Hard Synchronous Real-Time Communications with the Timed-Token MAC Protocol.

Jun Wang
PhD
2009

This is a digitised version of a dissertation submitted to the University of Bedfordshire.

It is available to view only.

This item is subject to copyright.
Hard Synchronous Real-Time Communications with the Timed-Token MAC Protocol

by

JUN WANG

A thesis submitted for the degree of Doctor of Philosophy of the University of Bedfordshire

July 2009
Hard Synchronous Real-Time Communications with the Timed-Token MAC Protocol

JUN WANG

ABSTRACT

The timely delivery of inter-task real-time messages over a communication network is the key to successfully developing distributed real-time computer systems. These systems are rapidly developed and increasingly used in many areas such as automation industry. This work concentrates on the timed-token Medium Access Control (MAC) protocol, which is one of the most suitable candidates to support real-time communication due to its inherent timing property of bounded medium access time.

The support of real-time communication with the timed-token MAC protocol has been studied using a rigorous mathematical analysis. Specifically, to guarantee the deadlines of synchronous messages (real-time messages defined in the timed-token MAC protocol), a novel and practical approach is developed for allocating synchronous bandwidth to a general message set with the minimum deadline ($D_{\text{min}}$) larger than the Target Token Rotation Time ($TT_{\text{RT}}$). Synchronous bandwidth is defined as the maximum time for which a node can transmit its synchronous messages every time it receives the token. It is a sensitive parameter in the control of synchronous message transmission and must be properly allocated to individual nodes to guarantee deadlines of real-time messages.

Other issues related to the schedulability test, including the required buffer size and the Worst Case Achievable Utilisation (WCAU) of the proposed approach, are then discussed. Simulations and numerical examples demonstrate that this novel approach performs better than any previously published local synchronous bandwidth allocation (SBA) schemes, in terms of its ability to guarantee the real-time traffic. This work also examines the $TT_{\text{RT}}$, which is the other important parameter of the timed-token MAC protocol for supporting real-time traffic. A proper selection of the $TT_{\text{RT}}$, which can maximise the WCAU of the proposed SBA scheme, is addressed.

The work presented in this thesis is compatible with any network standard where timed-token MAC protocol is employed and therefore can be applied by engineers building real-time systems using these standards.
# Table of Contents

Abstract .......................... I

Table of Contents ........................ II

Acknowledgements ........................ IV

Publications ........................ VI

Declaration ........................ VII

Glossary of Terms ........................ VIII

List of Tables ........................ XI

1 Introduction ........................... 1
   1.1 Background ............................... 1
   1.2 Aim and objectives ........................ 4
   1.3 Methodologies ............................ 5
   1.4 Thesis Organisation ....................... 5

2 Literature Review ........................ 8
   2.1 Hard Real-Time Communication ............... 8
      2.1.1 Real-Time LAN Models .................. 9
      2.1.2 Hard Real-Time Messages ............... 10
      2.1.3 MAC Protocols ........................ 15
   2.2 Real-Time MAC Protocols ..................... 17
      2.2.1 Timed-Token MAC Protocols ............... 17
      2.2.2 Other MAC Protocols .................... 20
   2.3 The Use of the Timed-Token MAC Protocol .... 22
2.3.1 Broadcast Networks ........................................... 22
2.3.2 Wireless LAN Networks ........................................ 25
2.3.3 Home Networks ................................................... 28
2.3.4 Notes on Employing the Timed-Token MAC Protocol with Other Network Standards ........................................... 29
2.4 Summary ........................................................... 31

3 Preliminaries .......................................................... 32
  3.1 The Network and Message Models .................................. 32
    3.1.1 Network Models ............................................. 32
    3.1.2 Message Models ............................................. 33
  3.2 Timed-Token MAC Protocol ........................................ 33
  3.3 Synchronous Bandwidth Allocation Schemes ....................... 35
    3.3.1 Classification ............................................. 35
    3.3.2 Constraints ................................................. 36
    3.3.3 Performance Metric ....................................... 37
    3.3.4 Previously Published SBA Schemes ......................... 39
  3.4 Protocol Timing Properties ...................................... 42
  3.5 Summary ........................................................... 44

4 A Generalised Synchronous Bandwidth Allocation Scheme ................ 46
  4.1 Bandwidth Allocation to Message Streams with their Deadlines less than $2 \cdot TTRT$ 47
    4.1.1 Message Streams with $D_i \leq P_i$ ............................ 48
    4.1.2 Message Streams with $TTRT \leq P_i < D_i$ ..................... 48
    4.1.3 Message Streams with $0 < P_i < TTRT < D_i$ .................. 50
    4.1.4 Summary ....................................................... 55
  4.2 Bandwidth Allocation to Message Streams with their Deadlines no less than $2 \cdot TTRT$ ........................................... 55
    4.2.1 Message Streams with Deadlines no larger than Periods ........ 55
    4.2.2 Message Streams with Deadlines larger than Periods ........... 56
    4.2.3 Summary ....................................................... 72
  4.3 A New Generalised SBA Scheme ................................... 72
    4.3.1 A Few Notes on the New Local SBA Scheme .................... 73
    4.3.2 Feasibility Analysis ......................................... 75
    4.3.3 Comparison with Some Other Local SBA Schemes ................ 76
  4.4 Summary ........................................................... 78

III
5 The Worst Case Achievable Utilisation

5.1 An Achievable Utilisation of the GLA Scheme
5.2 The WCAU of the GLA Scheme
5.3 Selection of TTRT
5.4 Summary

6 Simulation and Numerical Examples

6.1 Guaranteeing Synchronous Message Sets
   6.1.1 Synchronous Message Sets with TTRT < D_{min} < 2 \cdot TTRT
   6.1.2 Synchronous Message Sets with D_{min} \geq 2 \cdot TTRT
6.2 The Worst Case Achievable Utilisation
6.3 Summary

7 Conclusions

7.1 Contributions
7.2 Future Work

References

A Proof of Theorem 5
B Proof of Theorem 6
C Deducing H_i from Eq. (4.23)
D Deducing H_i from Eq. (4.32)
E Proof of Theorem 8
F Proof of Theorem 9
G Proof of Theorem 10
H Deducing \Delta in Eq. (4.42)
I Proof of Theorem 12
J Proof of Lemma 4
K Matlab code for simulation
Acknowledgements

I would like to express profound gratitude to my supervisors, Dr. Sijing Zhang and Professor Carsten Maple, for their invaluable encouragement, supervision and useful suggestions throughout this research work, and their careful reading and constructive suggestions on my writing during the review of the work presented in this thesis. I am also highly thankful to my other supervisor, Dr. Enjie Liu, for her valuable suggestions throughout the whole research process.

I would also like to thank all members from the Institute for Research in Applicable Computing at the University of Bedfordshire, especially Hui Song, Peng Wang, Peter Norrington and Jon Hitchcock, for their help and friendship in both research and life.

I am as ever, especially indebted to my parents, for their love, care and moral support throughout my life.

Finally, I owe my thanks to my wife, JIA NI, who took care of my health and endured my complaints about the stress during the thesis writing. It is her constant support and patience that made this thesis possible.
Publications


Declaration

I declare that this thesis is my own unaided work. It is being submitted for the degree of Doctor of Philosophy at the University of Bedfordshire.

It has not been submitted before for any degree or examination in any other University.

Name of candidate: ___________________________ Signature: ___________________________

