obviously, the task which is more critical but has a low priority may miss its deadline in this situation. However, many systems need to guarantee high-criticality task's schedulability even though in the worst case. That is really very important in many scenarios such as industrial production

line, vehicle driving system, etc. To improve the schedulability of high-criticality flows when the network is running, we introduce resource analysis into mixed-criticality industrial wireless networks. Mixed-criticality network can improve the schedulability of high-criticality flows by dynamically switching the network criticality, and resource analysis is a major way to analyze the schedulability in real-time systems. Combining mixed-criticality network and resource analysis, we can estimate the schedulability of networks with different critical levels.

In this chapter, we propose a novel industrial network model with EDF scheduling. Our objective is to improve the network reliability, especially for high-criticality flows to arrive at their destinations on time even though in the worst case. We analyze the network schedulability by the method of resource analysis. The network is reliable when the network resource supply is no less than the network upper demand in any length of time slot. The main challenges in our work are (1) how to evaluate network demand when a network switches from low-criticality mode to high-criticality mode and (2) how to tighten the network demand bound function to ensure that the analysis result is not too pessimistic. The network we focus on, in the beginning, works in low-criticality mode, and the flows transmit under the EDF policy [4] and source routing. The packets are transmitted from the source to the destination on the primary paths; when an error occurs or the demand changes, the network switches to high-criticality mode to enhance the schedulability of high-criticality flows. The network substitutes reliable graph routing for source routing. Furthermore, we present a supply/demand bound analysis method to analyze the schedulability of periodic flows in industrial wireless sensor networks. By comparing the relationship between network supply bound and demand, we can predict whether the network can be scheduled. The current study makes the following key contributions:

1. We propose a mixed-criticality industrial network, in which network routing switches from source routing to graph routing when the criticality mode changes.
2. We theoretically derive the supply/demand bound function as a novel analysis method for industrial wireless networks. By analyzing channel contention and transmission conflict, we obtain the upper-bound function of demand in any length of time slot. When given a network supply bound function, we can determine the schedulability of flows under different criticality modes.
3. We tighten the demand bound by analyzing *carry-over jobs* (which are released but not finished at the switching slot) and discussing the number of conflicts between two flows.
4. Our method can be applied for general networks. By calculating the maximum demand bound of networks, we can analyze network schedulability in the system design stage; after network deployment, the upper bound of communications can be obtained by our method.

## 3.2 System Model

We consider an industrial wireless network consisting of field devices, one gateway, and one centralized network manager. Our network is proposed in three aspects. We first propose a network model that is abstracted from mainstream industrial network standards. Then, we introduce a mixed-criticality network. Finally, we apply EDF scheduling in the industrial network.

### 3.2.1 Network Model

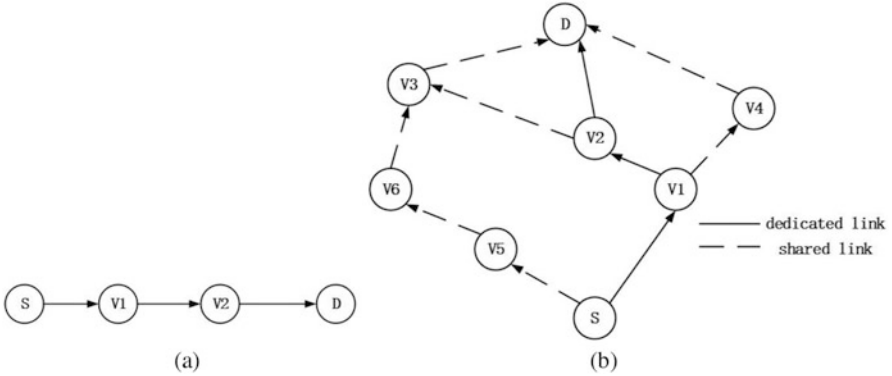
Without loss of generality, our model has the same salient features as WirelessHART and WIA-PA, which make it particularly suitable for process industries:

*Time Division Multiple Access (TDMA)* In industrial wireless sensor networks, time is synchronized and slotted. Because the length of a time slot allows exactly one transmission, TDMA protocols can provide predictable communication latency and real-time communication.

*Route and Spectrum Diversity* To mitigate physical obstacles, broken links, and interference, the messages are routed through multiple paths. Spectrum diversity gives the network access to all 16 channels defined in the IEEE 802.15.4 physical layer and allows per-time slot channel hopping. The combination of spectrum and route diversity allows a packet to be transmitted multiple times, over different channels and different paths, thereby handling the challenges of network dynamics in harsh and variable environments at the cost of redundant transmissions and scheduling complexity [5].

*Handling Internal Interference* Industrial networks allow only one transmission in each channel in a time slot across the entire network, thereby avoiding the spatial reuse of channels. Thus, the total number of concurrent transmissions in the entire network at any slot is no greater than the number of available channels.

With the above features, the network can be modeled as a graph  $G = (V, E, m)$ , in which the node set  $V$  represents the network devices (all sensor nodes in our model are fixed),  $E$  is the set of edges between these devices, and  $m$  is the number of channels. Network routing is shown in Fig. 3.1; our model supports both source routing and graph routing. Source routing is well known in academic research; we will not explore it in this article. Graph routing is a unique feature of industrial



**Fig. 3.1** Network routing. (a) Source routing. (b) Graph routing

wireless sensor networks. In graph routing, a routing graph is a directed list of paths that connect two devices. As shown in Fig. 3.1b, graph routing has a primary path and multiple backup paths. This provides redundancy in the route and improves the reliability. As stated in the standard of WirelessHART, for each intermediate node on the primary path, a backup path is generated to handle link or node failure on the primary path. The network manager allocates  $\alpha$  dedicated slots, a transmission and  $(\alpha - 1)$  retransmission on the primary path. A  $(\alpha + 1)$ th shared slot is allocated on the backup path, usually  $\alpha = 2$ . In a dedicated slot, one channel only allows one transmission. However, for the case of a shared slot, the transmissions having the same receiver can be scheduled in the same slot. The senders that attempt to transmit in a shared slot contend for the channel using a carrier sense multiple access with collision avoidance (CSMA/CA) scheme [2]. Hence, multiple transmissions can be scheduled in the same channel to contend in a shared slot. For instance, the network manager allocates two dedicated slots for the packet transmits from node  $S$  to node  $V_1$  in Fig. 3.1b. After the transmissions on the primary path, a third slot is allocated for the packet transmits from node  $S$  to  $V_5$  as a backup path. When two backup paths intersect at node  $V_3$ , they can avoid collision by CSMA/CA.

It is important to note that the receiver responds with an ACK packet before retransmission and backup; the sender retransmits or sends a backup packet when it does not receive an ACK. Because ACK is a part of the transmission, we do not need to especially analyze the demand of ACK.

