Burst Mode Bandwidth Allocation for Real-Time Messages in IEEE 802.12 Networks

Taewoong Kim, Namyun Kim, Heonshik Shin, and Naehyuck Chang
Computer Systems Laboratory
School of Computer Science and Engineering
Seoul National University, Seoul 151-742, Korea
E-mail: `{twkim,nykim,shinhs,naehyuck}@cse.snu.ac.kr`
Phone: +82-2-874-1836
Fax: +82-2-874-3105

Abstract. In IEEE 802.12 network, a repeater arbitrates among transmission requests on a round robin basis. A node can transmit a packet only when it is granted by the repeater. Recently, the IEEE 802.12 Committee approved a *burst mode* in which a node can transmit multiple packets per grant. In burst mode a bandwidth, or the number of packets per grant should be allocated taking into account the timing constraints of real-time messages. This paper proposes bandwidth allocation algorithms based on the burst mode to guarantee the deadlines of periodic real-time messages. Once the bandwidth of each node is derived from the lengths and deadlines of periodic messages, a node is allowed to transmit packets up to the allocated bandwidth per grant. Experimental results show that the proposed algorithms provide much higher guarantee ratio and graceful degradation for heavy network load, compared with the existing approaches.

Keywords: IEEE 802.12 network, burst mode, bandwidth allocation, real-time communication

1. Introduction

Distributed computer systems are widely used to support real-time applications. The timely delivery of messages over the network is essential to such systems. In distributed real-time systems, messages may be characterized into two classes: periodic and aperiodic. Periodic messages are generated at regular intervals and must be delivered within their hard deadlines. The consequence of missing deadlines causes a system crash or loss of profit. Aperiodic messages irregularly arise from the external events. They are generally assumed to have soft deadlines, because aperiodic messages with hard deadlines can be modeled into periodic messages. This paper concentrates on the timely delivery of periodic messages over the IEEE 802.12 network.

The IEEE 802 project has developed two new 100 Mbps LAN standards: Fast Ethernet (IEEE Std. 802.3u, 1995) and IEEE 802.12 (IEEE Std. 802.12, 1995; Watson et al., 1995; Molle and Watson, 1996). While Fast Ethernet retains the CSMA/CD MAC protocol, the IEEE 802.12

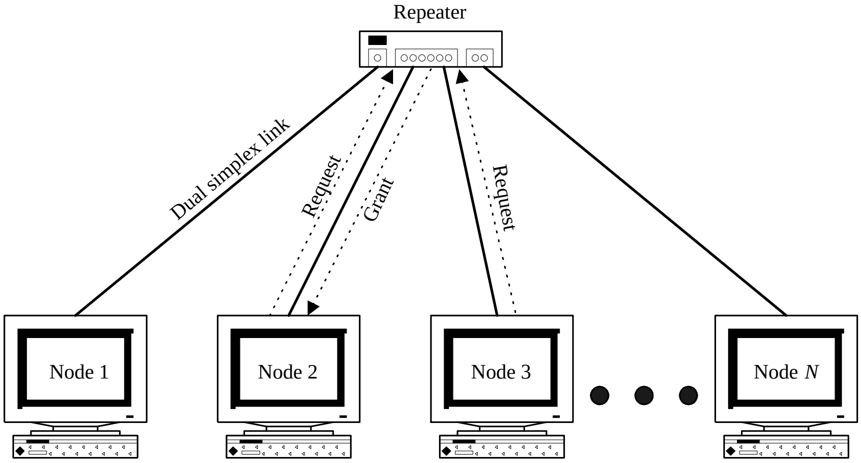


Figure 1. IEEE 802.12 network

standard provides a new MAC protocol called demand priority access method. This access method maximizes the network efficiency by eliminating packet collisions and token rotation delays. A simple IEEE 802.12 network consists of a repeater[1] and N nodes as shown in Figure 1. The physical links between the repeater and the nodes are shielded twisted pair or fiber optic cables configured as dual simplex.

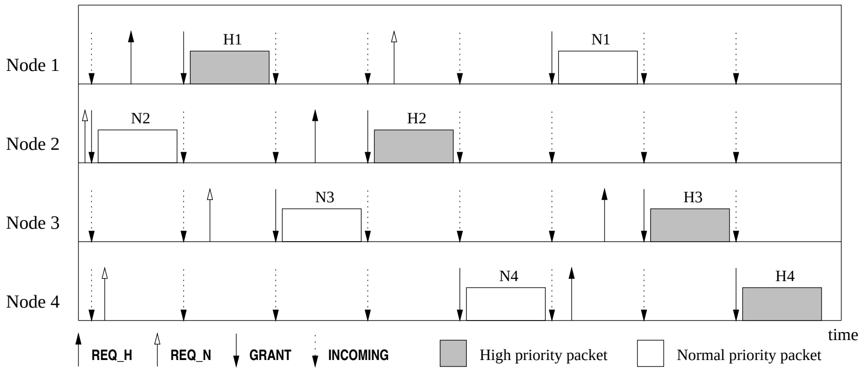
In demand priority access method, the repeater not only relays all frame transmissions among the nodes but also controls the network access from them. If a node has a packet to transmit, it first sends a request signal to the repeater. Prior to transmitting a packet, the node must wait until it has been granted permission by the repeater. To this end, the repeater polls its attached ports in cyclic round robin order to determine which has a pending request and then sends a grant signal to the requesting node. When the repeater sends a grant signal to a node, it also sends incoming signals to all the other nodes to make them ready to receive a packet.

The demand priority access method supports two priority levels of packet transmission: high and normal. A node indicates the priority of a packet by sending a request signal REQ_H or REQ_N to the repeater. All the high priority packets are always transmitted before normal priority packets. Thus, high priority packets may be used for time critical applications such as real-time video and audio. However, high priority packets are not allowed to interrupt the normal priority packet already in progress. Within the same priority level, the requests are served in

---

[1] In some literature, it is called a hub.

round robin order. To support this, the repeater maintains two next-port pointers, one for high priority and one for normal priority requests which retain the next ports to be served. Figure 2 shows an example of packet transmission in the IEEE 802.12 network. In this figure, the initial values of high priority and normal priority next-port pointers are node 1 and node 2, respectively. Because of the round robin polling, the granted node is selected by its location in the network rather than its request time. Since the network links between the repeater and nodes are dual simplex, a node can send a request signal to the repeater while it receives a packet from other nodes.



Request sequence: N2,N4,H1,N3,H2,N1,H4,H3

Packet sequence: N2,H1,N3,H2,N4,N1,H3,H4

Figure 2. An example of packet transmission sequence

The early IEEE 802.12 standard specifies a single packet mode only (IEEE Std. 802.12, 1995). In single packet mode, a node can send only one packet each time it is granted. Furthermore, a node is not allowed to transmit twice in a row if other nodes have the pending requests with the same priority. After transmission of a packet, the node should wait until other nodes with pending requests have transmitted their packets once. When a node has multiple packets to send, it has to exchange request/grant signals with the repeater for each packet transmission. Recently, to improve the efficiency of the network, the IEEE 802.12 standard adopted a *burst mode* in which a node can send multiple packets each time it is granted (IEEE Std. 802.12c, 1998).