Date: ___________________________
## Glossary of Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_i$</td>
<td>The maximum amount of time needed to transmit a message from stream $S_i$.</td>
</tr>
<tr>
<td>$D_i$</td>
<td>The relative deadline of messages from $S_i$. That is, if a message from $S_i$ arrives at time $t$, it must be sent before $t + D_i$.</td>
</tr>
<tr>
<td>$D_{min}$</td>
<td>Tightest lower bound on message deadlines, i.e., the minimum of all $D_i$ ($1 \leq i \leq n$).</td>
</tr>
<tr>
<td>$f_L$</td>
<td>A local SBA scheme.</td>
</tr>
<tr>
<td>$f_G$</td>
<td>A global SBA scheme.</td>
</tr>
<tr>
<td><strong>Get</strong>$_H$</td>
<td>An algorithm used to get $H_i$ for $S_i$ with $P_i &lt; TTRT &lt; D_i &lt; 2 \cdot TTRT$.</td>
</tr>
<tr>
<td>GLA</td>
<td>Generalised Local Allocation.</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Synchronous bandwidth allocated to node $i$, i.e., the maximum amount of time for which the node $i$ is allowed to transmit its synchronous messages each time it receives the token.</td>
</tr>
<tr>
<td>$H_{\text{SCHEME}}^i$</td>
<td>The $H_i$ produced by a local allocation scheme called $\text{SCHEME}$.</td>
</tr>
<tr>
<td>$\vec{H}$</td>
<td>Allocation vector, i.e., $\vec{H} = (H_1, H_2, \ldots, H_n)$.</td>
</tr>
<tr>
<td>$I(v)$</td>
<td>The tight upper bound on the (maximum) time which could possibly elapse in the worst case before a node uses up its next $v$ (where $v$ is an integer no less than one) allocated synchronous bandwidths ($H_i$).</td>
</tr>
<tr>
<td>IGLA</td>
<td>Improved Generalised Local Allocation.</td>
</tr>
<tr>
<td>$LC_i$</td>
<td>Late Counter of node $i$. Normally, $0 \leq LC_i \leq 1$.</td>
</tr>
<tr>
<td>$M$</td>
<td>A synchronous message set.</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of nodes on the token ring network.</td>
</tr>
<tr>
<td>$P_i$</td>
<td>The inter arrival period between any two consecutive messages from stream $S_i$.</td>
</tr>
<tr>
<td>$P$</td>
<td>The percentage of the number of infeasible synchronous message sets to the total number of tested synchronous message set.</td>
</tr>
</tbody>
</table>
\[ q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \]
\[ q_{\text{min}} = \left\lfloor \frac{D_{\text{min}}}{TTRT} \right\rfloor \]
\[ r_{D_i} = D_i - q_i \cdot TTRT \]
\[ r_{P_i} = P_i - \left\lfloor \frac{P_i}{TTRT} \right\rfloor \cdot TTRT \]
\[ S_i \] Synchronous message stream at node \( i \).
\[ \text{SBA} \] Synchronous Bandwidth Allocation.
\[ s_{i,j} \] The time that the transmission of the \( j \)th message in stream \( S_i \) is completed.
\[ t_{i,j} \] The arrival time of the \( j \)th message of stream \( S_i \) residing on the node \( i \).
\[ t_{l,i} \] The token’s \( l \)th arrival time at node \( i \).
\[ TTRT \] Target Token Rotation Time.
\[ TRT_i \] Token Rotation Timer of node \( i \). \( 0 \leq TRT_i \leq TTRT \).
\[ TRT'_i \] The amount of time left before node \( i \) initiates the ring recovery process, \[ TRT'_i = TRT_i + (1 - LC_i) \cdot TTRT. \]
\[ THT_i \] Token Holding Timer of node \( i \).
\[ U(M) \] Utilisation factor of the synchronous message set \( M \).
\[ U \] Achievable utilisation of a SBA scheme.
\[ U^* \] The least upper bound of achievable utilisations of a SBA scheme, i.e., WCAU of a SBA scheme.
\[ U_e(M) \] Effective utilisation factor of the synchronous message set \( M \).
\[ U_e \] Effective achievable utilisation of a SBA scheme.
\[ U_e^* \] The least upper bound of effective achievable utilisations of a SBA scheme.
\[ U_i \] \[ U_i = \frac{C_i}{\min(P_i, D_i)} \]
\[ \text{WCAU} \] Worst Case Achievable Utilisation.
\[ X^k_i(H) \] The minimum amount of available synchronous transmitting time in time interval \( (k-1) \cdot P_i + D_i \), based on the global information of the network.
\[ x^k_i \] The minimum amount of available synchronous transmitting time in time interval \( (k-1) \cdot P_i + D_i \), based on the node \( i \)’s local information.
\[ X_i(T) \] The minimum available time for node \( i \) to transmit its synchronous messages during the time interval \( T \).
\[ x_i(T) \] The minimum available synchronous message transmitting time for node \( i \) during \( T \) based only on node \( i \)’s local information.
\[ \tau \] Portion of \( TTRT \) unavailable for transmitting messages.
\[ \alpha \] Ratio of \( \tau \) to the Target Token Rotation Time \( (TTRT) \), i.e., \( \alpha = \frac{\tau}{TTRT} \).
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta$</td>
<td>The difference between the actual available transmitting time and the estimated transmitting time during a given time interval.</td>
</tr>
<tr>
<td>$\delta_1$</td>
<td>The difference between the actual number of messages and the average number of messages arriving at node $i$ during a given time interval.</td>
</tr>
<tr>
<td>$\delta_2$</td>
<td>The difference between the actually required synchronous bandwidth and the actually allocated synchronous bandwidth for node $i$.</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Components of application-to-application delay ........................................... 13
3.1 Six published local SBA schemes ................................................................. 41
3.2 The basic categories of SBA schemes ............................................................. 42
6.1 Synchronous message set A guaranteed by GLA, IGLA, NLA and FLA (TTRT =
10.0 and \(\tau = 0.0\)) ....................................................................................... 103
6.2 Synchronous message set B guaranteed by GLA and IGLA (TTRT = 10.0 and
\(\tau = 0.0\)) ........................................................................................................ 104
6.3 Synchronous message set C guaranteed by GLA((TTRT_i + 1)C_i) (TTRT = 10.0
and \(\tau = 0.0\)) ................................................................................................. 105
6.4 Synchronous message set D guaranteed by GLA(Get_H) (TTRT = 10.0 and
\(\tau = 0.0\)) ........................................................................................................ 106
6.5 Synchronous message set E with \(P_i = D_i \geq 2 \cdot TTRT\) guaranteed by all the local
SBA schemes (TTRT = 10.0 and \(\tau = 0.0\)) ....................................................... 107
6.6 Synchronous message set F with \(P_i = D_i \geq 2 \cdot TTRT\) guaranteed by LA_A/LA_M/LA_Z
and GLA/IGLA/NLA (TTRT = 10.0 and \(\tau = 0.0\)) ................................................ 109
6.7 The other example synchronous message set G with \(P_i = D_i \geq 2 \cdot TTRT\) guar-
anteed by LA_A/LA_M/LA_Z and GLA/IGLA/NLA (TTRT = 10.0 and \(\tau = 0.0\)) 110
6.8 Example synchronous message set H with \(P_i = D_i \geq 2 \cdot TTRT\) unscheduled by
LA_A/LA_M/LA_Z but guaranteed by PA (TTRT = 10.0 and \(\tau = 0.0\)) .......... 110
6.9 Synchronous message set I with \(P_i = D_i \geq 2 \cdot TTRT\) only guaranteed by
GLA/IGLA/NLA (TTRT = 10.0 and \(\tau = 0.0\)) .................................................... 111
6.10 The other example synchronous message set J with \(P_i = D_i \geq 2 \cdot TTRT\) only
guaranteed by GLA/IGLA/NLA (TTRT = 10.0 and \(\tau = 0.0\)) ......................... 111
6.11 Example synchronous message set K unscheduled by any local SBA scheme

\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \) .................................................. 112

6.12 Synchronous message set L with \( D_i \geq 2 \cdot \text{TTRT} \) guaranteed by \( LA_M, LA_Z, GLA \) and \( IGLA \) (\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \)) ............................................. 112

6.13 Synchronous message set M with \( D_i \geq 2 \cdot \text{TTRT} \) unscheduled by \( LA_Z \) (\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \)) .................................................. 113

6.14 The other example synchronous message set N with \( D_i \geq 2 \cdot \text{TTRT} \) unscheduled by \( LA_Z \) (\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \)) ............................................. 113

6.15 Synchronous message set O with \( D_i \geq 2 \cdot \text{TTRT} \) guaranteed by \( LA_Z \) and \( IGLA \)

\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \) .................................................. 114

6.16 Synchronous message set P with \( D_i \geq 2 \cdot \text{TTRT} \) guaranteed by \( GLA \) and \( IGLA \)

\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \) .................................................. 114

6.17 Synchronous message set Q with \( D_i \geq 2 \cdot \text{TTRT} \) only guaranteed by \( IGLA \)

\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \) .................................................. 115

6.18 Example synchronous message set R with \( D_i \geq 2 \cdot \text{TTRT} \) unscheduled by any local SBA scheme (\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \)) ............................................. 115

6.19 Synchronous message set S with \( D_i > \text{TTRT} \) guaranteed by \( GLA \) and \( IGLA \)

\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \) .................................................. 116

6.20 Example synchronous message set T with \( D_i > \text{TTRT} \) unscheduled by any local SBA scheme (\( \text{TTRT} = 10.0 \) and \( \tau = 0.0 \)) ............................................. 116

F.1 Sample Synchronous Message Set .............................................. 174
Chapter 1

Introduction

1.1 Background

A real-time system is defined as a system whose correctness depends not only on the logical results of computation, but also on the time at which the results are produced [71]. Most of these systems are also distributed as the applications themselves are often physically distributed. In such systems, communications through messages exchange between tasks residing on different nodes are necessary for completion of these distributed applications.

Distributed real-time systems may be categorised as soft real-time systems and hard real-time systems [4]. In soft real-time systems, although tasks and message transmissions have time requirements, the system can still function well even if some tasks cannot be finished before the time constraint, or some message transmissions fail. However, in a hard-real time system, every task and every message transmission must be finished before its deadline, correctly. The key to success in such hard real-time systems is the timely execution of data-processing tasks that usually reside on different nodes and communicate with one another to accomplish a common goal [57].

A distributed computer system can be viewed as consisting of three major layers: application processes, a distributed operating system, and a communication subsystem. In this three part structure:

- The application processes use the services provided by the distributed operating system and the communication subsystem.
- The distributed operating system unifies and integrates the control of distributed components, and provides a uniform high-level system interface to the application processes.
- The communication subsystem (hardware and software) facilitates the exchange of messages among the distributed components.

In a hard real-time distributed system, all these three layers should be time constrained, especially the communication layer which offers the fundamental functioning of the whole system and therefore whose speed, throughput, reliability and fault tolerance are limiting factors in the performance of the entire distributed system. Since a real-time local area network can potentially handle traffic generated by real-time systems, it can be used as a communication layer. In a hard real-time distributed system, in order to guarantee the connection of the local area network and the end-to-end deadline constraint of message transmission along the network, the use of proper MAC communication protocols should be considered carefully.

Since message transmission between different nodes of a distributed system often needs to share a single network channel resource, the use of controlled access protocols to schedule this resource sharing is important. Due to the particular difficulties inherent in the problem, a very limited number of protocols have been proposed for hard real-time communications. The timed-token MAC protocol, originally proposed by Grow [25], is suitable for hard real-time communication because of its inherent timing property of bounded medium access time, which is the basic requirement to ensure the end-to-end deadline constraint of message exchange. This research concentrates on issues related to deadline guarantees of hard real-time traffic using the timed-token MAC protocol.

