Worst-case response time of aperiodic message in WorldFIP

Zhi Wang, Ye-Qiong Song, Hai-Bin Yu, Youxian Sun

To cite this version:

Zhi Wang, Ye-Qiong Song, Hai-Bin Yu, Youxian Sun. Worst-case response time of aperiodic message in WorldFIP. 15th Triennial World Congress of the International Federation of Automatic Control Barcelona 2002 - B'02, Jul 2002, Barcelona/Spain, 6 p. inria-00107601

HAL Id: inria-00107601
https://inria.hal.science/inria-00107601

Submitted on 19 Oct 2006

WORST-CASE RESPONSE TIME OF APERIODIC MESSAGE IN WORLDFIP[1]

Zhi WANG[1*] Ye_qiong SONG[2] Hai-bin YU[3] Youxian SUN[1]

1   National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China

2   LORIA–ENSEM, 2, av. de la Forêt de Haye, 54516 Vandoeuvre-lès-Nancy, France

3   Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China

Author of correspondence: zhiwang_iipc@yahoo.com

Abstract: This paper presents an integrated approach to analyse worst-case responding time (WCRT) of aperiodic message in WorldFIP for guaranteeing realtime requirement of distributed control system (DCS). Aperiodic message transmission within WorldFIP is a very complicated procedure because it is concerned with left time in a microcycle and must successively follow three different stages. To decrease pessimistic factors in WCRT of aperiodic message, we integrate the procedures of transmitting periodic message, identifying aperiodic request and transmitting aperiodic message together, and deduce a tightly bound for WCRT of aperiodic message. The result is validated by case study at last. *Copyright©2001 IFAC*

Keywords: WorldFIP, Field Communication, Real-time, Aperiodic message, Worst-Case Responding Time

1.   INTRODUCTION

DCS, which mainly support realtime applications, such as automated manufacturing and industrial process control, et al., is characterized by the correctness of its tasks depending on both their logical results and the time at which these results appear. Within a DCS, tasks usually reside on distributed nodes and communicate with one another through message transfer to accomplish a common goal. Therefore, it is difficult to ensure timely results of tasks in a DCS without a network that supports the timely inter-task messages. So the ability to providing timely message delivery and predictable inter-task communication is of great importance to the underlying network and protocol because failure to meet the message transmission deadline may lead to a disaster (H.Koptez 1997; C.Bottazzo 1997; Tindell 1997).

For local network, IEEE802.4, IEEE802.5 and FDDI adopt the timed token media access control protocol for providing bounded media access time. The upper bound of average token rotation time and the maximum time elapsed between two token consecutively visiting a same node see (Sevcik et al., 1987). As to allocate synchronous bandwidth for FDDI see (Argrawl et a. 1994; Burns et al., 1999). However, due to the random media access nature of CSMA/CD (IEEE802.3) and CSMA/CA (CAN) protocols, it is difficult to make deadline guarantees for these kinds of network. Hence, methods of virtual-time protocol, window protocol, traffic adapting protocol and distributed priority queue are proposed to enhance their realtime capability (Zhao et al., 1990; Zuberi et al. 2000).

In this paper, we consider centralized-scheduling local network. Normally, this kind of network utilizes producer-distributor- consumer (PDC) model, in which a centralized bus arbitrator, i.e., distributor, relates producer and consumer according to a given schedule (Thomese, 1997). Examples of networks these use such model are WorldFIP and FF. As for PDC models, realtime requirement of periodic traffics can be easily guaranteed by reserving the required communication capacity in the distributor's schedule table. However, for aperiodic messages more complex problems exist, and different approaches to meet their temporal requirement are adapted by WorldFIP and FF. As for FF, considering PT counting, PT rotation period, and aperiodic message priority, WCRT of aperiodic messages and new PT counting mechanism are given in (Wang et al. 2001, Wang et al., 2001). As for WorldFIP, function relationship between WCRT of aperiodic messages and the schedule of periodic messages is given in (Tover et al 2000). However, Tover's result is over pessimistic because he considers identifying aperiodic request and transmitting aperiodic message as same service time, which equals to the larger between them. Further, the relation between indicating and identifying aperiodic request is not considered. This paper will focus on approaching WCRT of aperiodic message within WorldFIP, considering the above neglected factors in (Tover et al 2000).