### 3.2.2 Mixed-Criticality Network

A periodic end-to-end communication between a source and a destination is called a *flow*. Network switch instruction is a part of the control flow. Because we analyze network total demand, we need not distinguish whether a flow is a data flow or

a control flow. The total number of flows in the network is  $n$ , denoted by  $F = \{F_1, F_2, \dots, F_n\}$ .  $F_i$  is characterized by  $\langle t_i, d_i, \xi, c_i, \phi_i \rangle$ ,  $1 \leq i \leq n$ , where  $t_i$  is the period;  $d_i$  is the deadline;  $\xi$  is the criticality level (we focus on dual-criticality network  $\{LO, HI\}$ );  $\xi = 2$ , means the network allocates two slots, one transmission and one retransmission. Our model can be easily extended to networks with an arbitrary number of criticality levels (by increasing the number of retransmissions on the primary path);  $c_i$  is the number of hops required to deliver a packet from source to destination. When the network mode switches to high criticality, we denote the total transmission hops of both the primary path and shared paths as  $C_i$ ; and  $\phi_i$  is the routing path of the flow. Thus, we can describe each flow  $F_i$  as follows.  $F_i$  periodically generates a packet at its period  $t_i$ , and then sends it to the destination before its deadline  $d_i$  via the routing path  $\phi_i$  with  $c_i$  hops.

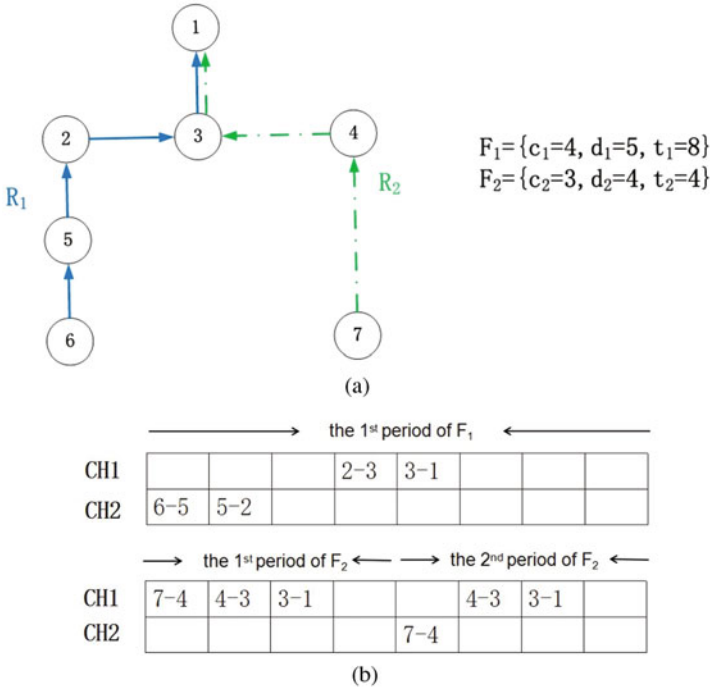
In the beginning, messages are transmitted under source routing in low criticality. When an error occurs or the demand changes, the control flow will send a switch instruction, and the network will switch to high-criticality mode. To enhance network reliability, the messages are transmitted under graph routing when the network is running on high-criticality mode. This is an irreversible process; high-criticality mode will never switch back to low-criticality mode (the analytical method of irreversible processes is similar to criticality mode switches from low to high). After the switch, we are not required to meet any deadlines for low-criticality flows, but high-criticality flows may instead execute for up to their high-criticality level characters.

### 3.2.3 EDF Scheduling in Industrial Networks

In this subsection, we provide an overview of the earliest deadline first scheduling under industrial wireless sensor networks to analyze network schedulability. EDF scheduling is a commonly adopted policy in practice for real-time CPU scheduling, cyber-physical systems, and industrial networks [6]. In an EDF scheduling policy, each job priority is assigned by its absolute deadline, and the transmission is scheduled based on this priority. Each node in our network is equipped with a half-duplex omnidirectional radio transceiver that can alternate its status between transmitting and receiving. There are two kinds of delay in industrial wireless sensor networks, which can be summarized as follows:

- Channel contention: each channel is assigned to one transmission across the entire network in the same slot.
- Transmission conflicts: whenever two transmissions conflict, the transmission that belongs to the lower-priority job must be delayed by the higher-priority one, regardless of how many channels are available. It is important to note that one node can perform only one operation (receiving or transmitting) in each slot.

In EDF scheduling, the priority is inversely proportional to its absolute deadline. We explain the operating principle of EDF scheduling in Fig. 3.2. There are two



**Fig. 3.2** An example for EDF scheduling. (a) Routing. (b) EDF scheduling

channels(CH1 and CH2) and flows in this network. At the beginning, the priority of  $F_2$  is higher than  $F_1$  since  $d_2 = 4 < d_1 = 5$ . Then the controller allocates CH1 for  $F_2$  first. The flow with lower priority must be delayed when transmission conflict occurs such as  $F_1$  will be delayed by  $F_2$  at the 3rd time slot. At the 5th time slot, the second packet is generated by  $F_2$  with an absolute deadline 8, which is larger than 5. Hence, the priority inversion, and CH1 are allocated to  $F_1$ .

Channel contention occurs when high-priority jobs occupy all channels in a time slot; a transmission conflict is generated when several transmissions involve a common node at the same dedicated slot, and a low-priority job is delayed by high-priority ones. However, for the case of shared slots, transmissions with the same receiver can be scheduled in the same slot. When channel contention occurs between backup paths, the senders on the backup path use a CSMA/CA scheme to contend for the channel, and a network delay will not result in this condition. For a network under graph routing, two paths  $\phi_i$  and  $\phi_j$  involving a common node may conflict in four conditions:

1.  $\phi_i$  is a primary path,  $\phi_j$  is a backup path;
2. both  $\phi_i$  and  $\phi_j$  are primary paths;
3. both  $\phi_i$  and  $\phi_j$  are backup paths;
4.  $\phi_i$  is a backup path,  $\phi_j$  is a primary path.

Except for condition 3, the other three conditions may generate transmission conflicts. Consequently, the total delay caused by these conditions depends on how their primary and backup paths intersect in the network.

In a real-time system, one task is schedulable when it could be executed completely before its deadline. Hence, the flow could be scheduled when all the packets generated by the flow could arrive destination before their relative deadlines. Then we define the network schedulability as whether or not all flows in a network are schedulable.

### 3.3 Problem Formulation

Given a mixed-criticality industrial network  $G = (V, E, m)$ , the flow set  $F$  and the EDF scheduling algorithm, our objective is to analyze the relationship between the maximum execution demand of the flows and network resource in any time interval such that the schedulability of the flow set can be determined. A successful method to analyzing the schedulability of real-time workloads is to use demand bound functions [7, 8]. We introduce this concept into industrial wireless sensor networks and propose two definitions as follows:

**Definition 3.1 (Supply Bound Function)** A supply bound function  $sbf(l)$  is the minimal transmission capacity provided by the network within a time interval of length  $l$ .