With the timed-token MAC protocol, messages are categorised into two types: synchronous messages and asynchronous messages. Synchronous messages, which are periodic and arrive at regular intervals, such as voice and video streams, have a hard real-time delivery constraint and thus must be sent before their deadlines. In contrast, asynchronous messages, which are non-periodic and have no delivery time constraints, can be transmitted on a best-effort basis. Synchronous traffic is assigned a guaranteed bandwidth while the leftover bandwidth (unallocated, unused or both) is dynamically shared among all the nodes for asynchronous traffic [26]. This research focuses on guaranteeing the deadlines of synchronous messages.

For a network with \( n \) nodes, at network initialisation time, all nodes negotiate a common value for the Target Token Rotation Time (TTTRT), which is an important protocol parameter giving the expected token rotation time. Each node \( i \) \( (i = 1, 2, 3, \ldots n) \) is then assigned a fraction of the TTTRT, known as its synchronous bandwidth \( H_i \), which is the maximum time that the node is allowed to transmit its synchronous messages every time it receives the token. It can then transmit its asynchronous messages if the token has rotated sufficiently fast so that it arrived earlier than expected since the token’s last arrival at the same node.
The timed-token protocol guarantees, to each node, an average channel bandwidth and a bounded channel access delay for synchronous traffic. However, this guarantee alone, although necessary, is not sufficient for the timely delivery of deadline constrained messages. To guarantee the deadline constraint of synchronous messages, network parameters, such as $H$, $TTRT$ and buffer size, should be chosen carefully. More specifically,

- The synchronous bandwidth is the most important network parameter. However, the timed-token MAC protocol does not specify any methods to this allocation. If the allocated synchronous bandwidth is too small, the node might not have enough network access time to transmit messages before their deadlines. In contrast, large synchronous bandwidth will result in a long token rotation time, which may also cause message deadlines to be missed. A study on how synchronous bandwidth can be carefully allocated in the development of a proposed new SBA scheme is given in Chapter 4.

- Each node has buffers for outgoing and incoming synchronous messages. The buffer size reflects the number of synchronous messages waiting in the queue. In practice, because the number of waiting messages must be upper bounded, the buffer must have a limited size. The necessary required buffer size for the new solutions will be discussed in Chapter 4.

- $TTRT$ should be chosen small enough to meet responsiveness requirements of all nodes, i.e., $TTRT$ should make the token rotate fast enough to satisfy the most stringent response-time requirements of all nodes. However, too small a value of $TTRT$ will result in less available network bandwidth utilisation and will limit network capacity. A study on selecting a proper $TTRT$ for the Synchronous Bandwidth Allocation (SBA) scheme proposed in this research is presented in Chapter 5.

Much work on a proper selection of these parameters has been done, especially on the synchronous bandwidth allocation (SBA) scheme [2, 3, 12, 29, 56, 90, 92, 94, 96], in order to guarantee the transmission of synchronous messages. A detailed review of these SBA schemes can be found in Section 3.3.4. A guaranteed message will always be transmitted before its deadline as long as the network operates normally. Although extensive work has been done on a proper allocation of synchronous bandwidths, it is still valuable to investigate such allocation since it is the most sensitive parameter for guaranteeing hard real-time traffic in a timed-token network. Also, there remain areas for enhancement in terms of guaranteeing capability. The main focus of the research is to develop a practically efficient SBA scheme (which can perform better than any previously reported) and to study some other related issues.

The timed-token MAC protocol has been directly incorporated into many network standards including the Fibre Distributed Data Interface (FDDI) [39], IEEE 802.4 [38], the High-Speed
Data Bus (HSDB), the High-Speed Ring Bus (HSRB) and the Survivable Adaptable Fibre Optic Embedded Network (SAFENET) [65], which have been used by many embedded real-time applications as backbone networks. The idea of this protocol has also been widely used in many other areas to support real-time communications. For example, the Wireless Token Ring Protocol (WTRP) [20], which is used as a distributed MAC protocol for ad-hoc networks, uses the token to control the bounded medium access time. It can also be built within a new layer on top of CSMA/CD to enhance the Ethernet to comply with the real-time requirements in the industrial context [18]. A detailed review on the usage of the timed-token MAC protocol and its basic token ring idea can be found in Section 2.3.

1.2 Aim and objectives

This research aims to develop an effective and efficient approach to guarantee the timely transmission of hard real-time traffic in a timed-token ring network. The objectives of this research are:

1. to develop a new SBA scheme which can
   - be used for guaranteeing a general set of synchronous messages with arbitrary deadline constraints (i.e., a message’s deadline could be larger than, equal to or smaller than its period) and legitimately short deadlines (i.e., the minimum deadline $D_{\text{min}}$ could be any small but larger than $TTRT$. That is, $D_{\text{min}} > TTRT$).
   - generally perform better than any previously published local SBA scheme.

2. to analyse the feasibility when using the proposed SBA scheme to allocate synchronous bandwidth to a synchronous message set.

3. to evaluate the Worst Case Achievable Utilisation (WCAU) of the proposed SBA scheme.

4. to assess the impacts of $TTRT$ on the guarantee rate of real-time traffic.

---

1 Although the timed-token MAC protocol was proposed for use in a ring topology network, its usage is not restricted in the physical ring topology. It can be used for other topology network which uses a token-passing mechanism, e.g., a token bus network. In this research, the ring topology is a logical ring topology which refers to the token-passing mechanism but not the real physical topology of a network. Thus, the result is not restricted to the ring topology network.
1.3 Methodologies

In hard real-time communication, it must be ensured that every individual message is sent before its deadline. Thus, formal and rigorous mathematical analysis, which is important and necessary for the problems investigated in this research, was adopted as the main methodology. Specifically,

- The symbols and symbolic notations are properly selected.
- Some reasonable assumptions are made.
- The rigorous notion of proof is used to identify the correctness of the results.

The main operations used here are floor and ceiling operations. The floor operation maps a real number to the next smallest integer. That is, \( \lfloor x \rfloor \) is the largest integer not greater than \( x \). The ceiling operation maps a real number to the next largest integer. That is, \( \lceil x \rceil \) is the smallest integer not less than \( x \). These two operations make the problem discontinuous.

Simulations and numerical examples are used to demonstrate the improved performance of the proposed solution and assess the performance gains. Specifically,

- the simulations are used to give a clear and comprehensive view of overall performance of the results.
- some typical numerical examples are used to demonstrate how to use the results in real systems and the superiority of these new results.

1.4 Thesis Organisation

The thesis is organised into seven chapters. In this chapter, the research field was first outlined in Section 1.1 where the topic and the motivation for further studies on proper synchronous bandwidth allocation within the timed-token MAC protocol to guarantee timely delivery of synchronous messages were presented. Section 1.2 stated the aims and objectives of this study. The main methodologies used in this research were briefly introduced in Section 1.3. The rest of this thesis is organised as follows:

In Chapter 2, an overview of the area of real-time communication is given, with the focus on the characteristics of hard real-time traffic and the MAC protocols which support the transmission of real-time traffic. The survey also covers the usage of the timed-token MAC protocol with different network standards in different areas.
In Chapter 3, the preliminaries under which the research is conducted are discussed. Specifically, first, the network and message models are introduced. These models are commonly used by many researchers in their related studies. An introduction to the timed-token protocol is then presented. Next, an introduction to SBA schemes is given, which includes the definition of SBA schemes, their classification (i.e., local and global SBA schemes), three constraints (i.e., protocol constraint, deadline constraint and buffer constraint) that have to be met for any schedulable synchronous message set, and a commonly used performance metric of the Worst Case Achievable Utilisation (WCAU). Some previously published local SBA schemes are also listed here for the purpose of comparison with the proposed SBA scheme in Chapter 6. Finally, the timing properties of the timed-token protocol are introduced. These properties are necessary for guaranteeing synchronous message deadlines and important for further research on the synchronous message transmission in a timed-token ring network. Several important results of timing properties are provided based on the time-line, as well as the comparison among these results.

In Chapter 4, an efficient and practical local SBA scheme is developed. The proposed new local scheme performs better than any of the previously published local SBA scheme in terms of guaranteeing synchronous message sets due to the fact that it calculates synchronous bandwidths based on the best results of the protocol timing properties, with locally available information. This new local scheme can be used for any synchronous message set with arbitrary deadline constraints. It also takes into account the synchronous bandwidth allocation for message sets with \( TTRT < D_{\text{min}} < 2 \cdot TTRT \). As a result, this local SBA scheme can be applied in theory to any synchronous message set. Before finishing this chapter, the feasibility of allocations produced by the proposed local scheme, a possible improvement of this scheme, and the comparison with some previously published local SBA schemes are also discussed to show the superiority of this new local SBA scheme.

In Chapter 5, the WCAU, which is a metric commonly used to evaluate and compare the performance of the different SBA schemes in the time-token network, is discussed. It is shown that the WCAU of the new proposed local SBA scheme is no worse than that of any other reported local SBA scheme. The result also implies that the new proposed local SBA scheme performs no worse than any of the previously published local SBA schemes. An important parameter of the timed-token MAC protocol, \( TTRT \), is also studied in this chapter. A proper selection of the \( TTRT \) is then suggested to achieve the highest possible WCAU of the proposed local SBA scheme.

In Chapter 6, the new proposed local SBA scheme is compared with all the previously published local SBA schemes via simulation and numerical examples, from the perspectives of
guaranteeing synchronous message sets and the worst case achievable utilisation. The results from this comparison show that the performance of the new local SBA scheme is superior to all other previously published local SBA schemes.