Previous studies have focused on the support for multimedia traffic in single packet mode (Kim, 1998a; Kim, 1998b; Grinham and Spratt, 1993; Martini and Ottensmeyer, 1995; Watson et al., 1995). Because multimedia traffic does not have a stringent timing constraint, the network bandwidth allocation schemes are based on rate based approaches. In rate based approaches, a time interval is defined in terms of target transmission time (Grinham and Spratt, 1993; Watson et al., 1995) and time frame (Kim, 1998a; Kim, 1998b), respectively. For the time interval, all nodes can transmit high priority packets up to their allocated bandwidth. The amount of traffic transmitted over the time interval is used to determine whether the timing constraint of a real-time message is satisfied. However, these rate based methods may not be applicable to hard real-time messages. The above approaches cannot guarantee the deadlines of real-time messages shorter than the length of the time interval. Moreover, the rate based methods require some run-time overheads: target transmission time mechanism in (Grinham and Spratt, 1993) needs to maintain a timer to estimate the amount of traffic over the time interval. Time frame mechanism in (Kim, 1998a; Kim, 1998b) demands (σ, ρ) traffic regulator per real-time stream (Cruz, 1991). In (Martini and Ottensmeyer, 1995), the performance of IEEE 802.12 network has been evaluated for multimedia traffic; however, hard real-time traffic is not considered. For periodic hard real-time messages, we have proposed a priority-driven message scheduling algorithm in the previous study (Kim et al., 1998b). This approach enforces the transmission of periodic messages in priority order to guarantee the deadlines of periodic messages. But it requires a run-time overhead to broadcast the priorities of periodic messages. We have also proposed a bandwidth allocation algorithm for periodic real-time messages in burst mode (Kim et al., 1998a). This paper extends our previous research.

The objective of this paper is to develop bandwidth allocation algorithms in burst mode, such that the deadlines of periodic messages are guaranteed. Throughout this paper, high priority packets are designed to deliver periodic real-time messages, while normal priority packets are allocated to aperiodic messages and nonreal-time traffic. The bandwidth is denoted by the maximum number of high priority packets to be transmitted when a node is granted. First, we define three constraints that the bandwidth allocation algorithm should satisfy. Second, feasibility conditions of periodic messages are formulated in terms of the bandwidth. Then two bandwidth allocation algorithms are devised to satisfy the three constraints by exploiting the feasibility conditions.

The rest of this paper is organized as follows: Section 2 describes the assumptions and the model of periodic messages and defines the problem of bandwidth allocation. Section 3 analyzes basic and enhanced feasibility conditions and then presents two bandwidth allocation algorithms. Section 4 evaluates the performance of the proposed algorithms. The paper concludes with Section 5.

2. Problem statement

2.1. ASSUMPTIONS AND MESSAGE MODEL

We intend to develop a bandwidth allocation algorithm based on the following assumptions.

A1. *A network consists of a repeater and N nodes.*
This paper deals with the single repeater network only. However, the result can be applicable to the network with multiple repeaters.

A2. *Regardless of the priority of a packet, its length is fixed.*
The IEEE 802.12 network supports 802.3 or 802.5 MAC frame. The length of a packet is fixed to the maximum length of 802.3 MAC frame. Because of per-packet signalling overhead, the throughput of the network is maximized when the packet has its maximum length (Kim, 1998c).

A3. *The maximum length of a periodic message is known a priori.*
The IEEE 802.12 MAC layer segments a periodic message into multiple packets to transmit it over the network. From this assumption, we can obtain the maximum number of packets of a periodic message.

A4. *There are no promotions of the normal priority packets.*
The overload of the high priority packets may defer the transmission of the normal priority packets indefinitely. To handle this situation effectively, the IEEE 802.12 standard specifies that a normal priority packet can be promoted to high priority when a specified time has elapsed from its request time. However, the promotions of the normal priority packets make the network unpredictable.

A5. *A node transmits its normal priority packets in single packet mode.*
As previously mentioned, high priority packets cannot preempt the normal priority packet being transmitted. Consequently, the burst mode transmission of normal priority packets may block the transmission of high priority packets for unnecessarily long time. By transmitting the normal priority packets in single packet mode, such a blocking time is bounded to at most one packet transmission time.

A6. *The packet transmission is reliable.*
From this assumption, we do not consider the transmission errors such as packet loss, checksum error, and link failure. Packet retransmission may be a possible solution to transmission errors.

However, packet retransmission requires additional network bandwidth.

There is a set of periodic message streams $M = \{M_1, M_2, \ldots, M_N\}$, where M_i denotes a periodic message generated at node i. Each periodic message consists of multiple packets. M_i is characterized by (f_i, p_i, d_i), where f_i, p_i, and d_i denote the number of packets, the period, and the hard deadline relative to the arrival of M_i, respectively. In other words, each message M_i is generated at time t ($= k \cdot p_i$ where $k \geq 0$), and f_i packets should be transmitted by the time $t + d_i$. It is assumed that d_i is less than or equal to p_i. We can obtain f_i from Eq. (1).

$$f_i = \left\lceil \frac{\text{The maximum length of } M_i}{\text{The maximum payload length of a packet}} \right\rceil \tag{1}$$

Without loss of generality, it is assumed that there is a periodic message on each node, because multiple periodic messages on a node can be transformed into a logically equivalent network with a periodic message per node (Agrawal et al., 1992; Kim, 2000). If a node i has k periodic messages, we can split the node i into k logical nodes which have a periodic message respectively.

The packet transmission time is contributed by packet propagation delay and inter-packet gap[2]. The packet transmission time T_{pkt} is defined as follows:

$$T_{pkt} = \text{packet propagation delay} + \text{inter-packet gap}. \tag{2}$$

where packet propagation delay is defined as follows:

$$\frac{\text{packet length}}{\text{link rate}} + \underbrace{\frac{2 \times \text{link distance}}{\text{signal speed}}}_{\text{(link propagation delay)}} .$$

We obtain a link propagation delay from the sum of signal propagation delay from the source to the repeater and from the repeater to the destination. According to Table I, the packet transmission time T_{pkt} amounts to 129.44 μs. The inter-packet gap includes the time taken to exchange request/grant signals and the repeater arbitration time. Table II summarizes major notations used in this paper.

---

[2] IEEE 802.12 specifies that inter-packet gap is 202-210 BT. A BT (bit time) is equal to 1/TxClk where TxClk is 30 Mhz (IEEE Std. 802.12, 1995).

Table I. Network parameters

Parameter	Description
Packet length	Header (18 bytes) + Payload (1500 bytes) = 12144 bits
Link rate	1×10^8 bits/s (100 Mbps)
Link distance	100 m
Signal speed	2×10^8 m/s
Inter-packet gap	7 μs

Table II. A summary of notations

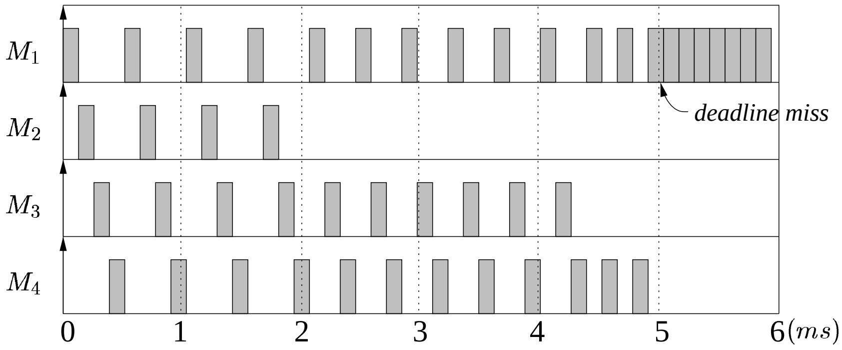
Symbol	Description
N	The number of nodes
M_i	A periodic message generated at node i
f_i	The number of packets of M_i
p_i	The period of M_i
d_i	Relative hard deadline of M_i
u_i	The network utilization of M_i
T_{pkt}	Packet transmission time
b_i	Bandwidth: the maximum number of high priority packets to be transmitted per grant at node i
R_i	The worst case response time of M_i
T_{IG}	The maximum inter-grant time
$BL_i(t)$	The number of pending high priority packets on node i at time t