**Definition 3.2 (Demand-Bound Function)** A demand bound function  $dbf(F_i, l)$  gives an upper bound on the maximum possible execution demand of flow  $F_i$  in any time interval of length  $l$ , where demand is calculated as the total amount of required execution time of flows with their whole scheduling windows within the time interval.

There are methods for computing the supply bound function  $sbf(l)$  in single-processor systems [9, 10]—for example, a unit-speed, dedicated uniprocessor has  $sbf(l) = l$ . We say that a supply bound function  $sbf$  is of no more than unit speed if

$$sbf(0) = 0 \wedge \forall l, k \geq 0 : sbf(l+k) - sbf(l) \leq k. \quad (3.1)$$

Because each channel can be mapped as one processor, the supply bound function  $sbf$  of the industrial network can be bounded as

$$sbf(0) = 0 \wedge \forall l, k \geq 0 : sbf(l+k) - sbf(l) \leq Ch \times k, \quad (3.2)$$

where  $Ch$  is the number of channels in the network. Furthermore, as a natural assumption of all proposed virtual resource platforms in the literature, we assume that the supply bound function is piecewise linear in all intervals  $[k, k+l]$ . In TDM

(time division multiple), the network supply bound function can be expressed as

$$sbf(l) = \max(l \bmod \Theta - \Theta + \Phi, 0) + \lfloor \frac{l}{\Theta} \rfloor \Phi, \quad (3.3)$$

where  $\Theta$  is the period of TDM, and  $\Phi$  is the length of slots allocated to the transmission.

In different modes, the schedulability of the flow set is determined as follows:

$$\sum_{F_i \in F} dbf_{LO}(F_i, l) \leq sbf_{LO}(l), \forall l \geq 0. \quad (3.4)$$

$$\sum_{F_i \in HI(F)} dbf_{HI}(F_i, l) \leq sbf_{HI}(l), \forall l \geq 0. \quad (3.5)$$

Similar to real-time scheduling, the flow set is scheduled when the network is satisfied by Eqs. (3.4) and (3.5). However, in contrast to real-time scheduling, there are two kinds of delays in industrial wireless sensor networks, channel contention and transmission conflicts. When a transmission conflict occurs, a high-priority job will influence a low priority job, and thus, the flows are not independent.

Note that transmission conflict is a distinguishing feature in industrial wireless sensor networks that does not exist in conventional real-time processor scheduling problems. To analyze the network demand in any time interval, we must consider the delay caused by transmission conflicts.

Moreover, in mixed-criticality networks, there may be some jobs that are released but not finished at the time of the switch to high-criticality mode; we define these jobs as *carry-over jobs*. We must analyze *carry-over jobs* to tighten the demand bound of the network.

### 3.4 Demand-Bound Function of Industrial Networks

In this section, we analyze the network demand bound function for a single-criticality network and mixed-criticality network. For the single-criticality network, we study the demand bound function from channel contention and transmission conflicts. On this basis, we then analyze the delay caused by *carry-over jobs* (the job that is released but not finished at the time of the switch) in the mixed-criticality network. Finally, we study the methods for tightening the network demand bound function.



### 3.4.1 Analysis of Single-Criticality Networks

In this subsection, we study the demand bound function under a single-criticality network in two steps. First, we formulate network transmission conflict delay with path overlaps; we then analyze the network dbf. To make our study self-contained, we present the results of the state-of-the-art demand bound function for CPU scheduling [11, 12]. Assuming that the flows are executed on a multiprocessor platform, the channel is mapped as a processor. We can obtain the network demand caused by channel contention in any time interval  $l$  as

$$dbf(l)^{ch} = \frac{1}{m} \sum_{i=1}^n \left[ \left( \left\lfloor \left\lfloor \frac{l - d_i}{t_i} + 1 \right\rfloor \right) c_i \right) \right]_0. \quad (3.6)$$

Equation (3.6) considers only the delay caused by channel contention, denoted as  $dbf(l)^{ch}$ . The jobs are conflicted when their transmission paths have overlaps. As shown in Fig. 3.3, the priority of the job in  $F_i$  is higher than the one in  $F_j$ , so the job in  $F_j$  may be delayed by in  $F_i$  at nodes  $V$  and  $V_1$  to  $V_h$  (we assume the network is connected and do not consider the case where the path disconnects).

Transmission conflicts are generated at the path overlaps, and the network requires more resources to solve the transmission conflicts. To obtain  $dbf(l)$  of the network, we must first study the relationship between conflict delay and path overlap. However, estimation transmission conflict delay by the length of the overlap is often a pessimistic method. As shown in Fig. 3.3, the delay is much smaller than the length of the path overlap. To avoid pessimistic estimation, we introduce the result proposed by Saifullah in [13]. The length of the  $k$ th path overlap is denoted as  $Len_k(ij)$ , and its conflict delay is  $D_k(ij)$ . For the overlap as  $V_1 \rightarrow \dots V_h$ , if there exists node  $u, w \in V$  such that  $u \rightarrow V_1 \rightarrow \dots V_h \rightarrow w$  is also on  $F_i$ 's route, then  $Len_k(ij) = h + 1$ . If only  $u$  or only  $w$  exists, then  $Len_k(ij) = h$ . If neither  $u$  nor  $w$  exists, then  $Len_k(ij) = h - 1$ . In our example,  $Len_1(ij) = 2$ ,  $Len_2(ij) = 7$  and  $D(ij) = D_1(ij) + D_2(ij)$ , which is at most  $2 + 3 = 5$ . Obviously,  $Len_k(ij)$  is the upper bound of  $D_k(ij)$ , which means  $Len_k(ij) \geq D_k(ij)$ . For the flow set  $F$ , the total delay caused by transmission conflicts  $\Delta$  is

$$\Delta = \sum_{1 \leq i, j \leq n} D_k(ij) \leq \sum_{1 \leq i, j \leq n} Len_k(ij). \quad (3.7)$$

By the Lemma proposed in [13], the estimation of the delay caused by overlap with a length of at least 4 can be tightened. We then formulate the total transmission conflicts between  $F_i$  and  $F_j$  as

$$\Delta(ij) = \sum_{k=1}^{\delta(ij)} Len_k(ij) - \sum_{k'=1}^{\delta'(ij)} (Len_{k'}(ij) - 3), \quad (3.8)$$



**Proof** At the beginning, the priority of  $F_i^k$  is higher than  $F_j^g$ , which means  $d_i^k < d_j^g$ . As Fig. 3.3 shows, two flows may conflict at  $V_1$ , and  $F_j$  is delayed by  $F_i$ . When  $F_i^k$  is forwarded to  $V_h$ , two jobs may conflict again. If  $F_i^k$  is blocked by  $F_j^{k+m}$ , we can obtain  $d_i^k > d_j^{g+m}$ . Because  $d_j^{g+m} > d_j^g$ , this contradicts with  $d_i^k < d_j^g$ . Hence,  $F_i^k$  will never be blocked by  $F_j^{g+m}$ .