In Chapter 7, the aims of this research are re-examined. The main results achieved and major contributions of this work are then summarised. Finally, some interesting topics for future research are discussed.

Appendices A to J give the proofs to theorems/lemmas and deduction of some complex equations developed in this research.
Chapter 2

Literature Review

There is much work that addresses the problems related to hard real-time communication in a distributed real-time system. Some main results and developments, with focus on the timed-token MAC protocol, will be reviewed in this chapter.

This chapter is organised as follows: Section 2.1 reviews some basic concepts and requirements in the area of hard real-time communication, including LAN models, hard real-time messages and MAC protocols. In Section 2.2, some of the main real-time MAC protocols are discussed, with the focus on the timed-token MAC protocol. Finally, Section 2.3 presents the applications of the timed-token MAC protocol.

2.1 Hard Real-Time Communication

The correctness of a time-critical distributed system depends on not only the logical results but also the time at which those results appear. Usually, a local area multiple-access network which can support the timely delivery of inter-task messages is used to ensure timely computational results. Local area multiple-access networks typically have a geographical expanse covering a building or a group of buildings, or a single ship or aircraft. They are one of the most common types of networks used to support distributed hard real-time applications.

To meet the time constraints of hard real-time messages, these messages must be properly scheduled for transmission. Scheduling messages in a multiple-access network is the function of the MAC protocol. MAC protocols have a significant impact on the real-time performance of a network. However, because of the nature of hard real-time communication, new concerns have to be considered. While traditional studies on MAC protocols typically focus on issues such as maximising the message throughput or minimising the average message delay [70, 74], a MAC
protocol for hard real-time message traffic must consider the timing constraints of individual messages. The most important design consideration is to ensure that message deadlines are met or that the number of missed deadlines is minimised.

This section first examines the LAN models used for local area multiple-access networks which can be used for hard real-time distributed systems, and then hard real-time messages in these system is studied. The functions and properties of MAC protocols to support real-time traffic are also discussed here.

2.1.1 Real-Time LAN Models

Nodes in a multiple-access network usually have a layered communication architecture. The left part of Fig. 2.1 shows the layered architecture of the ISO-OSI reference Model [97]. Each layer has a different set of protocols responsible for carrying out the functions required of the layer. For example, the protocols for the physical layer are responsible for the management of physical connections and for transmission of bits over the transmission medium. The protocols in the data link layer are responsible for providing reliable transmission of data frames across a communication link [54]. The data link layer is composed of two sub-layers: the Logical Link Control (LLC) sub-layer and the Medium Access Control (MAC) sub-layer. The LLC sub-layer provides communication services to the layer above it by implementing medium-independent data link functions, such as connection management, error handling, flow control and packetisation (or fragmentation), and has the overall responsibility for the exchange of data between nodes. A MAC protocol is responsible for selecting and sending messages over the shared channel of a multiple-access network. It is discussed later in this chapter that a proper designed MAC protocol is important to hard real-time communication.

IEEE 802 proposed a simple architecture that addresses the physical and data link layers in the ISO-OSI reference model. Beus-Dukic et al [7] suggested that any candidate architecture for time-critical communication should adhere to the ISO-OSI seven layers model. However, the ISO-OSI model is too general and too complex to meet hard real-time constraints [62]. The performance is simply functional without any requirement to be on time [45]. A simpler but time considered LAN architecture is more suitable for real-time communication. For example, the RTLANT (Real-Time Local Area Network) architecture, proposed by Arvind, Ramamritham and Stankovic [5] and depicted in the right part of Fig. 2.1, is a simpler four-layer LAN architecture for communications in distributed real-time system. However, additional functions are added to permit the communication timing requirements of applications to be specified dynamically and offers mechanisms to guarantee these requirements, if needed and if at all possible. In these
simplified LAN architectures, the MAC layer is still retained due to its important impact on scheduling the channel access and guaranteeing hard real-time communication.

### 2.1.2 Hard Real-Time Messages

**Message delays**

In a distributed hard real-time system, the application-to-application delay, which is defined as the time delay experienced by a message that is sent between application tasks residing on different nodes, is crucial in determining whether application deadlines will be satisfied.

The application-to-application delay relies on the implementation of the system. For the majority of multiple-access networks, protocols in layers above the MAC protocol are realised in the host processor where applications are executed, while the MAC is implemented by special network hardware. With this type of system implementation, a decomposition of the application-to-application delay (See Fig. 2.2) experienced by a message is given as follows:

- **Processing and queuing delay in the upper layers of the sending node**: The processing delay includes the time required to create message headers. The queuing delay includes both the time during which a message is waiting to be processed, and the time during which a message is waiting to be passed to lower layer protocols. Let $d_{up\_send}(M)$ denote the maximum delay that may be experienced by message $M$ in the upper layers of the sending node.
• **Message queuing delay in the MAC layer of the sending node**: This delay occurs when a message is waiting to be sent by the MAC protocol. It is the time between arrival of a message at the MAC transmission queue of the sending node and the completion of the transmission of all messages (if any) queuing in front of this message. Let $d_{MAC_{\text{send}}}(M)$ denote the maximum queueing delay that message $M$ may experience in the MAC layer of the sending node.

• **Channel access delay**: This delay is the time a head-of-queue message may experience in the MAC layer of the sending node before being transmitted over the channel, i.e., the time between the end of the previous transmission and the beginning of a new transmission. Let $d_{channel_{\text{access}}}(M)$ denote the maximum channel access delay that message $M$ may experience before being transmitted.

• **Message transmission delay**: This delay is the time required to physically transmit the message. Let $C_M$ be the transmission delay for message $M$.

• **Propagation delay**: This delay is the time required for a single bit to travel through the channel to the receiving node. Let this delay be denoted by $\tau$.

• **Message queuing delay in the MAC layer of the receiving node**: Once a message is received at its destination, it is stored in a queue at the MAC layer of the receiving node until
the message can be passed to the upper layer protocols. Let $d_{\text{MAC}_{\text{receive}}}(M)$ denote the maximum delay that message $M$ may experience in the MAC layer of the receiving node.

- **Processing and queuing delay in the upper layers of the receiving node**: After a message is passed to the upper layer protocols, it may again be queued, before the message header is removed and the data passed to the receiving application task. Let $d_{\text{up}_{\text{receive}}}(M)$ denote the maximum delay that may be experienced by message $M$ in the upper layers of the receiving node.

An upper bound on the application-to-application delay can be determined once the maximum delay of every component of the application-to-application delay can be determined (or estimated). Let $d_{\text{app}}(M)$ be such an upper bound on the application-to-application delay for message $M$, such that

$$d_{\text{app}}(M) = d_{\text{up}_{\text{send}}}(M) + d_{\text{MAC}_{\text{send}}}(M) + d_{\text{channel}_{\text{access}}}(M) + C_M + \tau$$

$$+ d_{\text{MAC}_{\text{receive}}}(M) + d_{\text{up}_{\text{receive}}}(M)$$

Several of the components of the application-to-application delay are not determined by the network but by the hosts. The processing and queueing delays in the upper layer protocols ($d_{\text{up}_{\text{send}}}(M)$ and $d_{\text{up}_{\text{receive}}}(M)$) depend mainly on the system software and the processor/memory speeds at the sending and receiving hosts. The queueing delay in the MAC layer of the receiving node ($d_{\text{MAC}_{\text{receive}}}(M)$) depends on how quickly a host processor can respond to a “message arrive” interrupt from the MAC layer. For a given system, an upper bound on these delays can often be determined [46].

Both of the message transmission delay ($C_M$) and the message propagation delay ($\tau$) can be regarded as fixed because they can be determined respectively by the allocated bandwidth of the channel and by the signal propagation speed. But, the queuing delay ($d_{\text{MAC}_{\text{send}}}(M)$) and the channel access delay ($d_{\text{channel}_{\text{access}}}(M)$) in the MAC layer of the sending node are a function of the specific MAC protocol used, as well as the local message scheduling policy adopted. An appropriately chosen MAC protocol can reduce the delay of messages with urgent time constraints (possibly resulting in larger delays for other messages) and thus help real-time messages to meet their deadlines.

Table 2.1 summarises the above discussion.

**Message classification**

In a distributed hard real-time system, hard real-time messages can be classified into two categories: *synchronous (periodic)* messages and *asynchronous (aperiodic)* messages.
Table 2.1: Components of application-to-application delay

<table>
<thead>
<tr>
<th>delay component</th>
<th>depends on</th>
<th>can be determined?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_{\text{up,send}}(M)$</td>
<td>system software and processor/memory speeds</td>
<td>Yes</td>
</tr>
<tr>
<td>$d_{\text{MAC,send}}(M)$</td>
<td>MAC protocol and message scheduling policy</td>
<td>No</td>
</tr>
<tr>
<td>$d_{\text{channel,access}}(M)$</td>
<td>MAC protocol and message scheduling policy</td>
<td>No</td>
</tr>
<tr>
<td>$C_M$</td>
<td>bandwidth of channel</td>
<td>Yes</td>
</tr>
<tr>
<td>$\tau$</td>
<td>signal propagation speed</td>
<td>Yes</td>
</tr>
<tr>
<td>$d_{\text{MAC,receive}}(M)$</td>
<td>interrupt response speed of processor</td>
<td>Yes</td>
</tr>
<tr>
<td>$d_{\text{up,receive}}(M)$</td>
<td>system software and processor/memory speeds</td>
<td>Yes</td>
</tr>
</tbody>
</table>

• synchronous messages: synchronous messages are used for transmitting sensory data or for inter-task communication between periodic tasks. A synchronous message stream consists of a sequence of messages with constant inter-arrival times. Messages in a synchronous message stream can often be characterised by:

  – An **arrival time**, which is the time at which the message becomes available for transmission.
  