2.2. PROBLEM DEFINITION

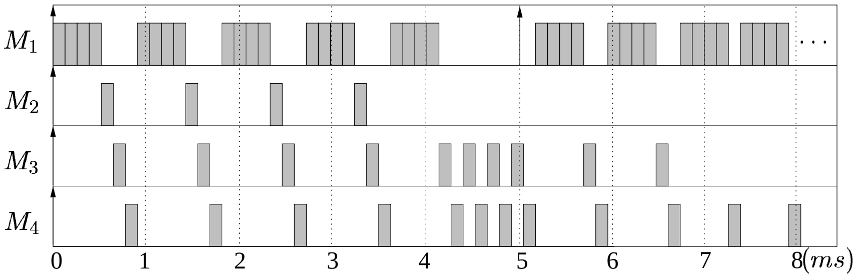
Let us define an inter-grant time as the time taken for a node with pending requests to receive the next grant signal upon its receipt of a grant signal. In single packet mode, each node can transmit a packet every $N \cdot T_{pkt}$ time units in the worst case, because the maximum inter-grant time amounts to $N \cdot T_{pkt}$ time units. The burst mode transmission may increase the inter-grant time, because a node is allowed to transmit one or more packets. In burst mode, a node receives smaller number of grants for a given time interval than the single packet mode. However, the burst mode transmission can improve the schedulability of periodic messages. Since a node transmits multiple packets per grant, it can complete the transmission of its periodic message earlier than it operates in single packet mode.

Table III. A set of periodic messages

Message	Number of packets	Period	Deadline
M_1	20	5 ms	5 ms
M_2	4	10 ms	8 ms
M_3	10	20 ms	8 ms
M_4	12	30 ms	10 ms



(a) Single packet mode



(b) Burst mode

Figure 3. Transmission of the messages given in Table III

For example, consider a set of periodic messages given in Table III. We assume that all the periodic messages in Table III arrive simultaneously at time 0. In Figure 3, the up-arrows represent the arrival of periodic messages and request/grant signaling is not depicted for brevity. As shown in Figure 3(a), the single packet mode cannot guarantee the deadline of M_1, whereas the burst mode transmission can guarantee the deadlines of all periodic messages if node 1 transmits four packets per grant. Although the burst mode transmission in Figure 3(b) increases the response time of M_2, M_3, and M_4 compared to Figure 3(a), these messages are still transmitted within the corresponding deadlines.

To guarantee the deadlines of periodic messages, we should control the burst mode transmission with caution. If a granted node monopolizes the network, the other nodes may not be able to transmit their periodic messages within their deadlines. Thus, it is necessary to restrict the number of packets to be transmitted whenever a node is granted. In this paper, the bandwidth is defined as the maximum number of high priority packets to be transmitted per grant. This paper then addresses the problem of bandwidth allocation such that all periodic messages can meet their deadlines. Let b_i denote the bandwidth of node i. In burst mode, the simplest way of bandwidth allocation may be to assign b_i to f_i: A node i is allowed to transmit the whole M_i at a time. We call this strategy FLA (full length allocation) scheme. However, FLA scheme provides poor performance because it makes the inter-grant time relatively long. The FLA scheme cannot guarantee the deadlines if the maximum inter-grant time is greater than the deadlines of periodic messages. The performance of the FLA scheme will be presented in Section 4.

To logically develop more effective method we begin by identifying the constraints to be satisfied as follows:

C1. *Bandwidth constraint*: $\forall i, 1 \leq b_i \leq f_i$
C2. *Deadline constraint*: $\forall i, T_{IG} \leq d_i$
C3. *Feasibility constraint*: $\forall i, R_i \leq d_i$.

where T_{IG} and R_i denote the maximum inter-grant time and the worst case response time of M_i, respectively. The response time of a periodic message is defined as the elapsed time from its arrival to the completion of transmission. The bandwidth constraint is necessary to avoid excessive allocation of the bandwidth. The deadline constraint ensures that M_i has at least one network access before its deadline, while the feasibility constraint guarantees that M_i meets d_i in the worst case. A brute-force searching algorithm takes $O(m^N)$ time to find a feasible bandwidth $(b_1, b_2, \ldots, b_N)$ which satisfies the above three constraints, where m denotes $\max(f_1, f_2, \ldots, f_N)$. Thus, we need more efficient bandwidth allocation algorithm to find a feasible bandwidth $(b_1, b_2, \ldots, b_N)$. Once b_i is determined, node i can transmit up to b_i high priority packets per grant.

3. Bandwidth allocation for real-time messages

Among the three constraints shown in Section 2.2, the feasibility constraint requires the computation of the worst case response time of a periodic message. First, we present two feasibility analyses to obtain the worst case response time of a periodic message. These feasibility analyses are used to determine whether periodic messages can meet their deadlines for an arbitrary $(b_1, b_2, \ldots, b_N)$ satisfying the constraints C1 and C2. And then we propose two bandwidth allocation algorithms to determine the bandwidth $(b_1, b_2, \ldots, b_N)$ and analyze their time complexities, respectively.

3.1. BASIC ANALYSIS OF FEASIBILITY

Before deriving the worst case response time of periodic messages, we first define the worst case scenario which causes the longest response time of periodic message M_i.

The worst case scenario

1. At time t, the repeater grants node j $(1 \leq j \leq N)$ to transmit a normal priority packet, and the value of high priority next-port pointer is equal to $(i \bmod N)+1$.

2. As soon as node j receives a grant signal, all periodic messages $M_1, M_2, \ldots$, and M_N arrive simultaneously.

Since node j receives a grant signal before the arrival of periodic messages, it begins the transmission of normal priority packet at time t. From time $t + T_{pkt}$, normal priority packets cannot be transmitted until the pending high priority packets have been transmitted. At time $t + T_{pkt}$, the repeater sends a grant signal to the node $(i \bmod N)+1$. In this case, node i experiences the longest waiting time before sending the first packet of M_i and continues to contend with all the other nodes for transmission of high priority packets. Thus, the periodic message M_i has the worst case response time.

We now consider the inter-grant time to derive the worst case response time of M_i. The inter-grant time dominates the response time of M_i, because only the granted node can transmit packets over the network. In the worst case, the round robin arbitration causes a node to wait for all the other nodes to send their packets. In burst mode, the maximum inter-grant time, or T_{IG} amounts to the sum of transmission times contributed by all nodes.

$$T_{IG} = \sum_{i=1}^{N} b_i \cdot T_{pkt} \tag{3}$$

To obtain the worst case response time of M_i, it is assumed that the inter-grant time is fixed to its maximum value T_{IG}. In other words, if a node i receives a grant signal from the repeater at time t, then it will receive another grant signal at time $t + T_{IG}$. Because this assumption is too pessimistic, it will be relaxed in Section 3.2.