We prove that  $F_i^{k+m}$  is blocked by  $F_j^g$  through an example. We use the following simple flow set:  $F_1 = \{c_1 = 1, d_1 = t_1 = 2\}$  and  $F_2 = \{c_2 = 1, d_2 = t_2 = 3\}$ .

At the beginning, the priority of  $F_1^1$  is higher than  $F_2^1$ , because the absolute deadline is 2 and 3, respectively. At time slot 2, another job is generated by  $F_1$  with the absolute deadline of 2. However, the absolute deadline of  $F_2^1$  is 1,  $F_1^2$  is blocked by  $F_2^1$ . Hence,  $F_i^{k+m}$  can be blocked by  $F_j^g$ .  $\square$

Because a path is a chain of transmissions from source to destination, in considering the conflict delay caused by multiple jobs of  $F_i$  on flow  $F_j$ , we analyze the number of conflicts for  $F_i$  and  $F_j$ . Thus, Lemma 3.2 establishes the upper bound of this value.

**Lemma 3.2** *When  $F_j$  and  $F_i$  conflict, within any time interval of length  $l$ , each job of  $F_j$  can be blocked no more than  $\lceil \frac{l}{t_i} \rceil$  times, and  $F_j$  can be blocked by  $F_j$  no more than  $\lceil \frac{l}{t_j} \rceil$  times.*

**Proof** Based on Lemma 3.1, we know that the priority inversion will occur in the process of transmission. If  $F_i^k$  is a higher-priority job than  $F_j^g$ , the jobs released after  $F_j^g$  must be blocked by  $F_i^k$  until  $F_i^k$  is finished. If all jobs generated by  $F_i$  satisfy  $d_i^{k+\lceil \frac{l}{t_i} \rceil} < d_j^g$ , where  $k$  and  $g$  are the first jobs for  $F_i$  and  $F_j$ , respectively, in  $l$ , then there are no more than  $\lceil \frac{l}{t_i} \rceil$  jobs of  $F_i$ . Beyond that, because there is no transmission conflict, the other jobs of  $F_j$  are not blocked by  $F_i$ . Hence,  $F_j$  can be blocked by  $F_i$  no more than  $\lceil \frac{l}{t_i} \rceil$  times. The same as  $F_i$ ,  $F_i$  can be blocked by  $F_j$  no more than  $\lceil \frac{l}{t_j} \rceil$  times.  $\square$

By Lemmas 3.1 and 3.2, we can estimate the network demand caused by the transmission conflict. Based on Eq. (3.6), we obtain the upper bound of  $dbf(l)$  as follows:

**Theorem 3.1** *In any time interval of length  $l$ , the demand bound function under a single-critical network (low-criticality mode) is upper-bounded by*

$$dbf_{LO}(l) = \frac{1}{m} \sum_{i=1}^n \left[ \left( \left\lfloor \frac{l - d_i}{t_i} + 1 \right\rfloor \right) c_i \right]_0 + \sum_{1 \leq i, j \leq n} (\Delta(i, j) \max\{\lceil \frac{l}{t_i} \rceil, \lceil \frac{l}{t_j} \rceil\}). \quad (3.9)$$

**Proof** Network demand is the upper bound in a time interval of length  $l$ , which consists of two parts, channel contention and transmission conflict. The demand of channel contention is bounded by Eq. (3.6). For the demand of the transmission conflict, we first analyze each time conflict delay for every two paths by Eq. (3.8); the number of conflicts can then be obtained by Lemma 3.2. We can obtain the network demand of transmission conflict as

$$\sum_{1 \leq i, j \leq n} (\Delta(ij)) \max\{\lceil \frac{l}{t_i} \rceil, \lceil \frac{l}{t_j} \rceil\}. \quad (3.10)$$

Hence, we can obtain the demand bound function under a single-critical network upper-bounded by Eq. (3.9).  $\square$

### 3.4.2 Analysis of Mixed-Criticality Networks

Based on the result proposed in Sect. 3.4.1, we extend the idea of a demand bound function to a mixed-criticality network. For illustration purposes, only a dual-criticality network is considered; this means that  $\xi$  has only two values, *LO* (low-criticality mode) and *HI* (high-criticality mode). Nevertheless, it can be easily extended to networks with an arbitrary number of criticality modes. We construct three demand bound functions: the demand bound function in low- and high-criticality modes ( $dbf_{LO}(l)$  and  $dbf_{HI}(l)$ ) and the demand bound function when network mode switches ( $dbf_{LO2HI}(l)$ ). We analyze  $dbf_{HI}(l)$  and  $dbf_{LO2HI}(l)$  under graph routing in this subsection.

The network begins from the low-criticality level, and all flows are served and executed as in a single-criticality network. When errors or emergencies occur, the centralized network manager will trigger the switching of the network mode from *LO* to *HI*. In high-criticality mode, the network turns to graph routing, and the flows in the low-criticality level are discarded; only high-criticality flows can be delivered. The job that is active (released, but not finished) from a high-criticality flow at the time of the switch is still running under source routing;  $n_{HI}$  is the number of high-criticality flows, and there are no more than  $n_{HI}$  carry-over jobs that are active at the time of the switch. We define these carry-over jobs as new flows  $F_{(n_{HI}+1)}, F_{(n_{HI}+2)} \dots F_{2n_{HI}}$ , which have the same characters as the corresponding flows in  $F$  except for  $c$  and  $t$ . For the new flow  $F_{p+n_{HI}}$ ,  $c_p > c_{(p+n_{HI})}$ , and as an accidental event,  $t_{(p+n_{HI})} \gg t_p$ .