  – A **length**, which is the number of time units required to transmit the entire message.
  
  – A **deadline**, which is the maximum time interval during which the transmission of the message must finish after the message’s arrival. Deadline defines a time constraint for the transmission of a synchronous message.

As discussed in Section 2.1.2, a message $M$ experiences several delays en route to its destination. Satisfying a message deadline is directly related to these delays. There are several different deadline constraints depending on which layer these constraints are defined.

Normally, a time constraint refers to the delays which a message may experience in the MAC layer of the sending node, i.e., the MAC layer deadline, denoted as $D_M$. Guaranteeing the MAC layer deadline constraint implies

$$D_M \geq d_{\text{MAC,send}}(M) + d_{\text{channel,access}}(M) + C_M$$

The sum of $d_{\text{MAC,send}}(M)$, $d_{\text{channel,access}}(M)$ and $C_M$ in the above inequality is defined as the response time of message $M$, which is the time between the arrival of a message at the MAC layer transmission queue (from upper layers of the sending node) and the completion of its transmission.
In a real system, message time constraints may also be given as application layer deadlines. The application layer deadline of a message $M$, denoted as $D_{\text{Mapp}}$, gives the maximum amount of time that may elapse before the message must be received by the application task to which it is sent. $D_{\text{Mapp}}$ can be expressed as

$$D_{\text{Mapp}} = D_M + d_{\text{up-send}}(M) + \tau + d_{\text{MAC-receive}}(M) + d_{\text{up-receive}}(M)$$

The above expression explores the relationship between deadlines defined in the MAC layer and the application layer. As discussed in Section 2.1.2, the upper bounds of all delays, except the queuing delay in the MAC layer of the sending node and the channel access delay, can be determined. Thus the MAC protocols, which show the significant impact on these two delays, should be chosen carefully to ensure the satisfaction of the time constraint no matter how the time constraint is defined.

Fig. 2.3 depicts the relationship between message characteristics and delays relating to the response time.

![Message characteristics and delays relating to the response time](image)

Figure 2.3: Message characteristics and delays relating to the response time

- asynchronous messages: asynchronous messages are often used to carry alert (alarm) information or for communication between aperiodic tasks. Compared to synchronous messages, asynchronous messages have stochastic inter-arrival times. Prior to their arrival the system may have little knowledge of their timing characteristics. However, like synchronous messages, asynchronous messages may be associated with an arrival time, a length and a deadline. Another time constraint that is often used for an asynchronous message is the laxity of that message [58]. The laxity of a message is defined as the amount of time remaining before transmission of the message must begin if the message is to meet its deadline.
In general, two strategies, the **guarantee strategy** and the **best effort strategy**, can be used in developing time constrained communication protocols based on the type of messages (synchronous messages or asynchronous messages) that the protocols are developed to support. In the guarantee strategy, an attempt is made to guarantee ahead of transmission time that the real-time messages will meet their deadlines. In the best effort strategy, the network will try to meet the message deadlines, but no guarantees are given. The systems where the best effort strategy is adopted can normally tolerate a certain amount of message loss.

This research focuses on guaranteeing synchronous message transmission during error free operation of the network via appropriate selection of network parameters.

### 2.1.3 MAC Protocols

As discussed earlier in this chapter, a proper designed MAC protocol is important to support hard real-time transmission over a local area multiple-access network. It is responsible for scheduling message transmissions over a shared channel to meet the time constraint. This section first introduces the functions of the MAC protocols, followed by a discussion on supporting hard real-time transmissions.

**Functions of MAC protocols**

The MAC Protocol performs two main functions: an access arbitration process and a transmission control process. The access arbitration process determines which node (with messages to send) is granted the right to use the channel, while the transmission control process determines how long a node can continue to send its messages once granted the transmission right.

It is important to choose a proper access arbitration process because, firstly, it affects issues such as fairness, and secondly, it influences metrics such as the throughput and the message delay. Further, for real-time communication, there are additional points to be addressed:

- **Timing correctness**: In order to satisfy the time constraints of hard real-time messages, the next node to gain access to the channel must be chosen correctly. For example, giving precedence to a node with a message whose time constraints are not tight could lead to the loss of a tightly constrained message on another node.

- **Overhead**: In a distributed network environment, nodes usually have little knowledge about the messages waiting on other nodes. Consequently, the access arbitration process may often make its decision based on incomplete or out-of-date information. In order to offset this problem, the access arbitration process may attempt to gather information about the state of other nodes. However, the extra overhead incurred in gathering information
about other nodes results in less channel bandwidth being available to normal message traffic.

For the transmission control processes, one common approach is to allow a node to send only a single message. The access arbitration process will then give another node access to the channel. Another approach is to allow a node to send any number of messages before control passes to another node. In general there are two classes of transmission control processes. In a static transmission control process, the amount of time that a node may continue to send messages over the channel is fixed at design time. In a dynamic transmission control process, the amount of send time is determined during network operation.

The operation of the transmission control process has a large impact on whether the time constraints of messages will be met. The time constraints of messages cannot be met predictably without a guaranteed minimum amount of time in which a node can uninterrupted send messages. On the other hand, allowing a node to occupy the channel for too long can result in message time constraints being violated on other nodes in the network.

Supporting real-time transmissions

To support hard real-time communications, a MAC protocol should exhibit the following properties [58]:

- **Synchronous message utilisation**: A protocol has an achievable utilisation if it can guarantee all synchronous messages from streams whose total utilisation is no more than the achievable utilisation. The worst case achievable utilisation of a protocol is the least upper bound of its achievable utilisations. Provided that the utilisation of a set of synchronous message streams is less than the worst case achievable utilisation, the synchronous messages are guaranteed to meet their deadlines. A message set whose deadlines are guaranteed to be met is known as a schedulable message set.

- **Asynchronous message handling**: In practice, a network for distributed hard-real time systems may handle asynchronous as well as synchronous messages. Even if asynchronous messages are not time constrained, it is important that they have a high throughput in order to avoid problems such as buffer overflow. If asynchronous messages are time constrained, then the protocol should attempt to maximise the proportion of asynchronous messages that satisfy their time constraints. If loss of asynchronous messages is not permissible, then techniques such as using a periodic server for asynchronous messages may be used. Provided that the network is not overloaded, a periodic server can guarantee that all of the asynchronous messages will meet their deadlines.
• **Robustness**: The protocol should be robust in the face of changes to the characteristics of synchronous message streams. A small change in the characteristics of the synchronous message streams should not affect their overall schedulability and a change in the streams originating at a given node should affect only that node.

• **Limited propagation of timing faults**: If the messages in a synchronous message stream are longer than specified or arrive more frequently than specified, then the synchronous message stream violates its specifications. This can disrupt the predicted operation of the network, resulting in deadlines being missed. It is desirable that such a violation of specifications should not affect the other streams in the network. Messages from other streams should still meet their deadlines.

• **Runtime overhead**: The runtime overhead of a protocol includes the time spent by the access arbitration process in arbitrating access to the communication channel. If two protocols give the same schedulability performance for real-time messages, the protocol with the lower runtime overhead is preferred because more channel bandwidth will be available to accommodate future changes in the system.

### 2.2 Real-Time MAC Protocols

As stated above, a properly designed MAC protocol is important to the transmission of real-time messages. In this section, a few real-time MAC protocols, which can help achieve guaranteed deadlines of synchronous messages, are reviewed.

#### 2.2.1 Timed-Token MAC Protocols

The timed-token MAC protocol [25] is a token passing protocol in which the amount of time that a node may hold the token is bounded. It uses transmission control processes in meeting message deadlines. With the property of the bounded channel access delay and the guaranteed channel bandwidth to each node on the network, the timed-token MAC protocol becomes one of the most attractive candidates for use in hard real-time environments. The use of the timed-token MAC protocol with different network standards in different areas to support real-time traffic will be discussed in Section 2.3.

As the work presented here is mostly related to guaranteeing transmission of synchronous messages by allocating proper synchronous bandwidths to nodes, the related details on description of the timed-token MAC protocol and review of synchronous bandwidth allocation will be left to the next chapter, which also serves as the preliminaries for this work. This part
only discuss some relevant issues including robustness, timing faults, runtime overhead and asynchronous message handling.

- **Robustness**: The characteristics of synchronous message streams can thus be altered and all messages will still meet their deadlines provided that the utilisation of the synchronous message streams remains below the Worst Case Achievable Utilisation (WCAU). As will be shown in the next chapter, the WCAU of a specific synchronous bandwidth allocation scheme may be derived.

- **Limited propagation of timing faults**: If a synchronous message stream violates its specification by reducing the inter-arrival period or increasing the message length, only the offending stream will suffer missing deadlines. Other synchronous streams will not suffer because the transmission control process will prevent the offending stream from using more than its share of the network bandwidth.

- **Runtime overhead**: In the timed-token MAC protocol, the access arbitration process merely passes the token from one node to the succeeding node. If the timed-token protocol is implemented on a ring, then there may be several packets on the ring at any time (as in the FDDI standard). In this case the timed-token protocol can achieve reasonable utilisation even when the network bandwidth is large.

- **Asynchronous message handling**: While the transmission of the time-constrained synchronous traffic is being guaranteed, it is also important to improve and even maximise the throughput of asynchronous messages, which is the number of non-real-time messages transmitted per unit of time.