When the inter-grant time is fixed to T_{IG}, the worst case response time M_i can be obtained as follows: From the worst case scenario, node i can send its packets after the other nodes have sent their packets as shown in Figure 4. Thus, node i experiences the longest waiting time before sending the first packet of M_i. This waiting time is equal to $T_{pkt} + (T_{IG} - b_i \cdot T_{pkt})$ time units. Node i has to receive $\lceil \frac{f_i}{b_i} \rceil$ grants to complete the transmission of M_i. It takes $(\lceil \frac{f_i}{b_i} \rceil - 1) \cdot T_{IG}$ time units for node i to receive the $\lceil \frac{f_i}{b_i} \rceil$-th grant. On receiving the $\lceil \frac{f_i}{b_i} \rceil$-th grant, node i transmits $(f_i - (\lceil \frac{f_i}{b_i} \rceil - 1) \cdot b_i)$ packets since it has already transmitted $(\lceil \frac{f_i}{b_i} \rceil - 1) \cdot b_i$ packets. Thus, we can obtain the worst case response time of M_i, or R_i as follows:

$$\begin{aligned} R_i &= T_{pkt} + (T_{IG} - b_i \cdot T_{pkt}) + (\left\lceil \frac{f_i}{b_i} \right\rceil - 1) \cdot T_{IG} + \left(f_i - (\left\lceil \frac{f_i}{b_i} \right\rceil - 1) \cdot b_i \right) \cdot T_{pkt} \\ &= \left\lceil \frac{f_i}{b_i} \right\rceil \cdot (T_{IG} - b_i \cdot T_{pkt}) + (f_i + 1) \cdot T_{pkt}. \end{aligned} \tag{4}$$

If R_i is less than or equal to d_i for a given $(b_1, b_2, \ldots, b_N)$, M_i is always transmitted within its deadline.

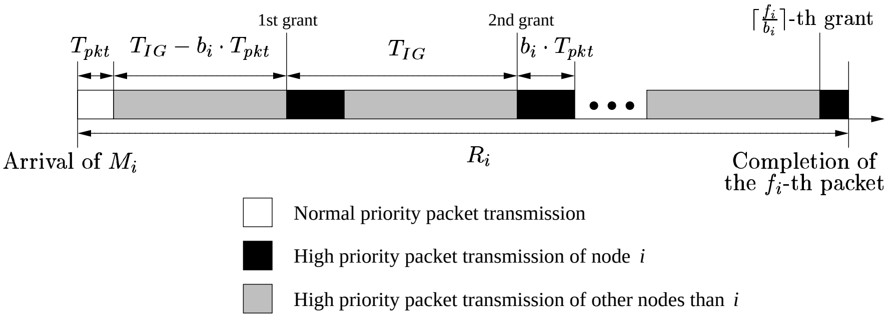


Figure 4. The worst case packet transmission of M_i

3.2. ENHANCED ANALYSIS OF FEASIBILITY

In the basic feasibility analysis, R_i is computed under the assumption that the inter-grant time is fixed to T_{IG}. In general, however, periodic messages have different periods and numbers of packets one another. Hence, the inter-grant time at a node may vary according to the arrival and completion of periodic messages on the other nodes. If we calculate the inter-grant time exactly, we can obtain more accurate response time of periodic messages. For example, Figure 5 shows the influence of the inter-grant time on the worst case response time of a periodic message. Under the worst case scenario described in Section 3.1, the timeline in Figure 5 shows the worst case response time of M_1 where $(f_1, f_2, f_3, f_4) = (10, 2, 6, 3)$ and $(b_1, b_2, b_3, b_4) = (2, 2, 3, 1)$, respectively. The up-arrow represents the arrival of a periodic message. The regions (a), (b), (c), and (d) represent the various inter-grant time at node 1: $6T_{pkt}, 3T_{pkt}, 2T_{pkt}$, and $4T_{pkt}$ time units, respectively. Moreover, the inter-grant time in region (c) amounts to $b_1 \cdot T_{pkt}$ time units since there is no pending high priority packet on nodes 2, 3, and 4.

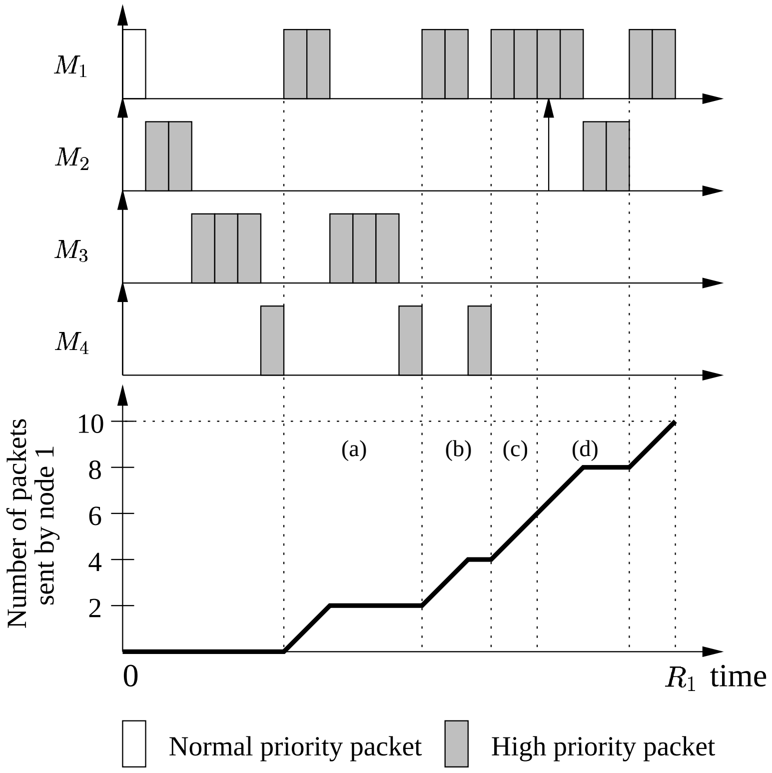


Figure 5. More accurate computation of R_1

We need to consider variable inter-grant time to obtain the accurate value of R_i. Let $BL_i(t)$ denote the number of pending high priority packets on node i at time t. $BL_i(t)$ is necessary to keep track of packet transmission behavior of node i. At time t, we can compute $BL_i(t)$ by subtracting the number of transmitted high priority packets from the number of arriving high priority packets. Hence, $BL_i(t)$ can be written as follows:

$$BL_i(t) = \left(\left\lfloor \frac{t}{p_i} \right\rfloor + 1 \right) \cdot f_i - \sigma_i. \tag{5}$$

where σ_i denotes the cumulative number of high priority packets sent by node i up to time t. It is initialized to be zero at time 0 and is increased as many as node i sends the high priority packets. Next, let n_i denote the number of high priority packets to be transmitted by node i when node i is granted at time t. Since node i cannot send more than b_i packets per grant, n_i can be written as follows:

$$n_i = \min(b_i, BL_i(t)). \tag{6}$$

From Eqs. (5) and (6), we can calculate the worst case response time R_i precisely. Figure 6 shows the algorithm which computes R_i.

```
Algorithm Compute_$R_i$
input: $M = \{M_i | M_i = (f_i, p_i, d_i),\ 1 \le i \le N\}$
       Bandwidth $(b_1, b_2, \ldots, b_N)$
output: $R_i$
begin
    for $j = 1$ to $N$ do $\sigma_j = 0$;
    $t = T_{pkt}$;
    for $j = 1$ to $\lceil \frac{f_i}{b_i} \rceil$ do begin
        for $l = 0$ to $N - 1$ do begin
            $k = ((i + l) \bmod N) + 1$;
            $n_k = \min(b_k, BL_k(t))$;
            $\sigma_k = \sigma_k + n_k$;
            $t = t + n_k \cdot T_{pkt}$;
        end for
    end for
    $R_i = t$;
    return $R_i$;
end
```

Figure 6. Computation of R_i

In Figure 6, the variable t denotes the elapsed time from the arrival of periodic message M_i. First, the algorithm initializes the value of

$(\sigma_1, \sigma_2, \ldots, \sigma_N)$. And then t is initialized to T_{pkt} to include the blocking time caused by transmission of a normal priority packet. The **for** loop (**Line 3-10**) iterates $\lceil f_i/b_i \rceil$ times since M_i requires $\lceil f_i/b_i \rceil$ grants to transmit M_i. The **for** loop (**Line 4-9**) computes the elapsed time from the arrival of M_i to completion of transmission when node i receives the j-th grant. The statement in **Line 5** makes node i send its packets according to the worst case scenario. If a node k receives a grant signal at time t, it sends n_k packets and n_k is added to σ_k.