2 PROCEDURES OF PERIODIC AND APERIODIC MESSAGE TRANSFERS

WorldFIP provides two types of basic functions: bus arbitration and production/consumption. At any time, only one station can perform the function of active bus arbitration. Other nodes just receive messages or send messages according to the order issued by the station. Hence, the transmission mechanism of WorldFIP is centralised since the access right of each producer to the WorldFIP is controlled by centralised active bus arbitrator, known a bus arbitrator (BA).

WorldFIP provides two types of data: variable and message. Both periodic and aperiodic traffics are allowed in both cases. For brief, variable and message are referred as message in following section.

2.1 PDC Model in WorldFIP

In WorldFIP, the transmissions of both periodic and periodic traffics are based on PDC model, which relates producers and consumers within the distributed system. In this model, for each message (both periodic and aperiodic) there is one producer, and one or more consumers. Producer sends each produced message to one or several consumers. Once a message has been produced, it is automatically sent to consumers according a given schedule without the need of a direct request.

In order to manage transaction associated to a single message, a unique identifier is associated to each message. The BA broadcasts a question frame ID_DAT, which includes the identifier of a specific message. The node that has the corresponding message, responds with a responding frame RP_DAT.

The message transfer implies the transmission of a pair of frames: ID_DAT and RP_DAT. We denote this sequence as an elementary transaction, since ID_DAT and RD_DAT must occur consecutively. Similarly, ID_RQ and RP_RQ (explained in section 2.3) also construct an elementary transaction.

2.2 Bus Arbitrator Table in WorldFIP

In WorldFIP, the network schedule is stored in a bus arbitrator table (BAT), which is made up of a set of basic schedule table, known as microcycle. Periodic messages are given access right according to their orders in each microcycle, then are aperiodic messages given if time left in this microcycle is enough. The portion of a microcycle reserved for periodic message is denoted as periodic window, whereas the time left after the periodic window is denoted as aperiodic window (shown in Fig.1). Schedules for each types of traffics are stored in their specific windows: periodic window and aperiodic window respectively. Once all microcycles have been performed, the BA repeats the network schedule from the first microcycle in the BAT.
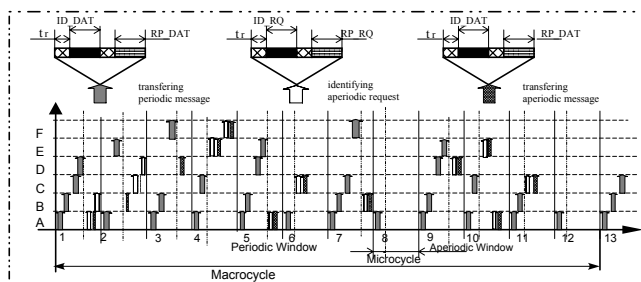

Fig.1 Periodic Window and Aperiodic Window

Two important parameters are associated with BAT, microcycle and macrocycle. The former imposes the maximum rate at which BA performs a set of scans, and the latter is minimum duration during which the sequence of microcycle is repeated. Usually, microcycle and macrocycle are set equal to the highest common factor (HCF) and the lowest common multiple (LCM) the required scan periodicities respectively.

2.3 Aperiodic Message Transfers in WorldFIP

Within a distributed system, not all tasks need exchanging messages periodically, typically such exchange will concern with event and alarms generated as a consequence of fault in the inter- connected device that have to be notified quickly to the fault management system. In order to increase network utilisation, it is preferable to map these tasks into aperiodic messages. To implement that, BA gives access rights gives to an aperiodic message only when realising the aperiodic message exists.

The details of BA scheduling aperiodic message is shown in Fig.2, where it is obvious that completing transfer of an aperiodic message needs experiencing three stages.

1. Indicating pending aperiodic message request (InPMAP): A node with a sending aperiodic request must wait for its next periodic message transfer (say periodic message X) to notify BA, via setting an aperiodic request bit in the RP_DAT frame. BA stores the indication of the yet not identified aperiodic request in pending request queue, by which BA is aware of a pending request in the node that produces periodic message X.

2. Identifying pending aperiodic message request (IdPMAP): In an aperiodic window, BA asks producer of periodic message X to transmit the list of identifiers of pending aperiodic requests in a node the producer belongs, by sending ID_RQ frame. The producer responds with RP_RQ frame, which concludes the list. Then BAT stores the list into another BA's queue, ongoing aperiodic queue.