When the network switches from *LO* to *HI*, the demand of carry-over jobs is

$$\frac{1}{m} \sum_{p=1+n_{HI}}^{2n_{HI}} c_p + \sum_{n_{HI} \leq p, q \leq 2n_{HI}} \Delta(pq). \quad (3.11)$$

Furthermore, the flows will generate new jobs when the network switches to high-criticality mode. Because each node except the destination on the primary path generates one backup path, the total number of paths for  $F_p$  is  $c_p + 1$  and the execution time for each backup path  $c_p^k$  can be obtained from the network easily. The total execution time of  $F_i$  can be denoted as  $C_p = c_p + \sum_{k=1}^{c_p} c_p^k$ . Therefore, network demand for channel contention under graph routing is

$$dbf_{HI}^{ch}(l) = \frac{2}{m} \sum_{p=1}^{n_{HI}} \left[ \left( \left\lfloor \frac{l - d_p}{t_p} + 1 \right\rfloor \right) C_p \right]_0. \quad (3.12)$$

Based on the rules of transmission conflict proposed in Sect. 3.2.3, a transmission conflict between two flows is generated only if there is at least one flow transmission on the primary path. Therefore, we analyze  $dbf_{HI}(l)$  by studying the transmission conflict generated on the primary path. For  $F_p^s$  and  $F_q^m$ , when given  $d_p < d_q$ ,  $F_q^m$  may be delayed by  $F_p^s$  and its backup paths. We denote the path set of  $F_p$  and its backup paths as  $I$ ; each path in  $I$  is denoted as  $p'$ . The upper bound delay of  $F_q^m$  caused by  $F_p^s$  is denoted as  $\Delta(p'q)$ .  $\Delta(p'q)$  can be formulated as

$$\Delta(p'q) = \sum_{p'=1}^{c_p+1} \left( \sum_{k=1}^{\delta(p'q)} Len_k(p'q) - \sum_{k'=1}^{\delta'(p'q)} (Len_{k'}(p'q) - 3) \right). \quad (3.13)$$

For the job on the backup path, a transmission delay occurs only when it conflicts with primary paths with high-priority jobs. When we reverse the priority of  $F_p^s$  and  $F_q^m$ , Eq. (3.13) is the upper bound additional demand of  $F_p^s$  caused by  $F_q^m$ . From the above, the network upper bound demand function under graph routing can be described as

$$dbf_{HI}(l) = \frac{2}{m} \sum_{i=1}^{n_{HI}} \left[ \left( \left\lfloor \frac{l - d_i}{t_i} + 1 \right\rfloor \right) C_i \right]_0 + \sum_{1 \leq p, q \leq n_{HI}} (\Delta(p'q) \max\{\lceil \frac{l}{t_p} \rceil, \lceil \frac{l}{t_q} \rceil\}). \quad (3.14)$$

We can then obtain  $dbf_{LO2HI}(l)$  as

$$dbf_{LO2HI}(l) = \frac{2}{m} \sum_{p=1}^{n_{HI}} \left[ \left( \left\lfloor \frac{l - d_p}{t_p} + 1 \right\rfloor \right) C_p + \frac{1}{2} c_p \right]_0 + \sum_{1 \leq p, q \leq 2n_{HI}} (\Delta(p'q) \max\{\lceil \frac{l}{t_p} \rceil, \lceil \frac{l}{t_q} \rceil\}). \quad (3.15)$$

Because transmission on a backup path occurs only when the two previous attempts fail, when the transmission success rate on the primary path satisfies the network packet reception ratio, the sender has no need to send a backup packet. Hence, the network upper bound demand function in this case can be rewritten as

$$\begin{aligned}
 dbf_{LO2HI}(l) &= \frac{3}{m} \sum_{p=1}^{n_{HI}} \left[ \left( \left\lfloor \frac{l-d_p}{t_p} + 1 \right\rfloor \right) c_p \right]_0 \\
 &+ \sum_{1 \leq p, q \leq 2n_{HI}} (\Delta(pq) \max\{\lceil \frac{l}{t_p} \rceil, \lceil \frac{l}{t_q} \rceil\}). \quad (3.16)
 \end{aligned}$$

### 3.4.3 Tightening the Demand Bound Functions

A loose demand bound function will lead to a pessimistic estimation of network schedulability. In this subsection, we tighten our demand bound functions by discussing the relationship between two flows and transmission conflict.

In our previous analysis Lemma 3.2, the number of conflict jobs is a conservative estimation as  $\max\{\lceil \frac{l}{t_i} \rceil, \lceil \frac{l}{t_j} \rceil\}$ . However, this value can be reduced by classifying discussions. We divide this value into the following categories:

- $t_i \leq t_j$ , and  $d_i \leq d_j$ .
- $t_i \leq t_j$ , and  $d_i \geq d_j$ .

When the path of  $F_i$  and  $F_j$  have overlaps, they may generate transmission conflicts. The delay caused by conflict cannot occur in each slot because the flow does not transmit between  $d$  and  $t$ . Obviously, when one flow works in its ideal time (between  $d$  and  $t$ ), there is no transmission conflict between  $F_i$  and  $F_j$ .

Condition 1 is shown in Fig. 3.4a; conflict occurs only when both  $F_i$  and  $F_j$  have job transmissions on the path. For a given  $l$ , the number of conflicting jobs can be expressed as

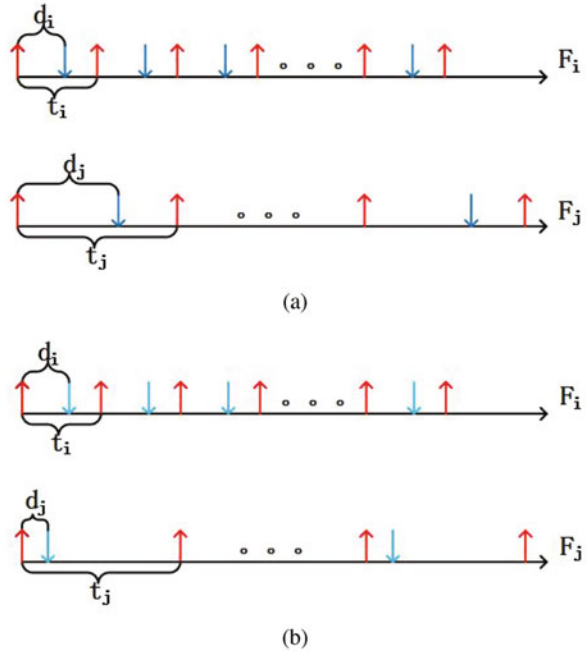
$$\lceil \frac{l}{t_j} \rceil (\lfloor \frac{d_j}{t_i} \rfloor + 1). \quad (3.17)$$

Similarly, we can obtain the number of conflicting jobs in condition 2 as

$$\lceil \frac{l}{t_j} \rceil (\lfloor \frac{d_j}{t_i} \rfloor + 1) = 2 \lceil \frac{l}{t_j} \rceil. \quad (3.18)$$

We denote the number of conflicts as  $Num(ij)$ . When we know each flow's routing information, the estimation of  $Num(ij)$  can be further precise. By taking the modulus of  $\frac{d_j}{t_i}$ , we can estimate the maximum length of  $F_i$ 's residual path as

**Fig. 3.4** Classified discussion. (a) Condition 1. (b) Condition 2



$\| \frac{d_j}{t_i} \|$ . The delay on this residual path is denoted as  $\psi$ , and we can obtain  $\psi$  as follows:

- If  $F_i$  has an overlap with  $F_j$  on this residual path,  $\psi = \Delta(\| \frac{d_j}{t_i} \|)$ , where  $\Delta(\| \frac{d_j}{t_i} \|)$  is the delay on the residual path whose length is  $\| \frac{d_j}{t_i} \|$ .
- If  $F_i$  has no overlap with  $F_j$  on this residual path,  $\psi = 0$ .