3.3. BANDWIDTH ALLOCATION ALGORITHM

In this section, we present two bandwidth allocation algorithms and analyze their time complexities. The proposed bandwidth allocation algorithms are called a SBA (smallest bandwidth allocation) algorithm and a PBA (proportional bandwidth allocation) algorithm, respectively. The SBA algorithm is devised for offline bandwidth allocation whereas the PBA algorithm is for online bandwidth allocation.

First, we describe the SBA algorithm. The SBA algorithm exploits searching with backtracking to find a feasible bandwidth $(b_1, b_2, \ldots, b_N)$ such that all periodic messages in M meet their deadlines. We try to find a feasible bandwidth while the constraints C1 and C2 are met. If either C1 or C2 is not satisfied, there is no feasible bandwidth allocation for a given set of periodic messages. We start the search with $(b_1, b_2, \ldots, b_N) = (1, 1, \ldots, 1)$. For a given $(b_1, b_2, \ldots, b_N)$, if the constraint C3 of M_i is met (i.e., $R_i \leq d_i$), we mark the message M_i schedulable and proceed to check the constraint C3 of the next message M_{i+1}. Otherwise (i.e., $R_i > d_i$), we increase b_i by 1 to reduce R_i and then backtrack. The reason of backtracking is that the increase of b_i may change the worst case response times of $M_1, M_2, \ldots$, and M_i which have been computed already. Figure 7 shows the outline of the SBA algorithm.

First of all, we set the initial value of each b_i to 1 and then compute T_{IG} (**Line 1**). Hence, the initial value of T_{IG} is equal to $N \cdot T_{pkt}$ (**Line 2**). Next, for each b_i, $1 \leq i \leq N$, we inspect whether the constraints C1, C2, and C3 are met. If either C1 or C2 is not met, the algorithm terminates and returns FAILURE (**Line 6**). If the constraint C3 is satisfied, the algorithm marks the message M_i schedulable (**Line 7**). Otherwise (i.e., $R_i > d_i$), we increase b_i by 1 to reduce R_i (**Line 9**). By definition, the increment of b_i increases T_{IG} (**Line 10**). Moreover, it also affects $R_1, R_2, \ldots$, and R_i that have been examined already. Hence, the algorithm should backtrack (**Line 12**). Note that **break** statement causes the **for** loop (**Line 5-14**) to exit immediately and enables the **repeat** loop (**Line 4-15**) to start a new iteration. The

```
Algorithm SBA
input: $M = \{M_i | M_i = (f_i, p_i, d_i),\ 1 \le i \le N\}$
output: $(b_1, b_2, \ldots, b_N)$
begin
    for $i = 1$ to $N$ do $b_i = 1$;
    $T_{IG} = \sum_{i=1}^{N} b_i \cdot T_{pkt}$;
    mark all periodic messages unschedulable;
    repeat
        for $i = 1$ to $N$ do begin
            if ($b_i > f_i$ or $T_{IG} > d_i$) return FAILURE;
            if ($R_i \le d_i$) mark $M_i$ schedulable;
            else
                $b_i = b_i + 1$;
                $T_{IG} = T_{IG} + T_{pkt}$;
                mark $M_i$ unschedulable;
                break;
            endif
        end for
    until (all periodic messages are schedulable)
    return SUCCESS;
end
```

Figure 7. Smallest bandwidth allocation algorithm

algorithm repeats the above procedure until all messages satisfy the constraint C3. The SBA algorithm never decreases the value of b_i, since the decrement of b_i does not reduce R_i. By doing this, the SBA algorithm skips unnecessary feasibility check. According to Figure 7, the SBA algorithm eventually checks the feasibility of all bandwidth pairs that satisfy constraints C1 and C2, if it does not find a feasible bandwidth $(b_1, b_2, \ldots, b_N)$. If the SBA algorithm returns FAILURE, there is no feasible bandwidth allocation. Thus, we can insist that the SBA algorithm should find a feasible bandwidth whenever it exists. Once the bandwidth b_i is determined at pre-run-time, there is no run-time overhead, while the rate based mechanisms in (Grinham and Spratt, 1993; Watson et al., 1995; Kim, 1998a; Kim, 1998b) have to maintain a timer or traffic regulator at run-time.

Tables IV and V show the behavior of the SBA algorithm for the message set given in Table III. Note that the underlined worst case response time is greater than the corresponding deadline. In Table IV, R_1, R_2, R_3, and R_4 are computed using Eq. (4). At each step, we check to see if R_i is less than or equal to d_i. In Step 1, since R_1 is greater than

Table IV. The SBA algorithm using basic feasibility condition

Step	(b_1, b_2, b_3, b_4)	R_1	R_2	R_3	R_4	Action
1	(1,1,1,1)	$\underline{10.484}$ *ms*	-	-	-	increase b_1
2	(**2**,1,1,1)	$\underline{6.601}$ *ms*	-	-	-	increase b_1
3	(**3**,1,1,1)	$\underline{5.436}$ *ms*	-	-	-	increase b_1
4	(**4**,1,1,1)	4.659 *ms*	3.753 *ms*	$\underline{9.190}$ *ms*	-	increase b_3
5	(4,1,**2**,1)	$\underline{5.307}$ *ms*	-	-	-	increase b_1
6	(**5**,1,2,1)	4.789 *ms*	4.789 *ms*	5.954 *ms*	$\underline{14.108}$ *ms*	increase b_4
7	(5,1,2,**2**)	$\underline{5.307}$ *ms*	-	-	-	increase b_1
8	(**6**,1,2,2)	$\underline{5.307}$ *ms*	-	-	-	increase b_1
9	(**7**,1,2,2)	4.659 *ms*	6.342 *ms*	7.895 *ms*	9.449 *ms*	terminate

Table V. The SBA algorithm using enhanced feasibility condition

Step	(b_1, b_2, b_3, b_4)	R_1	R_2	R_3	R_4	Action
1	(1,1,1,1)	$\underline{6.083}$ *ms*	-	-	-	increase b_1
2	(**2**,1,1,1)	$\underline{5.824}$ *ms*	-	-	-	increase b_1
3	(**3**,1,1,1)	$\underline{5.048}$ *ms*	-	-	-	increase b_1
4	(**4**,1,1,1)	4.530 *ms*	3.753 *ms*	7.378 *ms*	8.672 *ms*	terminate

d_1, the algorithm continues to increase b_1 and recalculate R_1 until R_1 is less than or equal to d_1. In Step 4, we mark M_1 schedulable and check the feasibility of M_2 and M_3. We increase b_3, because R_3 is greater than d_3. In Step 5, the increment of b_3 makes R_1 greater than d_1. In Step 6, we check the feasibility of M_1, M_2, and M_3 again and increase b_4 to reduce R_4. In Steps 7 and 8, we increase b_1 since R_1 is greater than d_1. Finally, we obtain the resulting bandwidth $(b_1, b_2, b_3, b_4) = (7, 1, 2, 2)$.

In Table V, R_1, R_2, R_3, and R_4 are computed by the algorithm in Figure 6. Note that R_i's in Table V are more accurate than those in Table IV. The result of bandwidth allocation is $(b_1, b_2, b_3, b_4) = (4, 1, 1, 1)$. When an enhanced feasibility condition is used, the SBA algorithm allocates a smaller bandwidth and finds the final solution faster.