3. Transmitting ongoing aperiodic message (TOAM): Finally, BA processes aperiodic messages that are always stored in ongoing aperiodic queue, in an aperiodic window. For each transfer of aperiodic message, BA uses the same mechanism as that used for periodic messages.
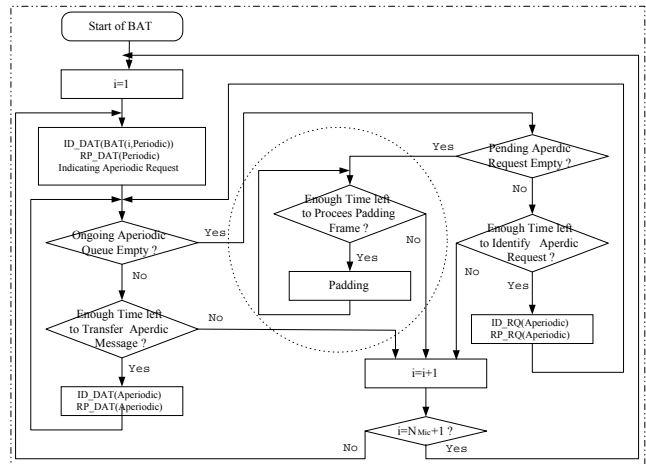
Fig.2 Schedule of BAT for Periodic and Aperiodic Messages

The complete procedure for an aperiodic message from its arrival to completing its transmission is shown in Fig.3.

Fig.3 Responding Procedure of an Aperiodic Message

It is important to note that the delay of BA processing ongoing aperiodic queue or ongoing aperiodic queue depends on length of the two queues and length of aperiodic window. The latter is function of periodic traffic pattern and corresponding schedule algorithm. ID_RQ/ RP_RQ and

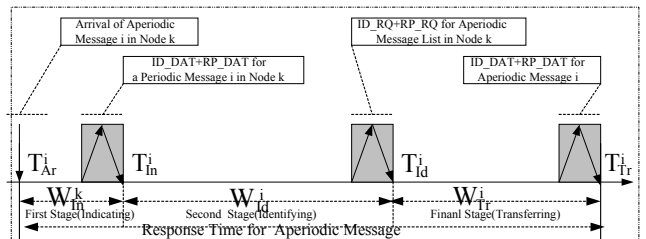ID_DAT/ RP_DAT can be carried out in an aperiodic window if there are enough time left to completely process them, otherwise they must be carried out in two aperiodic windows.

Another is worthwhile to note is that pending request queue is processed if and only if ongoing aperiodic queue is empty. As is depicted in Fig.2

3. MODEL OF APERIODIC MESSAGE COMMUNICATION

3.1 Model of Periodic and Aperiodic Messages

Consider a DCS where there are n nodes interconnected by a WorldFIP network. Assume there are n_a periodic message M_p^i $(i=1,\cdots,n_p)$ and n_p aperiodic messages M_a^i $(i=1,\cdots,n_a)$ within the DCS.

Note that WorldFIP utilises centralized schedule scheme to periodic messages and aperiodic messages according to their temporal characteristics, so n_p and n_a are irrespective of n.

$$M_p^i = (C_p^i, T_p^i, D_p^i) \tag{1}$$
$$M_a^i = (C_a^i, T_a^i, D_a^i) \tag{2}$$

where, T_p^i, D_p^i and C_p^i correspond to periodicity, deadline and transaction length of the periodic message M_p^i respectively, C_a^i, T_a^i and D_a^i correspond to periodicity, deadline and transaction length of the aperiodic message M_a^i respectively.

Normally, the arrival duration of aperiodic message is irregular, however we assume there are minimum inter-arrival time for aperiodic message in order to guarantee its timing constraints. That means we evaluate the realtime capacity of WorldFIP at the peak load of aperiodic message, and in this case T_a^i corresponds to the minimum inter-arrival time between two consecutive requests of M_a^i.

For a periodic message M_p^i, completing its transmission only needs an elementary transaction ID_DAT/RP_DAT, and the time duration for this transaction is

$$C_p^i = bps^{-1}.\big(len(ID\_DAT_p^i) + len(RP\_DAT_p^i)\big) + 2 \times t_r \tag{3}$$

where bps denotes network data rate, and $ID\_DAT_p^i$ $RP\_DAT_p^i$ and denotes number of bit in frame of ID_DAT and RP_DAT for M_p^i, t_r is turnaround time, the time elapsed between any two consecutive frames.