The number of conflicts can be expressed as

$$Num(ij) = \lceil \frac{l}{t_j} \rceil (\lfloor \frac{d_j}{t_i} \rfloor + \psi). \tag{3.19}$$

We can then obtain the network demand bound functions as follows:

**Theorem 3.2** *In any time interval of length  $l$ , the demand bound function in each mode can be expressed as*

$$dbf_{LO}(l) = \frac{1}{m} \sum_{i=1}^n \left[ \left( \left\lfloor \frac{l - d_i}{t_i} + 1 \right\rfloor \right) c_i \right]_0 + \sum_{1 \leq i, j \leq n} (\Delta(ij) Num(ij)). \tag{3.20}$$

$$dbf_{LO2HI}(l) = \frac{2}{m} \sum_{p=1}^{nHI} \left( \left( \left\lfloor \frac{l - d_p}{t_p} + 1 \right\rfloor \right) C_p + \frac{1}{2} c_p \right) \Big]_0$$

$$+ \sum_{1 \leq p, q \leq 2n_{HI}} (\Delta(p'q)Num(pq)). \quad (3.21)$$

$$dbf_{HI}(l) = \frac{2}{m} \sum_{p=1}^{n_{HI}} \left( \left\lfloor \left( \left\lfloor \frac{l-d_p}{t_p} + 1 \right\rfloor \right) C_p \right\rfloor \right)_0 + \sum_{1 \leq p, q \leq 2n_{HI}} (\Delta(p'q)Num(pq)). \quad (3.22)$$

The network demand bound function is  $dbf(l) = \max\{dbf_{LO}(l), dbf_{LO2HI}(l), dbf_{HI}(l)\}$ , and the network can be scheduled when  $dbf(l)$  is no less than  $\min\{dbf_{LO}(l), dbf_{LO2HI}(l)\}$ .

**Proof** The proofs of demand bound functions are similar to in Theorem 3.1. The difference is that we reduce the number of conflicts by classifying the discussion, and the demand bound functions are tightened. Because there are *carry-over jobs* at the switching time,  $dbf_{LO2HI}(l)$  must be greater than  $dbf_{HI}(l)$ . When the network supply in a time interval of length  $l$   $sbf(l)$  is larger than  $dbf_{LO}(l)$ , the network can be scheduled in low-criticality mode; when  $dbf_{LO2HI}(l) \leq sbf(l) < dbf_{LO}(l)$ , the network can be scheduled in high-criticality mode; when  $sbf(l) > \max\{dbf_{LO}(l), dbf_{LO2HI}(l)\}$ , the network cannot be scheduled. Hence, the network can be scheduled when  $dbf(l)$  is no less than  $\min\{dbf_{LO}(l), dbf_{LO2HI}(l)\}$ .  $\square$

### 3.5 Performance Evaluations

In this section, we conduct experiments to evaluate the performance of our proposed methods. Our method is first compared with the simulation result. We then compare our method with the supply/demand bound function analysis without tightening.

To illustrate the applicability of our method, for each parameter configuration, several test cases are generated randomly. For each test case, the network gateway is placed at the center of playground area  $A$ , and the other nodes are deployed randomly around the gateway. According to the suggestion in [14], given the transmitting range  $d = 40$  m, the number of nodes  $n$  and the playground area  $A$  should satisfy

$$\frac{n}{A} = \frac{2\pi}{d^2\sqrt{27}}. \quad (3.23)$$

If two nodes can communicate with each other, which means that the distance between two nodes is less than  $d$ , they are adjacent nodes. By repeatedly connecting the nearest node from the source node to the gateway, the network topology can be obtained. If some source nodes cannot connect to the gateway, their locations are generated randomly again.



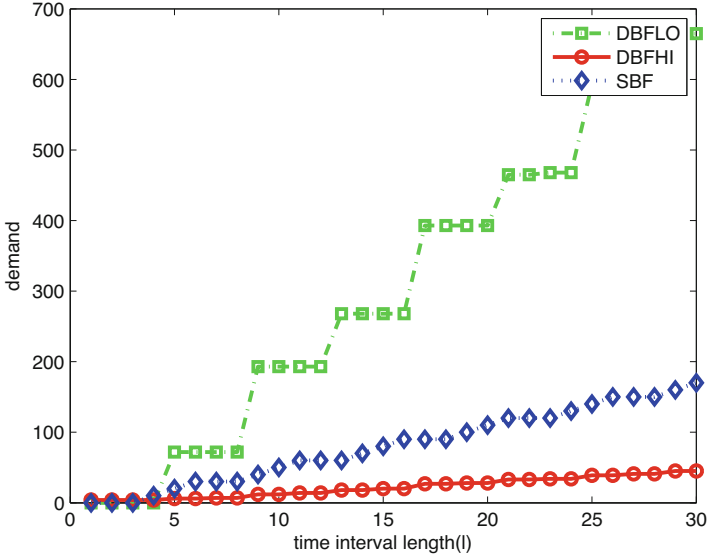
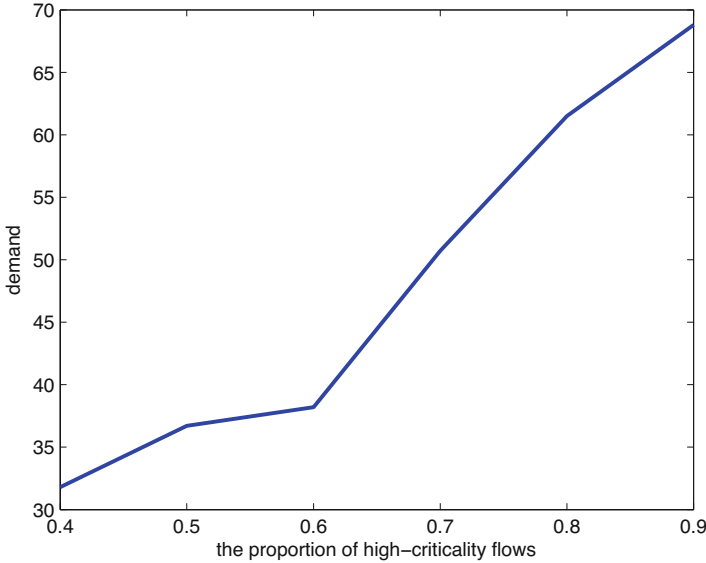


Fig. 3.5 Relationship between demand bound functions and supply bound function

Our simulations use the utilization  $u$  to control the workload of the entire network. To make flow sets available, we specify the network utilization  $U = \sum u_i (U < 1)$ , and the UUniFast algorithm [15] is used to generate each flow's utilization  $u_i$  ( $u_i = \frac{c_i}{t_i}$ ). The result generated by the UUniFast algorithm follows a uniform distribution and is neither pessimistic nor optimistic for the analysis [15].