There might be multiple feasible bandwidth pairs for a given periodic messages. The SBA algorithm returns a feasible bandwidth pair with the smallest value. In case of Table V, for example, another bandwidth pair $(4,2,1,1)$ can also transmit all messages within their deadlines. However, a bandwidth pair $(4,2,1,1)$ overallocates the network bandwidth for M_2 compared with $(4,1,1,1)$. For efficiency and saving of network bandwidth, the SBA algorithm terminates as soon as it finds a feasible bandwidth pair.

We analyze the time complexity of the SBA algorithm. Let c_i denote the number of computation of R_i to determine the bandwidth b_i. The total number of iterations of the SBA algorithm is $\sum_{i=1}^{N} c_i$. According to the SBA algorithm, we need to recompute $R_1, R_2, \ldots, R_i$ whenever we increase b_i. Thus, c_i can be obtained as follows:

$$c_i = \begin{cases} b_i + \sum_{j=i+1}^{N}(b_j - 1) & 1 \leq i \leq N-1 \\ b_i & i = N. \end{cases} \qquad (7)$$

We can prove the correctness of Eq. (7) by using mathematical induction.
Basis: For a given $(b_1, b_2, \ldots, b_N)$, if $\forall b_i = 1$, then $\forall c_i = 1$ where $1 \leq i \leq N$.
Hypothesis: When $b_i = k \geq 2$ where $1 \leq i \leq N$,

$$\begin{aligned} c_1 &= b_1 + (b_2 - 1) + \cdots + \overbrace{(k-1)}^{(b_i-1)} + \cdots + (b_N - 1) \\ c_2 &= b_2 + (b_3 - 1) + \cdots + \overbrace{(k-1)}^{(b_i-1)} + \cdots + (b_N - 1) \\ & \ldots \\ c_i &= \overbrace{\ k\ }^{b_i} + (b_{i+1} - 1) + (b_{i+2} - 1) + \cdots + (b_N - 1) \\ & \ldots \\ c_{N-1} &= b_{N-1} + (b_N - 1) \\ c_N &= b_N. \end{aligned}$$

Induction: When $b_i = k + 1$, the SBA algorithm recomputes only $R_1, R_2, \ldots,$ and R_i. From hypothesis, we can write $(c_1, c_2, \ldots, c_N)$ as follows:

$$\begin{aligned} c_1 &= b_1 + (b_2 - 1) + \cdots + (k-1) + \cdots + (b_N - 1) + 1 \\ c_2 &= b_2 + (b_3 - 1) + \cdots + (k-1) + \cdots + (b_N - 1) + 1 \\ & \ldots \\ c_i &= k + (b_{i+1} - 1) + (b_{i+2} - 1) + \cdots + (b_N - 1) + 1 \\ & \ldots \\ c_{N-1} &= b_{N-1} + (b_N - 1) \\ c_N &= b_N. \end{aligned}$$

We can rewrite the above equations as given in Eq. (7).

From Eq. (7), we obtain the total number of iterations:

$$\sum_{i=1}^{N} c_i = \sum_{i=1}^{N} b_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (b_j - 1). \tag{8}$$

By replacing b_i's with $b_{max} = \max(b_1, b_2, \ldots, b_N)$, Eq. (8) can be rewritten as

$$\begin{aligned}\sum_{i=1}^{N} c_i &\leq \sum_{i=1}^{N} b_{max} + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (b_{max} - 1) \\ &= b_{max} \cdot N + \sum_{i=1}^{N-1} (N-i) \cdot (b_{max} - 1) \\ &= \frac{(b_{max} - 1)}{2} \cdot N^2 + \frac{(b_{max} + 1)}{2} \cdot N.\end{aligned}$$

Consequently, the SBA algorithm has $O(mN^2)$ time complexity, where m and N denote b_{max} and the number of periodic messages, respectively. Because the SBA algorithm is evaluated prior to run-time, its complexity is acceptable. However, the time complexity of the SBA algorithm is not acceptable for online bandwidth allocation of periodic messages.

Now, we introduce the PBA (proportional bandwidth allocation) algorithm for online bandwidth allocation and admission control. The network utilization of M_i denoted by u_i is defined as $\frac{f_i \cdot T_{pkt}}{p_i}$. Let M_l represent the periodic message with the smallest network utilization, i.e. $u_l = \min(u_1, u_2, \ldots, u_N)$. Since the bandwidth of periodic message is proportional to its network utilization, each b_i can be set relative to the bandwidth of M_l, or b_l as shown in Eq. (9).

$$b_i = \left\lfloor \frac{u_i \cdot b_l}{u_l} \right\rfloor \tag{9}$$

From C1 constraint $b_i \leq f_i$, Eq. (9) can be rewritten:

$$b_i = \left\lfloor \frac{u_i \cdot b_l}{u_l} \right\rfloor \leq f_i \Rightarrow \frac{u_i \cdot b_l}{u_l} < f_i + 1 \Rightarrow b_l < \frac{(f_i + 1) \cdot u_l}{u_i}.$$

Since the bandwidth b_l denotes the number of high priority packets, b_l should be integer. Thus, b_l must satisfy the following inequalities.

$$\forall i, i \neq l, b_l \leq \left\lfloor \frac{(f_i + 1) \cdot u_l}{u_i} \right\rfloor \text{ and } b_l \leq f_l \tag{10}$$

Therefore, we can obtain the range of b_l as follows:

$$1 \leq b_l \leq b_l^{max}. \tag{11}$$

where $b_l^{max} = \min\left(f_l, \min\{\lfloor\frac{(f_i+1)\cdot u_l}{u_i}\rfloor | i \neq l\}\right)$. Because each of b_i's is relative to b_l, a feasible bandwidth can be determined much faster. Figure 8 illustrates the outline of the PBA algorithm.

```
Algorithm PBA
input: $M = \{M_i | M_i = (f_i, p_i, d_i),\ 1 \leq i \leq N\}$
output: $(b_1, b_2, \ldots, b_N)$
begin
1:      mark all periodic messages unschedulable;
2:      find $M_l$ and compute $b_l^{max}$;
3:      $b_l = 1$;
4:      while ($b_l \leq b_l^{max}$) do begin
5:          for $i = 1$ to $N$ do $b_i = \left\lfloor \frac{u_i \cdot b_l}{u_l} \right\rfloor$;
6:          compute $T_{IG}$;
7:          for $i = 1$ to $N$ do begin
8:              if ($T_{IG} > d_i$) return FAILURE;
9:              if ($R_i \leq d_i$) mark $M_i$ schedulable;
10:             else
11:                 $b_l = b_l + 1$;
12:                 mark $M_i$ unschedulable;
13:                 break;
14:             endif
15:         end for
16:         if (all periodic messages are schedulable)
17:             return SUCCESS;
18:         endif
19:     end while
20:     return FAILURE;
end
```

Figure 8. Proportional bandwidth allocation algorithm

The PBA algorithm asserts the constraint C1 to be satisfied by the **while** loop (**Line 4-19**). And the constraints C2 and C3 are examined in **Line 8** and **Line 9**, respectively. According to Figure 8, the PBA algorithm has the time complexity of $O(mN)$, where m and N denote b_l^{max} and the number of periodic messages, respectively. The PBA algorithm has a lower time complexity than the SBA algorithm, so it can be used for online bandwidth allocation and admission control. Suppose that there are N periodic messages and we have a new periodic message

M_{N+1}. We need to determine whether M_{N+1} can be transmitted within its deadline while guaranteeing the existing N periodic messages. If there is a feasible bandwidth pair $(b_1, b_2, \ldots, b_N, b_{N+1})$, we admit a periodic message M_{N+1}. Otherwise, we reject M_{N+1}. By using the PBA algorithm, we can check if a feasible bandwidth pair exists or not. It is noteworthy, however, that the PBA algorithm may not allocate feasible bandwidth for a given message set whereas the SBA algorithm can do so. For example, the PBA algorithm returns FAILURE for the message set given in Table III. The performance of the SBA and PBA algorithms will be discussed in the next section.