From the previous section we know that completing an aperiodic message transfer needs two elementary transactions, IdPMAP and TOMA. Therefore, C_a^i actually consists of two components, C_{Id}^i and C_{Tr}^i, which denote time duration of IdPMAP and TOMA for M_a^i respectively. We rewrite expression (4) as following:

$$M_a^i = (C_{Id}^i, C_{Tr}^i, T_a^i, D_a^i) \tag{4}$$

$$C_{Id}^i = bps^{-1}.\big(len(ID\_RQ_a^i) + len(RP\_RQ_a^i)\big) + 2 \times t_r \tag{5}$$

$$C_{Tr}^i = bps^{-1}.\big(len(ID\_DAT_a^i) + len(RP\_DAT_a^i)\big) + 2 \times t_r \tag{6}$$

3.2 Building BAT for Periodic Messages

For periodic messages, a common approach is to utilise well known scheduling algorithms such as Rate Monotonic (RM) and Earliest Deadline First (EDF), which are optimal with respect to fixed and priority schemes. By the way, applying Deferred Release (DR) algorithm can decrease communication jitter caused by RM or EDF.

Assume an example that a WorldFIP interconnects 6 nodes; each owns a periodic message, shown in Table.1, and an aperiodic message.

Table.1 Example set of periodic message

Variable Identifier	A	B	C	D	E	F
Periodicity (ms)	1	2	2	3	4	6
C_p^i ($100\,\mu s$)	2	2	2	2.5	2.5	3

Let N_{Mic} denote number of microcycles within one macrocycle. Then, N_{Mic} equals to LCM/HCF if the BAT is constructed according to rule of HCF and LCM.

The value of the microcycle and macrocycle of BAT for periodic message in Table.1 are respectively set to 1ms and 12 ms, and number of microcycle is 12. A feasible schedule in BAT for all periodic messages is illustrated in Table.2.

Table.2 A feasible schedule for periodic messages in Table.1

Periodic Messages	Microcycle											
	1	2	3	4	5	6	7	8	9	10	11	12
A	1	1	1	1	1	1	1	1	1	1	1	1
B	1	0	1	0	1	0	1	0	1	0	1	0
C	1	0	0	1	1	0	1	0	1	0	1	0
D	1	0	0	1	0	0	0	1	0	1	0	0
E	0	1	0	0	1	0	0	0	1	0	0	0
F	0	0	1	0	0	0	0	1	0	0	0	0

Note that the schedule in Table.2 is not confined to RM, EDF or DR.

3.3 Response Time of Aperiodic Message

The duration starting from T_{Ar}^i, arrival of M_a^i into a node, ending with T_{Tr}^i, completion of transferring refers to response time of M_a^i. Let R_A^i denote response time of M_a^i. It is obvious that R_A^i includes the following three components according to the scheduling mechanism of WorldFIP for aperiodic message.

(1). Waiting time of indicating aperiodic request (WTInAR) W_{In}^k, during which the arrival of an aperiodic message to node k at the instants of T_{Ar}^i has not been realised by BA until there are periodic messages in node k be transferred at the instants of T_{Tp}^k. It is obvious that BA will be aware that there is a pending aperiodic request in node k after the time of W_{In}^k is elapsed.

(2). Waiting time of identifying aperiodic request (WTIdAR) W_{Id}^i, during which the pending aperiodic request for M_a^i request stays in the BA's pending queue and its identifier is unclear till the related pairs of frame ID_RQ and RP_RQ is processed in an aperiodic window. At the end of the interval W_{Id}^i, BA will be aware the identifier of the pending aperiodic request in node k.

(3). Waiting time of transmitting ongoing aperiodic message (WTTOAP) W_{Tr}^i, during which the identifiers of aperiodic message request stays in the BA's ongoing request queue till the related pairs of frame ID_DAT and RP_DAT is processed in an aperiodic window. Hence,

$$R_A^i = W_{In}^k + W_{Id}^i + W_{Tr}^i \quad \left(M_a^i \in \text{node } k\right) \tag{7}$$

R_A^i must be upper bound to guarantee a bounded response time for aperiodic message, but exactly calculating it is difficult. The first component of R_A^i is related to periodicity of periodic messages in node k. The last two components are related to periodic message pattern, since the capacity of aperiodic window for processing the two components depends on the schedule in the BAT, which is determined according to periodic message pattern.