Although most of the work on the timed-token MAC protocol focuses on selecting parameters to guarantee synchronous message transmission, some researchers have tried to address the problem in terms of the transmission of asynchronous messages. Ciminiera *et al* [15] have derived a lower bound for the minimum amount of throughput for asynchronous messages with the timed-token MAC protocol. In [27, 28], Hamdaoui *et al* reduced the mean response time of asynchronous messages while still providing the same quality of service to synchronous messages. The timed-token protocol assigns high priority to synchronous messages; thus asynchronous messages may need to wait a long time before transmission. However, they proposed a scheduling algorithm to allow a node to transmit asynchronous messages first, if there are any, up to asynchronous bandwidth (called “non real time capacity” in their papers). The node can then transmit synchronous messages, if there are any, for as long as the allocated synchronous bandwidth, which can be calculated by any synchronous bandwidth allocation scheme. The leftover
available transmission time will be used to continue transmitting asynchronous messages until either its overall capacity is exhausted, or there are no more asynchronous messages left. One advantage of this method is that it is not restricted to a particular synchronous bandwidth allocation scheme. However, their work can only be used for message streams with deadlines equal to periods. Gencata et al [23, 24] studied the influence of the asynchronous threshold values on the throughput of synchronous and asynchronous traffic. An asynchronous threshold value is defined as a value which can be assigned to each node to limit its asynchronous data transmission. The asynchronous transmission time of a node is then calculated by subtracting the time elapsed since the previous capture of the token and the asynchronous threshold from the TTRT. Asynchronous messages are allowed to be transmitted only if the calculated transmission time is positive. In [24], they proposed two methods to calculate these values. In the first method the proper asynchronous threshold values are calculated to share out the asynchronous bandwidth between nodes in given proportions. In the second method, the actual TTRT value is defined as the difference between TTRT and one asynchronous threshold value.

Thus, asynchronous threshold values can be used to reduce the token rotation time by limiting the asynchronous data transmission in the network. TTRT is an important protocol parameter for transmitting synchronous messages. However, in the original protocol, it is determined in the ring setup process and cannot be changed after the activation of the ring. This may become inadequate for satisfying the real-time requirements when the synchronous message load in the network changes. By using their methods, because asynchronous threshold values can be modified, the actual TTRT value (and the worst case token rotation time) can be adjusted dynamically, corresponding to the changes of the data load or the priority of a node, without interrupting the data transmission.

While improving the throughput of asynchronous messages transmission is important, it is sometimes also important to have asynchronous messages’ deadlines guaranteed. In such cases, three basic methods, the periodic server, the conservative estimation and the dynamic reservation, can be used [58].

- **Periodic server**: With this method, a periodic server is maintained for each source of asynchronous messages that must be guaranteed [72].

- **Conservative estimation**: With this method, messages are guaranteed based on the estimated amount of time that may pass before a given node can access the channel. As long as the estimate is always conservative, message guarantees will always be valid. This method requires that a protocol with bounded access time, such as timed-token protocol, be used [58].
• Dynamic Reservation: With this method, whenever an asynchronous message arrives at a node, a special control message is broadcast (by the node) to all other nodes to inform them that a given time interval is reserved for future use by the node to send the just-arrived asynchronous message. This approach has been adopted for use with the token passing protocol [60].

2.2.2 Other MAC Protocols

Time Division Multiple Access (TDMA)

TDMA protocol uses round-robin scheduling among the synchronous streams [48]. In the protocol, time is divided into frames and further into slots. Synchronous messages can be guaranteed by dedicating one or more slots in each frame to messages from a given stream. A stream that is allocated the \( i \)-th slot of a frame will have exclusive access to the \( i \)-th slot of every succeeding frame.

A TDMA protocol is adopted in the Maintainable Real-Time System (MARS), a fault-tolerant distributed real-time system for process control [47], in order to ensure deterministic load-independent communication delays, in spite of the disadvantage of inflexibility and inefficiency (under low-load conditions) which are inherent with the TDMA protocol. The predictable performance under peak load conditions has been chosen as one of the main design aims of MARS. To achieve this determinism, MARS is strictly periodic (time driven) – all messages are sent as periodic messages.

Controller Area Network (CAN)

Tindell et al [76, 81, 80] studied the problem of guaranteeing message deadlines in a Controller Area Network (CAN). CAN is a well-designed communication bus for sending and receiving short real-time control messages. The bus is designed to connect control systems over a small area (such as automobiles) at a speed up to 1 Mbps.

CAN is a broadcast bus where a number of processors are connected to the bus via an interface. A data source is transmitted as a message, consisting of between 1 and 8 bytes (octets). A data source may be transmitted periodically, sporadically, or on demand. Each data source is assigned a unique identifier (an 11-bit number). The identifier serves two purposes: filtering messages upon reception, and assigning a priority to the message. Data messages transmitted from any node on a CAN bus do not contain addresses of either the transmitting node, or of any intended receiving node. Instead, the content of the message (e.g., Revolutions Per Minute of a car engine, Hopper Full, etc.) is labelled by an identifier that is unique throughout the network.
All other nodes on the network receive the message and each performs an acceptance test on the identifier to determine if the message (and thus its content) is relevant to that particular node. If the message is relevant, it will be presented to the host processor to be processed; otherwise it is ignored. The unique identifier also determines the priority of the message. The lower the numerical value of the identifier, the higher the priority. The numerical value of each message identifier (and thus the priority of the messages from a particular data source) is assigned during the initial phase of the system design. The use of the identifier as priority, combined with a systematic approach to contention in a carrier-sense broadcast bus, enables global priority arbitration when two or more nodes compete for access to the bus at the same time. In CAN, the highest priority message is guaranteed to gain bus access as if it were the only message being transmitted. Lower priority messages are automatically re-transmitted in the next bus cycle, or in a subsequent bus cycle if there are still other higher priority messages waiting to be sent.

To determine the priority of competing messages (so as to realise global priority arbitration), CAN uses the method of CSMA/CD (like Ethernet) but with the enhanced capability of non-destructive bitwise arbitration to provide collision resolution. The priority of a CAN message is determined by the binary value of its identifier. The identifier field of a CAN message is used to control access to the bus by taking advantage of certain electrical characteristics. With CAN, if multiple nodes are transmitting concurrently and one node transmits a 0 bit (termed dominant), then all the nodes monitoring the bus see a 0. Conversely, only if all nodes transmit a 1 bit (termed recessive) will all nodes monitoring the bus see a 1. So, in effect, the CAN bus acts like a large AND-gate, with each node being able to see the output of the gate. Such electrical characteristics (with the wired-and mechanism) is used to resolve any potential bus conflicts and enables global priority arbitration as follows. Each node waits until the bus becomes idle. When silence is detected, each node begins to transmit the highest priority message held in its queue whilst monitoring the bus. The message is coded so that the most significant bit of the identifier field is transmitted first. If a node transmits a recessive bit of the message identifier, but monitors the bus and sees a dominant bit then a collision is detected. The node knows that the message it is transmitting is not the highest-priority message in the system, stops transmitting, and waits for the bus to become idle again. If a node transmits a recessive bit and sees a recessive bit on the bus then it may be transmitting the highest-priority message, and proceeds to transmit the next bit of the identifier field. Since CAN requires identifiers to be unique within the system, a node transmitting the last bit (least significant bit) must be transmitting the highest-priority queued message, and hence can start transmitting the body of the message. Note that the highest-priority message undergoes the arbitration process without
incurring any disturbance since all other nodes will have backed off and discontinue transmission until the bus is next idle. The whole message is transmitted without interruption.

While the problem is generally perceived that it was impossible to determine the worst case response time of a given message (i.e., the longest time between queuing the message at the source processor and the message arriving at the destination processor), an analysis developed for a simple CAN model is given by Tindell et al [76, 81, 80] that can bound such latencies for all CAN messages, including not only the highest-priority message but also the lowest priority message. Therefore it is possible to guarantee even a message with the lowest priority. Because CAN is primarily a priority-based bus, they conducted the analysis based on an analysis previously developed for fixed priority processor scheduling [6, 75, 77, 78, 79]. In their analysis, messages do not have to be strictly periodic, and they can be sporadic. The analysis developed has been applied to a realistic case study. Their analysis shows that CAN is eminently suitable as a bus for hard real-time applications.

The drawback of CAN is that its maximum speed (1Mbps) is relatively slow. Further, the relatively high overhead incurred in the bitwise arbitration process forms a substantial portion of the short CAN message, which could lead in practice to a low utilisation of available network bandwidth.

2.3 The Use of the Timed-Token MAC Protocol

The timed-token protocol, as a MAC layer logical ring protocol, has been incorporated into some network standards such as FDDI, IEEE 802.4 and SAFENET. Studies were also reported on how to use it with many other network standards, including the IEEE 802.3 Ethernet standard (often called Automated Factory Networks or Industrial Ethernet) and the popular IEEE 802.11 wireless LAN standards (including its application in the area of Home Networking), which are important for building the future pervasive computing infrastructure. This section will review the application of the timed-token protocol in different areas to show that the timed-token protocol and its basic token passing idea are important to many network standards for supporting real-time traffic including Quality of Service (QoS) required data.

2.3.1 Broadcast Networks

Broadcast networks are typically set up as a shared-bus network, where any node can talk directly to any other node or all other nodes. The best-known example of such a network is Ethernet. Although widely deployed today, Ethernet was not originally developed to meet the requirements of real-time traffic due to the use of Carrier Sense Multiple Access/Collision
Detection (CSMA/CD) and Binary Exponential Backoff (BEB) in resolving concurrent medium accesses. When two or more nodes try to transmit a message simultaneously, the BEB algorithm can take up to 7151 time slots before the packet is delivered or discarded [30]. This unrealistically large waiting time and yet rarely occurring collisions make Ethernet highly impractical for real-time transmission. To support real-time traffic, several techniques have been developed, including the adoption of the timed-token protocol and the idea of the token ring network.