4. Performance evaluation

4.1. EXPERIMENTAL SETUP

Various sets of periodic messages have been generated randomly to evaluate the performance of the proposed algorithms. It is assumed that the deadline of a periodic message equals its period, and the length and period of the message have an exponential distribution. The mean length and period of messages are 100 Kbytes and 100 ms, respectively. In the experiment, varying values are assigned to the network utilization of periodic messages and the number of nodes. We choose five different degrees of network utilization of periodic messages: 50%, 60%, 70%, 80%, and 90%. The number of nodes considered is 4, 8, 12, and 16. We generated 10,000 sets of periodic messages for each pair of $\{50\%, 60\%, 70\%, 80\%, 90\%\} \times \{4, 8, 12, 16\}$. The cardinality of each set corresponds to the number of nodes.

We have conducted the simulations to measure the *guarantee ratio* of an algorithm. A set of periodic messages is said to be guaranteed, if all the periodic messages in the set meet their deadlines. The guarantee ratio of an algorithm is defined as follows:

$$\frac{\text{The number of message sets guaranteed by the algorithm}}{\text{The total number of message sets (10,000)}}.$$

Obviously, the higher the guarantee ratio becomes, the more sets of periodic messages the algorithm can transmit. We consider the following algorithms in the simulation.

SPM Single packet mode: It also represents the rate based bandwidth allocation methods based on the single packet mode.

FLA Full length allocation scheme.

RMS Periodic message transmission in rate monotonic priority order (Liu and Layland, 1973): In this scheme periodic messages are transmitted based on scheduling mechanism proposed in (Kim et al., 1998b).

SBA-B SBA scheme with basic feasibility condition.

SBA-E SBA scheme with enhanced feasibility condition.

PBA Proportional bandwidth allocation scheme.

4.2. EXPERIMENTAL RESULTS

The experimental results are given in Figure 10 and Figure 11. First, we shall discuss two extreme cases of bandwidth allocation: SPM and FLA. The guarantee ratio of SPM decreases abruptly as the network utilization increases; also it decreases as the number of nodes increases. In the SPM, a node should receive the grant signals as many as the number of packets to be transmitted. Consequently, the response time of a periodic message depends on its number of packets and the inter-grant time. Because the number of packets is proportional to the network utilization and the inter-grant time is to the number of nodes, the SPM shows a low guarantee ratio in case of a large number of nodes with high network utilization.

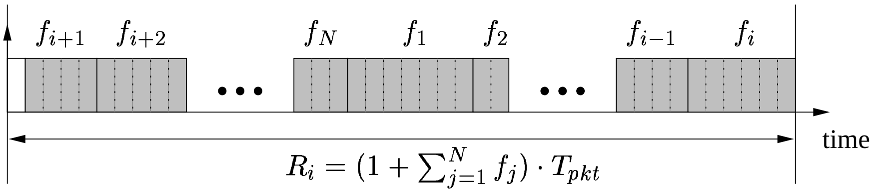


Figure 9. R_i under the FLA scheme

The guarantee ratio of FLA scheme is much lower than those of other algorithms. In the FLA scheme, the time at which a node receives the grant signal dominates the response time of its periodic message since the bandwidth is allocated to transmit the whole message per grant. In the worst case, a node i should wait for $(1 + \sum_{j \neq i} f_j) \cdot T_{pkt}$ time units to receive a grant signal. Thus, the worst case response time R_i can be obtained as shown in Figure 9. The low guarantee ratio of FLA scheme is primarily due to the long waiting time. However, FLA scheme can provide better guarantee ratio when network utilization is very low or the number of packets is relatively small.

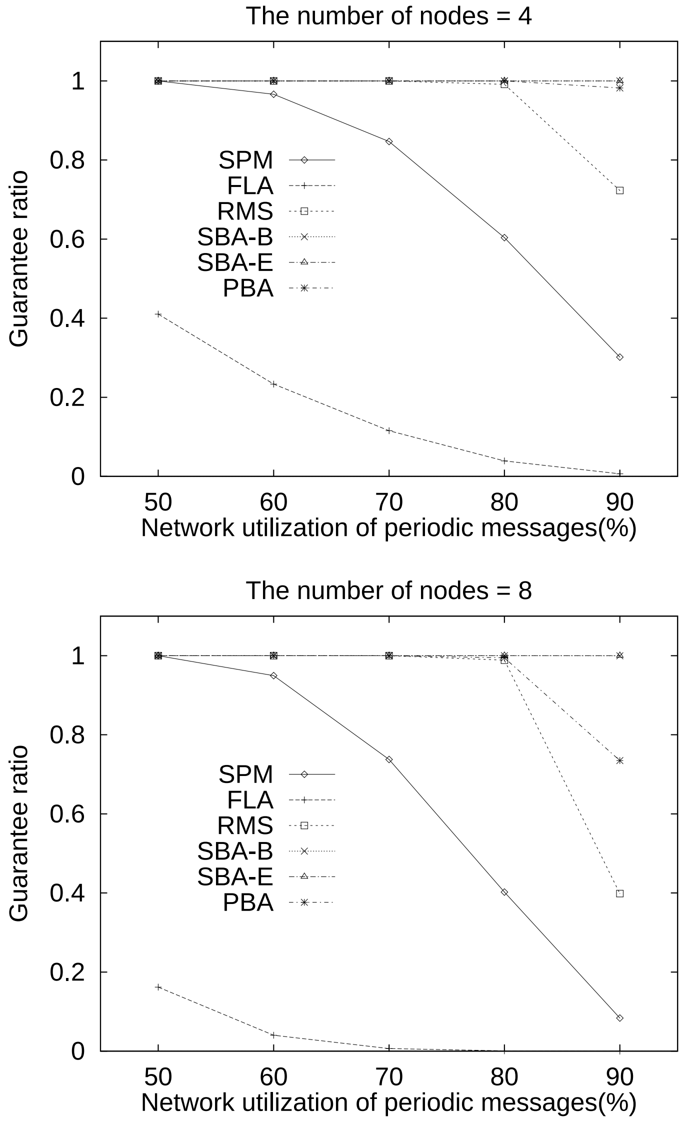


Figure 10. Guarantee ratios: 4 nodes and 8 nodes

By virtue of priority broadcast, the RMS scheme supports a priority-driven preemptive message transmission on a packet basis (Kim et al., 1998b). The guarantee ratio of RMS is nearly 100% for the network utilization lower than 80%. It is shown that rate monotonic task scheduling has 88% utilization bound in the average case (Lehoczky et al., 1989). Similar to task scheduling, message transmission using RMS results in a low guarantee ratio for 90% network utilization. Moreover, RMS has run-time overhead of priority broadcast over the network, although such a broadcast requires low network bandwidth (Kim et al., 1998b).