3.4 Upper Bound of Indicating Request Delay

Requesting node processes a local aperiodic request through setting *request bit* in a RP_DAT frame, so indicating pending aperiodic request is done if and only if the node produces RP_DAT frame. Therefore, WTInAR exists. WTInAR in node is related to scanning periods of periodic messages produced in the node, and it does not exceed the minimum periodicity of these periodic messages. The reason is obvious that an aperiodic request at most wait the periodicity of the periodic message with the minimum periodicity even the aperiodic request can't be indicated by other periodic messages.

For an aperiodic message in node k, it will suffer worst-case WTInAR if it arrives node k while a periodic message just leaves node k, since BA can't realise the aperiodic request for the aperiodic message until the next transmission of the periodic message is done. Further, periodic messages are not polled regularly due to jitter inherent to communication schedule in BAT, so it is difficult to get exact WTInAR even it is possible.
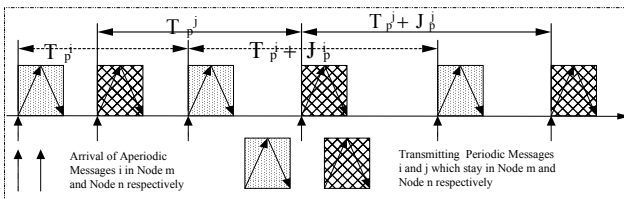

Fig 4. Procedure of Indicating Request of an aperiodic message.

Therefore, we have to investigate upper bound of WTInAR instead of exact value of WTInAR through investigating all periodic messages in a node.

$$W_{In}^k = \min_{M_a \in \text{node } k} \left\{ \max \left\{ J_p^i \right\} + T_p^i + C_p^i \right\} \tag{8}$$

J_p^i denotes communication jitters of that periodic message M_p^i, which are obvious as shown in Fig.4.

3.5 Busy Period of Aperiodic Messages

IdPMAP and TOAM are function of periodic message traffic pattern, since aperiodic window length is function of periodic message schedule in BAT. Thus, the exact characterization of periodic message traffic pattern is fundamental for IdPMAP and TOAM.

Given a message set (irrespective of periodic or aperiodic), we have different patterns of message arrival. It is not different to show that the worst-case pattern is that instances of all messages are released as soon as possible. That is these first instances are released simultaneously, and the others are released according to their respective maximum rate. That means the upper bound for WTIdAR and WTTOAP occur when both the periodic and aperiodic messages are simultaneously at their maximum load. The result is quite intuitive, and has been proved by an well-known theorem worked by (Liu and Layland, 1973).

Theorem 3.1 (Liu and Layland) When a schedule algorithm is used to schedule a set of tasks on a processor, if there is a overflow for a certain arrival pattern, then there is an overflow without idle time prior to it in the pattern in which all task instances are released as soon as possible.

The theorem suggests studying the worst-case response time in busy period, the duration from the instant of first instance is released to the first processor idle time. It also turns out that busy period length does not depend on schedule algorithm, as far as it is non-idling, but only on the task arrival pattern.

Although theorem 3.1 only exists when non-idling in processor, the concept of busy period can be utilized to schedule of BA, in which idling is allowed.

Definition 3.1 Maximum aperiodic message workload (MWAM), $W(t)$ is defined as the maximum amount of processing time requested by all instances of aperiodic message whose release time are in the interval of [0,t].

Definition 3.2 Aperiodic message busy period (AMBP) is a time interval that begins from instant of first instance being released to no ongoing aperiodic message waits to be processed under all instances being released as soon as possible.

The aperiodic traffic is maximized if each instance of aperiodic message needs a specific IdPMAP. As a consequence, the aperiodic window will perform alternately sequences of ID_RQ/RP_RQ and ID_DAT/RP_DAT, and each instance of aperiodic message need two transactions concerning with IdPMAP and TOAM. Thus, the exact MWAM is

$$W(t) = \sum_{i=1}^{n_a} \left(1 + \left\lfloor \frac{t}{T_a^i} \right\rfloor \right) \cdot \left(T_a^i + T_a^i\right) \tag{9}$$

Let W_A^i be i^{th} aperiodic window length, then,

$$W_A^i = MicP - \sum_{j=1}^{n_p} (table[j,i^*]) \times C_p^j \qquad (10)$$

within expression (10), $table[j, i^*]$ denotes whether periodic message j is scheduled in i^{th} microcycle. i^* denotes the order of ith microcycle in BAT, $i^*=[(i-1) \bmod N_{Mic}]$.