**RETHER**

The real-time network RETHER [84, 85, 86], developed at the State University of New York at Stony Brook, is based on Ethernet. It uses a token-based technique to regulate access of all the nodes to the network. This circumvents the danger of collisions inherent to the CSMA/CD protocol employed on Ethernet networks.

RETHER has two operating modes, a regular CSMA/CD mode in absence of real-time traffic and a token passing mode for real-time operations which is essentially a token-bus network mode. The token passing mode will be initiated as soon as a node has real-time traffic to transmit. However, the token does not follow a simple logical circular path through the network anymore. The token is guided according to a certain schedule. It first flows to all nodes which have real-time traffic to send. These nodes may have different bandwidth requirements that translate into different token holding times. If there is still time available before the next token rotation period, the token will flow through the nodes that do not have real-time traffic. This path is interrupted as soon as the schedule dictates a node's need for transmitting real-time traffic. The system will switch back to CSMA/CD mode when there is no node to send real-time messages.

The system employing RETHER works on a single link (contention domain). A switch will split the system in different domains (as many domains as ports on the switch). Each domain will have an independent token operation.

**RTnet**

RTnet is a distributed real-time network protocol [31, 32]. It can be used on fully-connected local area networks with a broadcast capability such as Ethernet to meet real-time requirements. RTnet is a token based protocol whose idea originally comes from the timed-token MAC protocol. Thus it inherits some important properties from the timed token protocol, including collision prevention and deterministic media access. Like most token based protocols, it allows for only one node to hold the token and transmit messages as long as Token-Holding Time (THT) at any given time. However, compared with RETHER which works between the CSMA/CD mode and
the timed-token mode, RTnet always works in the timed-token mode and thus will not suffer
the long delay when RETHER transients between the two modes. RTnet also introduces some
important properties which are different from the normal token protocols, including RETHER.

- RTnet uses a token as a form of shared memory between the nodes to store the information
  of the whole network and message streams. A fully-connected, single-segment network
  thus can be supported because any node can realise the current state of the network
  and messages, and can forward the token to any other node directly. This broadcast
  capability of RTnet makes it naturally suitable for working with broadcast networks, such
  as Ethernet, and supports on-the-fly addition and removal of network nodes.

- Most token-based protocols use a simple static round-robin method to forward the token
  among the nodes, even for RETHER which has two working modes. However, RTnet uses
  preemptive Earliest Deadline First (EDF). It can also support any other desired real-time
  scheduling algorithm such as Rate Monotonic (RM) or Deadline Monotonic (DM). These
  scheduling algorithms distribute the token dynamically according to bandwidth demands.

- The $THT$ is not fixed in RTnet anymore due to the use of the shared memory formatted
token and the EDF scheduling algorithm. When the $THT$ has expired the network sched-
ule determines the next node that may transmit and recalculates its $THT$ based on the
network state kept in the token.

- A monitor node is introduced to watch the current token holder. This mechanism is used
to deal with token loss, token duplication or deadline misses due to network faults. A
monitor is not a fixed specific node in the network but the previous token holder. When
the node passes its holding token, it enters into the monitor state until the succeeding
token holder passes its token and in turn becomes a monitor.

Both RETHER and RTnet are examples of explicit-token Ethernet-based protocols. These
protocols use a token message to allow nodes to transmit. Although these protocols deal with
different traffic patterns, they inevitably introduce extra overhead in the case of token loss. The
alternative mechanism is implicit-token Ethernet-based protocols which are based on the Timed
Packet Release (TPR) mechanism. In implicit-token protocols, a token is implicitly passed on
to the next node independently of whether or not the previous node on the ring transmits its
message. The absence of message means that a node is giving up its right to use the bus. Thus,
the bus can be utilised more efficiently. The Virtual Token Passing Ethernet (VTPE) is such an
Ethernet deterministic proposal based on implicit token rotation [9, 11, 10]. It uses the producer-
consumer cooperation model to exchange data over the bus. The producers are active devices

24
that can access the bus when they are allowed to do it while consumers are passive devices and can only consume the data on the bus. Every producer has a constant Node Address (NA) and an Access Counter (AC) which is increased after the interrupt caused by a frame sent to the bus. The VTPE system architecture consists of a producer’s logical ring and every producer can only transmit a single frame when it holds the virtual token, which is the time at which its AC equals its NA. If the node has nothing to transmit, the bus becomes idle and, after a certain time, all the access counters are increased by one. Thus the next producer is then allowed to access the bus. Since there is no extra specific token in the protocol, the VTPE, as its proposer claimed, can be used at field level, resource constrained devices such as embedded systems. The Double Ring Service for Real-Time Ethernet (DoRiS) [67], is a technique mixing two approaches together, TDMA and implicit token, to support real-time traffic. TDMA is used to create two windows to predictably deal with hard and soft messages, respectively. Similarly to VTPE, the bus access in each window is controlled by organising the nodes in two logical rings, one for each window, and by using an implicit token so that bus utilisation can be optimised. Interested readers can find out about other implicit-token protocols, e.g., TEMPRA [66].

2.3.2 Wireless LAN Networks

Wireless LAN networks are becoming more and more popular due to their advantages of no-wires and mobility properties. There are many challenges in the research in this area. One of them is using a proper MAC to control nodes to access medium and determine QoS and channel utilisation [17]. However, the unreliability and instability properties of wireless channels make it extremely challenging to achieve real-time transmission in a wireless networking environment. Especially when the network is constructed using an ad hoc mode where no central coordinator can be used to coordinate the transmission between stations as in infrastructure mode, the problem becomes more difficult because of hidden terminals, dynamic network structures and a partially connected network topology.

Unfortunately, the de facto wireless local area network standard, IEEE 802.11 [1], is not suitable for real-time transmission. The basic access method of its MAC layer is the Distributed Coordination Function (DCF), which is a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). This method does not support real-time traffic due to its random access method, unpredictable exponential backoff mechanism, frequent retransmissions and equal treatment of all types of traffic. The 802.11 also defines an alternative access method to offer time-bounded service known as the Point Coordination Function (PCF) which has a central station polling the slave stations. However, PCF has been identified as having many inadequacies and it is
not supported in most vendors’ products. When used for ad hoc networks, contention among stations using the 802.11 MAC protocol can cause severe throughput degradation if the load becomes high, which also results in large medium access times.

From another perspective, the token-based technique is suitable for real-time transmissions because of its contention-free nature, bounded media access time and attractiveness under both light and heavy loading. It outperforms CSMA schemes and has been proved to be the most reliable MAC scheme in the industry [21]. Although compared with its use in the wired area, using the token passing structure in a wireless ad hoc network has more challenges due to the unreliable characteristics of wireless medium, the hidden terminals and the dynamics of nodes [40], this technique is gaining interest amongst wireless ad hoc network researchers due to its unrivalled advantages over the existing IEEE 802.11 standards.

Token Passing Tree (TPT) [40] is a protocol for supporting QoS applications in indoor ad hoc networks in which terminals have low mobility and limited movement space. Each node in a network is assumed to try to communicate with one another over a single hop, or with an access point or cluster head to reach a hidden node. Thus, each sender has sight of the access point or cluster head, which is reasonable for hierarchical ad hoc networks. The TPT structure looks like a bi-directional tree with the token-rotation beginner being the root with two logical token-passing directions: forward and backward. New nodes can join the TPT by using the exchange of Request to Join/Clear to Join with the nodes in the TPT, in parallel with and without influence on the normal token passing operation. The TPT and its performance are derived directly from the timed-token protocol, and can provide a delay bounded service for real-time traffic. The TPT also manages hidden terminals by its hierarchical network structure and tree token passing loop. Due to its use in unreliable wireless environments, the TPT adopts a distributed timer to detect token passing failure and ensures that the token passing loop is recovered without causing much overhead. However, the TPT does not work well in the case of a network without the cluster where multi-TPT may exist because the broadcasted message may not reach all nodes. With the multi-TPT, the token rotation of one TPT may interfere with the token rotation of its neighbour’s TPT.

Wireless RETHER protocol [69] is a modified version of Ethernet RETHER protocol mentioned above. Unlike the Ethernet RETHER protocol which circulates a software token among the Ethernet nodes within a fixed length cycle, the Wireless RETHER protocol implements a centralised token-passing protocol because of the movement of the mobile nodes and the hidden terminal problems. A central server, Wireless Rether Server (WRS), is placed directly between the access point and the wired network and is responsible for granting the token to wireless LAN nodes, called Wireless Rether Client (WRC), using a mandatory token-ACK message pair.
Wireless RETHER supports a non-work-conserving service model where the token is still passed among non-real-time nodes even when there is no non-real-time traffic. However, there are several problems with the Wireless RETHER protocol including a single point of failure in the WRS because of the centralised token and transmission overhead caused by the token-ACK message pair. Also, this protocol can only be used for an infrastructure mode wireless network.

The Wireless Token Ring Protocol (WTRP) [20] is in fact a token bus protocol. It follows a distributed approach that supports many topologies, as not all nodes need to be connected to each other or to a central station. The main functions of its MAC layer are ring management and timing of the transmissions. The nodes in the network can form more than one logical ring, which can be merged together when independent rings using the same channel come into range of each other. Every ring is ensured to have only one token passing along the logical ring by introducing the notion of token priority computed from the ring address and special fields in the token. Stations in the ring can only accept a token that has greater priority than the token the station last accepted. The WTRP is much better than IEEE 802.11 in terms of support real-time traffic, however, it may still not be attractive for military networks because excluded nodes may remain disconnected for an unbounded time before rejoining the network [41].