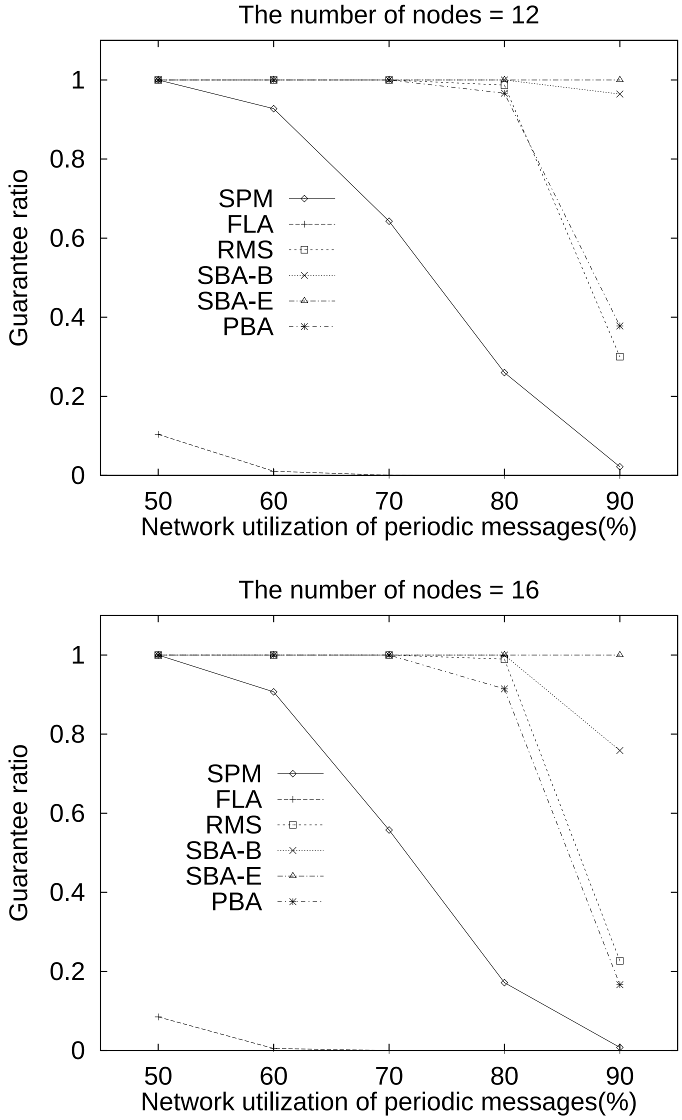


Figure 11. Guarantee ratios: 12 nodes and 16 nodes

As shown in Figure 10, the SBA-B and SBA-E schemes provide similar guarantee ratios when the number of nodes is 4 or 8. As the number of nodes increases, the SBA-B has slightly lower guarantee ratio than that of SBA-E in case of high network utilization (See Figure 11). Because the SBA-E makes use of more accurate feasibility condition than the SBA-B, it allocates the bandwidth efficiently. The SBA-E admits more sets of periodic messages than the SBA-B and does not waste the network bandwidth. In summary, the experimental results show that the SBA-B and SBA-E algorithms significantly improve the guarantee ratio compared to the existing SPM and RMS schemes.

According to the experimental results, the guarantee ratio of the PBA algorithm approximates that of the SBA algorithm up to 80% network utilization. In case of 90% network utilization, its guarantee ratio is not acceptable except for 4-node case in Figure 10. As shown in Figure 11, the guarantee ratio of the PBA can be lower than that of the RMS under 90% network utilization. Although the PBA algorithm provides low guarantee ratio under high network utilization, it takes shorter time to allocate the bandwidth than the SBA. Thus, the PBA can be used for online bandwidth allocation algorithm under the reasonable network utilization (e.g., below 80%).

5. Conclusion

This paper addresses the burst mode in allocating bandwidth for periodic real-time messages over the IEEE 802.12 network. As regards the two feasibility conditions, we calculate the worst case response times of periodic messages and allocate the bandwidth based on the lengths and deadlines of periodic messages. We propose two bandwidth allocation algorithms: the SBA algorithm for offline allocation and the PBA algorithm for online allocation and admission control. These algorithms ensure the timely transmission of real-time periodic messages. They do not require a run-time overhead such as keeping a timer or broadcasting a priority, because the bandwidth is represented by the number of high priority packets per grant. The experimental results show that the proposed schemes significantly improve the guarantee ratio, compared with the existing approaches. In this paper, aperiodic messages are assumed to be transmitted by the normal priority packets. However, this assumption may result in the long response times of aperiodic messages. The main reason is that the normal priority packets yield to the high priority packets and are transmitted in single packet mode. Further studies will be required to shorten the response times of aperiodic messages effectively.

References

Agrawal, G., B. Chen, W. Zhao, and S. Davari: 1992, 'Guaranteeing Synchronous Message Deadlines with Timed Token Protocol'. In: *Proceedings of International Conference on Distributed Computing Systems.* pp. 468–475.

Cruz, R. L.: 1991, 'A Calculus for Network Delay, Part I: Network Elements in Isolation'. *IEEE Transactions on Information Theory* **37**(1), 114–131.

Grinham, J. and M. Spratt: 1993, 'IEEE 802.12 Demand Priority and Multimedia'. In: *Proceedings of the 4th International Workshop on Network and Operating Systems Support for Digital Audio and Video.* pp. 75–86.

IEEE Std. 802.12: 1995, *Demand Priority Access Method, Physical Layer and Repeater Specification for 100 Mb/s Operation.* IEEE.

IEEE Std. 802.12c: 1998, *Supplement to ANSI/IEEE Std. 802.12.* IEEE.

IEEE Std. 802.3u: 1995, *Media Access Control(MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100Base-T (Clauses 21-30).* IEEE.

Kim, P.: 1998a, 'Deterministic Service Guarantees in 802.12 Networks, Part I: The Single-Hub Case'. *IEEE/ACM Transactions on Networking* **6**(5), 645–658.

Kim, P.: 1998b, 'Deterministic Service Guarantees in 802.12 Networks, Part II: The Cascaded Network Case'. In: *Proceedings of IEEE INFOCOM.* pp. 1376–1383.

Kim, P.: 1998c, 'Resource Reservation in Shared and Switched Demand Priority Local Area Networks'. Ph.D. thesis, Dept. of Computer Science, University of London.

Kim, T.: 2000, 'Guaranteed Service for Real-Time Message Streams in IEEE 802.12 based Distributed Systems'. Ph.D. thesis, School of Computer Science and Engineering, Seoul National University.

Kim, T., N. Kim, N. Chang, and H. Shin: 1998a, 'Bandwidth Allocation for Transmission of Real-Time Messages in Burst Mode over IEEE 802.12 Network'. In: *Proceedings of the 15th IFAC Workshop on Distributed Computer Control Systems.* pp. 19–24.

Kim, T., H. Shin, and N. Chang: 1998b, 'Scheduling Algorithm for Hard Real-Time Communication in the Demand Priority Network'. In: *Proceedings of the 10th Euromicro Workshop on Real Time Systems.* pp. 45–52.

Lehoczky, J. P., L. Sha, and Y. Ding: 1989, 'The Rate-Monotonic Scheduling Algorithm: Exact Characterization and Average Case Behavior'. In: *Proceedings of the Real-Time Systems Symposium.* pp. 166–171.

Liu, C. L. and J. W. Layland: 1973, 'Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment'. *Journal of the ACM* **20**(1), 46–61.

Martini, P. and J. Ottensmeyer: 1995, 'Real-Time Communication in the Demand-Priority LAN: The Effects on Normal Priority Traffic'. In: *Proceedings of 20th Conference on Local Computer Networks.* pp. 350–357.

Molle, M. and G. Watson: 1996, '100Base-T/IEEE 802.12/Packet Switching'. *IEEE Communications Magazine* **34**(8), 64–73.

Watson, G., A. Albrecht, J. Curcio, D. Dove, S. Goody, J. Grinham, M. Spratt, and P. Thaler: 1995, 'The Demand Priority MAC Protocol'. *IEEE Network* **9**(1), 28–34.