Let AMBP be described as interval $[0, Len_{AB}]$. According to definition 3.1, transfer capability of aperiodic window within AMBP can't be less than its MWAM during the same period. Hence,

$$Len_{AB} = \min \left\{ \psi \left| \sum_{i=1}^{\left\lceil \frac{\psi}{MicP} \right\rceil} W_A^i \geq \sum_{i=1}^{n_a} \left(1 + \left\lfloor \frac{\psi}{T_a^i} \right\rfloor \right) \left(T_a^i + T_a^i \right) \right\} \right. \qquad (11)$$

Notes that the transfer capability available of aperiodic window is wasted when that does not suffice the requirement of transactions of IdPMAP or TOAM. IdPMAP or TOAM must be completed without interrupt within an aperiodic window. We refer that property as atomic property of aperiodic transaction. However, the atomic property of aperiodic transaction is not considered in equation (11), which actually is only a necessary condition that AMBP must meet. Hence, we must apply feasible transfer capability considering atomic property.

Further, AMBP may not start at the beginning of the macrocycle because of the actual schedule of BAT for periodic messages, we must calculate beginning from any aperiodic window.

Let Num_{AB} be the number of aperiodic windows necessary for aperiodic message within $[0, \bar{Len}_{AB}]$, where $\bar{Len}_{AB}$ denote AMBP considering atomic property of aperiodic transactions of IdPMAP and TOAM; Num_{AB}^i be the number of aperiodic windows necessary for aperiodic message, which starts from i^{th} aperiodic window; N_A^i be the number of identifying aperiodic request or transmitting ongoing aperiodic message in i^{th} aperiodic window.

$$Num_{AB}^i = \min \left\{ \psi \left| \sum_{j=i}^{i+\psi-1} N_A^j \geq \sum_{i=1}^{n_a} \left\lceil \frac{\psi \cdot MicP}{T_a^i} \right\rceil \cdot 2 \right\} \right. \qquad (12)$$

$$Num_{AB} = \max \left(Num_{AB}^i \right) \qquad (13)$$

For calculating exact Num_{AB}^i, we must consecutively calculate from N_A^i to the last the aperiodic window. The reason is that the difference transactions times for IdPMAP and TOAM make N_A^i is related to type of last transaction within $(j-1)^{th}$ aperiodic window.

$$N_A^i = \left\lfloor \frac{W_A^i}{Sa+Rq} \right\rfloor \cdot 2 + \left\lfloor \frac{W_A^i - \left\lfloor \frac{W_A^i}{Sa+Rq} \right\rfloor \cdot 2 \cdot (Sa+Rq)}{Rq} \right\rfloor \qquad (14)$$

N_A^i being add indicates the last transaction in i^{th} aperiodic window performs is identifying aperiodic request,

otherwise is transmitting ongoing aperiodic message. According to the type of last transaction within previous aperiodic window,

$$N_A^{i+1} = \tilde{N}_A^{i+1} + \left\lfloor \frac{W_A^{i+1} - \tilde{N}_A^{i+1} \cdot \left(\max\left(C_{In}^i\right) + \max\left(C_{Tr}^i\right)\right)}{\max\left(C_{In}^i\right) \cdot \left(\left(N_A^i+1\right)\%2\right) + \max\left(C_{Tr}^i\right) \cdot \left(\left(N_A^i+1\right)\%2\right)} \right\rfloor \qquad (15)$$

where $\tilde{N}_A^{i+1} = \left\lfloor \frac{W_A^{i+1}}{\max\left(C_{In}^i\right) + \max\left(C_{Tr}^i\right)} \right\rfloor \cdot 2$

Similarly, we can deduce number of transactions for aperiodic message in any aperiodic window.

$$N_A^j = \tilde{N}_A^j + \left\lfloor \frac{W_A^{j1} - \tilde{N}_A^j \cdot \left(\max\left(C_{In}^i\right) + \max\left(C_{Tr}^i\right)\right)}{\max\left(C_{In}^i\right) \cdot \left(\left(N_{All\_AW}^{i,j-1}+1\right)\%2\right) + \max\left(C_{Tr}^i\right) \cdot \left(\left(N_{All\_AW}^{i,j-1}+1\right)\%2\right)} \right\rfloor \qquad (16)$$

where $N_{All\_AW}^{i,j} = \sum_{k=i}^{j} N_A^k$

Based on the above results, a sufficient condition for busy period of aperiodic message can be deduced.