Figure 3.5 is an example of the relationship between the demand bound function in different criticality modes and the supply bound function. In this example, according to the actual situation, we set the number of nodes as  $n = 70$  and the number of flows as  $F = 20$ . At the beginning, with the network running in low-criticality mode, the demand is zero. At time slot 5,  $DBFLO$  is 72, which is larger than the upper bound of network supply; the network then switches to high-criticality mode. Considering *carry-over jobs*, we can calculate the demand in high-criticality mode from time slot 5. Because the network demand is less than the supply, this example is a stable network. Furthermore, Fig. 3.5 reveals that the demand bound functions are stepwise increasing. This is because  $dbf(l)$  is the network demand over a period of time. When a job has enough time slots to transmit (e.g., a job is just released), its demand is zero and does not require immediate execution. With the decrease of the remaining time, the job becomes urgent. When the remaining time for the job is  $c$ , the job must be forwarded immediately; otherwise, it will miss the deadline. The job demand is then changed to the number of hops  $c$ .

Figure 3.6 is the variation tendency of  $DBFHI$  with the proportion of high-criticality flows. Because changing the proportion of  $n_{HI}$  does not affect network demand in low-criticality mode, Fig. 3.6 shows the network demand only in high-criticality mode. Obviously, the network demand is increasing with the increasing

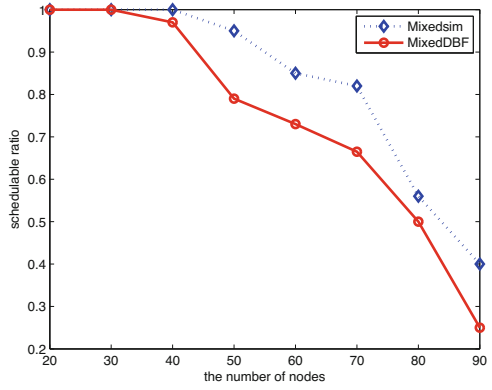


**Fig. 3.6** Variation tendency of *DBFHI* with the percentage of high-criticality flows

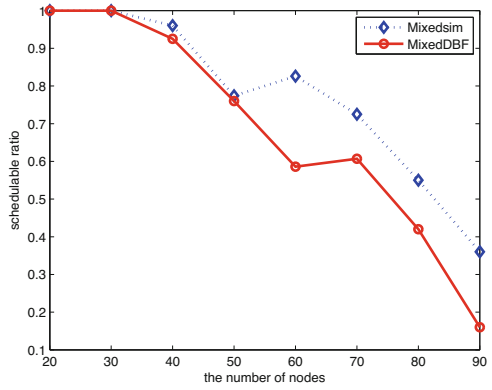
proportion of high-criticality flows. At the beginning (0.4–0.6), the network demand increases slowly. From 0.7–0.9, the demand of the network increases rapidly. This is because more flows in high-criticality mode generate more transmission conflicts in conditions 1, 2, and 4. The network needs more resources to ensure that the job meets its deadline. This phenomenon is enhanced severely with increasing  $P$ .

To analyze the correctness of our method, we compare the network schedulability ratio between the simulation result (denoted as MixedSim) and our method (denoted as MixedEDF) in Fig. 3.7. For each point in the figures, more than 100 test cases are randomly generated. From the figures, we can know that our algorithm can accurately evaluate the network schedulability ratio regardless of which parameters are used. Because we pessimistically estimate transmission conflicts to guarantee our method's reliability, the evaluation value of the network demand bound is larger than the actual demand. In Fig. 3.7a and b, the proportions of high-criticality flows are  $P = 0.4$  and  $P = 0.5$ , respectively. With the increasing of nodes, the network schedulability ratio declines in both situations. However, the schedulability ratio in Fig. 3.7b falls faster than in Fig. 3.7a. This is because the network generates more transmission conflicts when increasing the number of high-criticality flows. Note that compared with Fig. 3.7a, Fig. 3.7c has 0.1 additional utilization, so the spacing between the simulation curve and analysis curve is expanded. Although there are fluctuations between 30 to 60, our method can always bound the schedulable ratio (the fluctuations are caused by the randomly generated network environment). Because the two figures generate test cases according to the respective utilization, their test cases are different. When network utilization increases, the number of

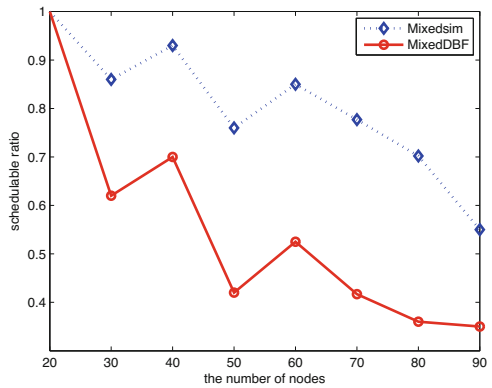
**Fig. 3.7** Relationship between schedulability ratio and the number of nodes. (a)  $U = 0.5, P = 0.4$ . (b)  $U = 0.5, P = 0.5$ . (c)  $U = 0.6, P = 0.4$



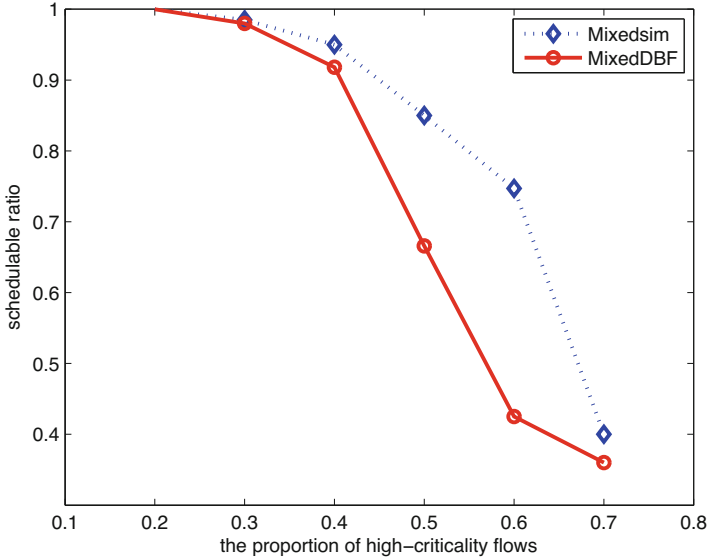
(a)



(b)



(c)



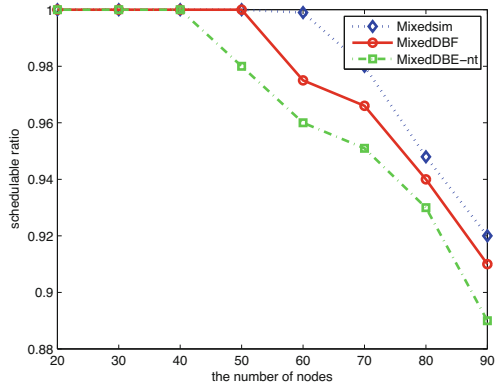
**Fig. 3.8** Relationship between schedulability ratio and the proportion of high-criticality flows

hops from source to destination increases. This increases the number of potential conflicts. The estimation result then becomes more pessimistic.