$$\tilde{L}en_{AB} = (Num_{AB}-1) \cdot MicP + \sum_{k=i}^{n_p} table\left(i, (Num_{AB}-1)^*\right) + \qquad (17)$$
$$\left(\sum_{k=i}^{n_a} \left\lceil \frac{\tilde{L}en_{AB}}{T_a^i} \right\rceil - \left\lfloor \frac{Num_{AB}-1}{2} \right\rfloor \right) \cdot \left(\max\left(C_{In}^i\right) + \max\left(C_{Tr}^i\right)\right)$$

AMBP must be modified if release jitter J_a^i is considered.

$$W(t) = \sum_{i=1}^{n_a} \left(1 + \left\lfloor \frac{J_a^i + t}{T_a^i} \right\rfloor \right) \cdot \left(T_a^i + T_a^i \right) \qquad (18)$$

$$Num_{AB}^i = \min \left\{ \psi \left| \sum_{j=i}^{i+\psi-1} N_A^j \geq \sum_{i=1}^{n_a} \left\lceil \frac{J_a^i + \psi \cdot MicP}{T_a^i} \right\rceil \cdot 2 \right\} \right. \qquad (19)$$

$$\bar{L}en_{AB} = (Num_{AB}-1) \cdot MicP + \sum_{k=i}^{n_p} table\left(i, (Num_{AB}-1)^*\right) + \qquad (20)$$
$$\left(\sum_{k=i}^{n_a} \left\lceil \frac{J_a^i + \tilde{L}en_{AB}}{T_a^i} \right\rceil - \left\lfloor \frac{Num_{AB}-1}{2} \right\rfloor \right) \cdot \left(\max\left(C_{In}^i\right) + \max\left(C_{Tr}^i\right)\right)$$

3.6 Worst-Case Response Time of Aperiodic Messages

Note that equation (17) and (20) may include instances of aperiodic messages, whose indicating aperiodic request may not occur within this busy period (See fig.5). Therefore these equations are only a sufficient condition that AMBP must meet, still they are upper bound of AMBP.

Hence these equations must be modified through excluding these instances and evaluating them again till they conform to definition 3.1. For example of Fig.6, busy period 1 to definition 3.1, but busy period 1 does. From equation (7) we know that response time of aperiodic messages consists of three components, and we apply AMBP as the last two components. If we just add InPMAP to AMBP as worse-case response time, that is over pessimistic because they are not completely independent. For an aperiodic message, its indicating aperiodic request may occur in busy period 2 or only in busy period 1, so we have to consider its worse-case response time from the two cases.

Let $\tilde{z}_i$ denote worst-case response time of aperiodic message i in case of 1, $\tilde{z}_i$ denote worst-case response time of aperiodic message $\tilde{i}$ in case of 2. Let $\tilde{L}en_{AB}$ denote

length of busy period 2.

In the first case of indicating aperiodic request occurring in busy period 2, IdPMAP and TOAM can't exceed busy period 2 length, hence IdPMAP plus length of busy period 2 is upper bound of aperiodic message response time.

$\widetilde{L}en_{AB}$ can be calculated as the left time of $\widetilde{L}en_{AB}$ after subtracting identifying aperiodic request time and transmitting aperiodic message time of aperiodic messages, whose identifying aperiodic requests do not occur in busy period 2, and subtracting related periodic message transfer time.

$$\overset{\approx}{R}{}_A^i = W_A^i + \widetilde{L}en_{AB} \qquad (18)$$

In the second case of no indicating aperiodic request occurring in busy period 2, IdPMAP is larger than busy period 2, hence IdPMAP plus TAMD can't exceed the length of busy period 1 subtracting the length of busy period 2. Therefore, the left time of busy period 1 plus IdPMAP is upper bound of aperiodic message response time.

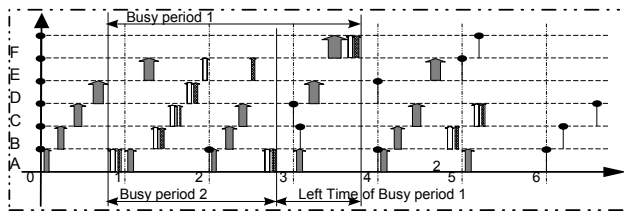$$\widetilde{R}{}_A^i = W_A^i + \vec{L}en_{AB} - \widetilde{L}en_{AB} \qquad (19)$$

Fig.5 Distribution of Busy Period of Aperiodic Messages

4 NUMERICAL EXAMPLES

Example.1 Consider a distributed system with 6 nodes connected by a WorldFIP network, and each node exists a periodic message and an aperiodic message (as defined in Table.2 and Table.3 respectively). BA regulates periodic messages according to schedule in Table.3.

Table.3 Example set of aperiodic message

Aperiodic message	A	B	C	D	E	F
Periodicity (ms)	2	3.1	3.4	4	5	8

Assume $C_{In}^i = 90\mu s$ and $C_{Tr}^i = 60\mu s$, then, $N_A^1 = 2$. Similarly, we can get N_A^i until $N_{Aw\_All}^{1,j} \le Len_{AB}$. Based on the above results, $Num_{AB} = 5$ is deduced.

Table.4 AMBPs starting at different microcycle

Microcycle	1	2	3	4	5
N_{AW}^i		5	4	4	3	4

Table.5 Worst_case response times of aperiodic messages

R_a^A	R_a^B	R_a^C	R_a^D	R_a^E	R_a^F
4.74	5.74	6.74	7.774	7.74	7.21

5 CONCLUSION

For hard realtime task, its WCRT must be less than its deadline, therefore deducing an exact WCRT is of importance. To our best knowledge, only Tover deeply address the issue of finding WCRT of aperiodic message. In this paper, we reduce pessimism in Tover's result through separating pending request indication and ongoing periodic transfer, and improving aperiodic message busy period. Further, a case study indicates tightly bound of WCRT for aperiodic message. My ongoing work will consider a pending request indicating multiple aperiodic messages to reduce pessimism deeper. Additionally, not all aperiodic messages have realtime requirement, and these messages are random in nature, therefore it is necessary to research them from average property to improve utilization of WorldFIP.

REFERENCE

1. H. Kopetz, Real Time System Design Principles for Distributed Embedded Application, *Kluwer Academic Publishers*, 1997
2. C. Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Application, *Kluwer Academic Publisher*, 1997
3. K. Tindell and A. Burns, Analysis of Hard Real-Time Communication, *Real-Time Systems*, 9(2), pp 147-173, 1997
4. K.C. Sevcik and J. Jonson, Cycle Time Properties of the FDDI Token Ring Protocol, *IEEE Trans on Soft.Eng*, Vol.13(3), pp 376-385,1987.
5. G. Agrawal, B. Chen and Wei Zhao, Guaranteeing Synchronous Message Deadline with Timed Token Medium Access Control Protocol, *IEEE Trans on Computer,* 43(3), pp 327-339, 1994.
6. A. Burns and H. Yang, Finding the Minimum Available Transmission Time for the Timed Token Medium Access, Control Protocol, *Procs of Euromicro Conference on Real-Time Systems*, pp185-194, 1999.
7. W. Zhao, J. Stankovic, and K. Ramamritham, A Window Protocol for Transmission of Time Constrained Messages, *IEEE Transactions on Computers*, Vol.39(9), pp1186- 1203, 1990.
8. K. M. Zuberi and K. G. Shin, Design and Implementation of Efficient Message Scheduling for CAN, *IEEE Trans. on Computers*, vol. 49(2), pp 182-188, 2000
9. J. Thomese, The Fieldbus, *Procs of Intelligent Components and Instruments for Control Application*, 1997, 13-23
10. E. Tovar, F. Vasques, Distributed Computing for the Factory-floor: a Real-Time Approach Using WorldFIP, *Computers in Industry*, Vol. 44(1), pp11-30, 2000.
11. Z. Wang, Modeling and Analysis of Fieldbus based Distributed Realtime System, *PHD Dissertation, Chinese Academy of Sciences,* 2000
12. Liu and Layland, Scheduling Algorithms for Multiprogramming in a Hard Realtime Environment, *Journal of ACM*, Vol.20 (1), pp40-61, 1973
13. Z. Wang, Y. Zhou, Y.X Sun and T.R. Wang, How to Support Realtime Traffic of Urgent Aperiodic Message Using Foundation Fieldbus, *IFAC Conference on New Technologies for Computer Control*, 2001
14. Z. Wang, Y.X Sun and T.R. Wang, Response Time of Urgent Aperiodic Message in Foundation Fieldbus, *International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2001
15. Y. Zhou, Z. Wang, and T.R. Wang, A New Method to Build the FF Schedule Time List, *IFAC Conference on New Technologies for Computer Control*, 2001
16. Y.Q. Song, Performance Analysis of Periodic and Aperiodic Real-Time Message Transmission in FIP Networks. *IFAC Parallel and distributed computing systems*, 1996.