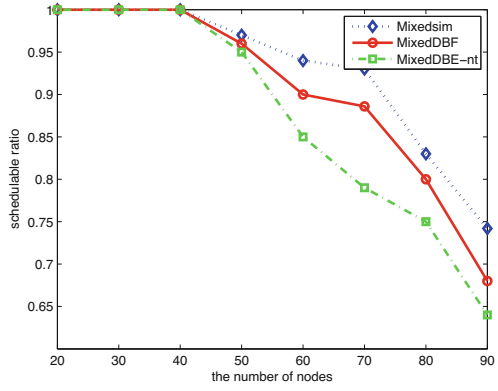
Figure 3.8 is the relationship between the schedulability ratio and the proportion of high-criticality flows. It is easy to understand that the schedulability ratio declines with the increasing proportion of high-criticality flows. However, the spacing between the two curves changes with  $P$  (small–big–small). This is because our method should consider the transmission conflicts in all situations to ensure reliability. In the beginning, there are only a few conflicts in high-criticality mode. With increasing high-criticality flows, the strict estimation considers each path overlap as a transmission conflict, which leads to larger spacing between two curves. When  $P = 0.7$ , the number of conflicts increases in MixedSim, which reduces the schedulability ratio, and then the difference becomes small.

We illustrate the advantage of MixedDBF by comparing it with the supply/demand bound function analysis without tightening (denoted as MixedDBF-nt) in Fig. 3.9. Obviously, MixedDBF is better than MixedDBF-nt regardless of the conditions. With increasing network utilization or proportion of high-criticality flows, the error of MixedDBF-nt grows faster than MixedDBF. The reason is that both increasing network utilization and the number of high-criticality flows will increase the number of path overlaps. MixedDBF tightens the delay caused by the transmission conflict by Eq. (3.19). With increasing overlaps, the effect of Eq. (3.19) will be better. Hence, the error of MixedDBF-nt grows faster than MixedDBF.

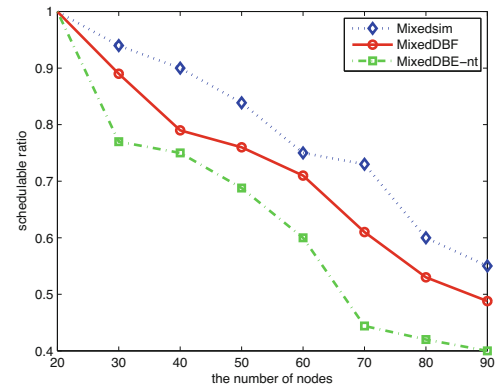
**Fig. 3.9** Schedulability comparison among MixedSim, MixedDBF, MixedDBF-nt. **(a)**  $U = 0.4, P = 0.2$ . **(b)**  $U = 0.5, P = 0.2$ . **(c)**  $U = 0.4, P = 0.6$



(a)



(b)



(c)

## 3.6 Summary

WirelessHART adopts reliable graph routing to enhance network reliability. However, graph routing introduces substantial challenges in analyzing the schedulability of real-time flows. Too much transmission load will increase conflicts and reduce network performance. Disaster may happen when critical tasks miss their deadlines in this situation. Hence, firstly, we propose a novel network model that can switch routing based on the criticality mode of networks. When errors or accidents occur, the network switches to high-criticality mode and low-level critical tasks are abandoned. Secondly, we analyze the demand bound of mixed-criticality industrial wireless sensor networks under the EDF policy and formulate network demand bounds in each criticality mode. Thirdly, we tighten the demand bound by analyzing carry-over jobs and classifying the number of conflicts to improve analysis accuracy. The simulations based on random network topologies demonstrate that our method can estimate network schedulability efficiently.

## References

1. Chen DJ, Nixon M, Mok A (2010) Why WirelessHART. Springer, Berlin
2. Saifullah A, Gunatilaka D, Tiwari P, Sha M, Lu CY, Li B, Wu CJ, Chen YX (2015) Schedulability analysis under graph routing in WirelessHART networks. In: Real-time systems symposium (RTSS). IEEE, Piscataway, pp 165–174
3. Wu C, Gunatilaka D, Saifullah A, Sha M, Tiwari PB, Lu CY, Chen YX (2016) Maximizing network lifetime of WirelessHART networks under graph routing. In: IEEE first international conference on internet-of-things design and implementation (IoTDI), pp 176–186
4. Giuseppe L (2005) Earliest deadline first. <http://retis.sssup.it/~lipari/courses/rtos/lucidi/edf.pdf>
5. Saifullah A, Xu Y, Lu CY, Chen YX (2010) Real-time scheduling for WirelessHART networks. In: Real-time systems symposium (RTSS). IEEE, Piscataway, pp 150–159
6. Spuri M, Reflex P (1996) Analysis of deadline scheduled real-time systems. HAL-INRIA.
7. Baruah SK, Mok AK, Rosier LE (1990) Preemptively scheduling hard-real-time sporadic tasks on one processor. In: Real-time systems symposium (RTSS). IEEE, Piscataway, pp 182–190
8. Rox J, Ernst R (2013) Compositional performance analysis with improved analysis techniques for obtaining viable end-to-end latencies in distributed embedded systems. *Int J Softw Tools Technol Transf* 15(3):171–187
9. Mok AK, Feng X, Chen DJ (2001) Resource partition for real-time systems. In: Real-time technology and applications symposium (RTAS). IEEE, Piscataway, pp 75–84
10. Shin I, Lee I (2003) Periodic resource model for compositional real-time guarantees. In: Real-time systems symposium (RTSS). IEEE, Piscataway, pp 2–13
11. Ekberg P, Wang Y (2012) Bounding and shaping the demand of mixed-criticality sporadic tasks. In: The euromicro conference on real-time systems (ECRTS). IEEE, Piscataway, pp 135–144
12. Tilleenius M, Larsson E, Badia RM (2015) Resource-aware task scheduling. *Trans Embed Comput Syst* 14(5):1–25
13. 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

14. Camilo T, Silva JS, Rodrigues A, Boavida F (2007) Gensen: a topology generator for real wireless sensor networks deployment. In: The international conference on software technologies for embedded and ubiquitous systems. Springer, Berlin, pp 436–445
15. Bini E, Buttazzo CC (2005) Measuring the performance of schedulability tests. *Real-Time Syst* 30(1):129–154

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