The High-Frequency Token Protocol (HFTP) [41] was designed based upon WTRP by adding mechanisms for relaying tokens around link outages and for merging rings without disintegrating those rings. Compared with WTRP, HFTP is more attractive for its fast recovery from disruptions and tolerance for long link turnaround times.

There are numerous other studies on using the idea of token ring to support real-time traffic on wireless networks and some of them will be briefly discussed below. The T-MAH protocol [61] organises the network into clusters called Token Groups which can be overlapped by one or more nodes. The protocol also introduces the concept of Token Group Head as the leader of the group. In each group logical infrastructure, a token passing protocol is used to schedule the token passing along the logical ring to control real-time transmission. The hybrid Token-CDMA MAC protocol is proposed in [52], which is based on a token passing scheme with the incorporation of code division multiple access (CDMA). It combines the advantages of guarantee-access characteristic of the token passing mechanisms and the supportability of multiple packet transmissions. It also takes the error-prone wireless channel condition into consideration. VTP-CSMA [63] is a virtual token passing procedure. It circulates a virtual token among real-time devices with a complement from an underlying traffic separation mechanism that prioritises the real-time traffic over the non-real-time traffic. Ripple [14] is a wireless token-passing protocol for wireless mesh networks, protecting nodes from unintentional packet collisions and maximising spatial frequency reuse. Interested readers can find more relevant work, e.g., [82] and [53].
2.3.3 Home Networks

The home network, which completes the broadband access scenario in approximately the last 100 yards, has more stringent requirements to support entertainment multimedia streams like video among TVs, PCs, DVDs and so on. To simplify use and to reduce cost, most research has focused on the no-new-wires solutions, which include the old wired technologies using either existing in-home cables or wireless technologies to effectively reduce the cost. Some typical solutions include the phone line based solution HomePNA [34], the power line based solution HomePlug [33] and the voice-only-supporting solution HomeRF [64]. However, due to the adaptation of the conventional CSMA/CD (HomePNA) or CSMA/CA (HomePlug and HomeRF), these solutions suffer problems of high delay and low channel utilisation of real-time traffic by the backoff at heavy load levels. As these solutions do not work well with the real-time traffic, other solutions are proposed. The timed-token MAC protocol, which is identified as one of the best guarantee solutions, is also taken into consideration by much work. The Token Passing Tree (TPT) [40] discussed earlier can be used as a protocol for home networks. One dedicated home network protocol using timed-token protocol is the HomeMAC [35, 37, 36].

HomeMAC is a no-new-wire solution for home network MAC protocol. Its working mechanism is similar to RETHER used for Ethernet. The HomeMAC features adaptive hybrid access protocol of CSMA/CD and the timed-token protocol to support real-time traffic. The HomeMAC classifies stations into two groups: Resource Reservation (RSV) stations (which have real-time (RT) traffic to transmit by allocating bandwidth) and Non-RSV (NRSV) stations. It maintains two logical rings: the Normal ring which includes all stations and the RSV ring which consists of the RSV stations and the Resident Gateway (R/G). HomeMAC starts to access a medium by the 1-persistent CSMA/CD and works between the CSMA/CD mode and the timed-token mode. HomeMAC differs from RETHER by using the R/G, which can normally be found in a home network. The transition between two working modes is controlled by the R/G using a request-and-ACK format. When the network works in the timed-token mode, it is the R/G which creates and maintains the token.

Takahashi et al [73] also considered using token passing and retransmission by the hub station in the asynchronous transfer mode of Wireless 1394 [88] in high data rate home network. They adopted token passing to improve the waiting time performance by transferring the token packets among stations directly. The performance on waiting time of their approach is superior to that of the polling scheme originally used by Wireless 1394. In [44], three transmission schemes were proposed to improve the transmission of the most notable Variable Bit Rate (VBR) video MPEG, which can be widely found in a home network, over a timed-token MAC network.
2.3.4 Notes on Employing the Timed-Token MAC Protocol with Other Network Standards

Due to its bounded channel access delays and guaranteed amount of communication capacity for every node in the network, the timed-token MAC protocol is one of the most attractive candidates to support real-time communication. This section presents a brief review of the application of the timed-token in different areas as a MAC layer logical ring protocol to support real-time traffic, including QoS required data. It is shown here that the timed-token can be carefully designed to work with the widely-used Ethernet, the wireless networks and the new emerging home networks. Actually, the timed-token and its basic ideas have also been used in other areas such as supporting the rate-guaranteed flows in order to meet the required QoS in multi-service networks [49, 50], where the Timed Token Service Discipline (TTSD) is applied at the output link of a switch adopting the same rules used to control medium access in the timed-token protocol. PROcess FIeld BUS (PROFIBUS) [83], as one of the most widely known fieldbuses used in the industrial arena, employs a simplified version of the timed-token protocol as its MAC protocol.

Some considerations when the timed-token passing protocol is employed with other network standards are summarised below.

- Generally, there are two methods to manage the passing of the token: distributed or centralised. In a distributed design, the token is passed among the nodes along a logical ring while in a centralised design, the relay of the token to the next node must be granted by a central server by some request-ACK-like mechanisms. The advantage of introducing a central server is the awareness of the whole network in the central server, which can be used to coordinate the transmission of the token. However, it also introduces a single point of failure in the central server, the wastage of available bandwidth and the additional cost of the central server. In the wired area where the communication channel is usually stable, the distributed token could be a better choice. In the context of wireless networks, things are more complex. Because the wireless channel is unstable and the mobile nodes can move out of each other’s transmission range, direct token passing between wireless nodes may be impractical, thus using the centralised design could be a good choice. This is especially true in an infrastructure mode network where Access Point (AP) or wireless router can be taken as the centre server and the hidden terminal problem can be circumvented naturally. However, even with many advantages, most ad hoc networks still adhere to the distributed token design for its dynamic property. These two methods can be combined together. Every node in a logical ring can be elected as a central server for a certain amount of time.
by some mechanisms. This method combines the advantages of the two methods, but at the mean time, the overall overheads of the protocol would be increased.

• The structure of the ring and the schedule of the token also need to be carefully considered. The nodes in a network can form a single logical ring or multiple logical rings. The token always travels from one node to its succeeded node in every logical ring. However, from the perspective of the schedule of the token, there are different ways to construct every logical ring. The nodes can be taken equally or can be grouped into different groups by some mechanisms. For example, the nodes can be divided into two groups based on whether they have real-time traffic to send, or the nodes can be divided into several groups according to the priorities of the real-time traffic residing on the nodes. The travel of the token among nodes can follow the incremental order of physical addresses of nodes in a single group, or can first visit the nodes which have real-time/high priority traffic and then the remaining nodes in multiple groups. For different schedules of the token, a proper selection of $TTRT$ is always important. Multiple logical rings can usually be found in wireless networks where the nodes in the network are partially connected and the hidden terminal problem needs consideration. Additional attention should be paid to multiple logical rings: the relationship of these rings (e.g., parallel or hierarchical) and the merging of the rings.

• Another important issue is the bandwidth allocation which decides the duration that a node can access the media when it holds the token. The proper bandwidth allocation can help achieve an efficient real-time message transmission, a high bandwidth utilisation and a fair access among all contending nodes in the networks. The bandwidth can be statically assigned to every node at network initialisation time, or it can be decided dynamically at runtime for each time the node receives the token. This research focuses on the bandwidth allocation for the original timed-token MAC protocol. However, the results for the timed-token MAC protocol can also be applied to some other token-based MAC protocols [19].

• Other considerations include the token format, token error, nodes leaving or joining the network dynamically, combination with other technologies and the place to implement. The token could be a single bit, a piece of shared memory containing all the information of the network or other formats. The design of the format must make sure that the token can be passed efficiently, especially when the token contains much information (because the transmission of the token takes time and consumes a certain amount of bandwidth). Token error includes token loss and token duplication. A new token should be recreated in an appropriate place, such as a central server or a precedence node, when the token is determined to be lost. Duplicate tokens should be eliminated. Dealing with the token
error is relatively difficult in wireless networks where channel error can occur frequently and networks are normally partially connected. Nodes dynamically leaving or joining a network also need to be carefully considered, especially in a wireless network where nodes can move freely. Timed-token can also be used with other technologies such as TDMA and CDMA to support real-time traffic more effectively. Usually, to make a non-real-time protocol like Ethernet and 802.11 support real-time traffic, a real-time solution can be integrated into these protocols, or it can be built as a top layer of these protocols. The advantage of the integration method is that it can significantly reduce the protocol overhead and simplify the management. The disadvantage of this method is that the solution is not supported by most hardware vendors as it changes the original protocols and increases the hardware cost. In contrast, building a layer on top of existing protocols, usually a software top layer, can be more easily implemented and modified on existing hardware without costing much extra.

2.4 Summary

In the area of hard real-time communication, LAN models and MAC protocols must be carefully selected to guarantee the transmission of hard real-time messages. With the bounded media access time, the timed-token MAC protocol becomes one of the most attractive candidates for use to support hard real-time traffic via its transmission control processes. It has been incorporated into many network standards, including IEEE 802.3 Ethernet and IEEE 802.11 wireless LAN, to support real-time transmission.

Hard real-time messages can be classified into two categories: synchronous and asynchronous. Different strategies are used based on the type of messages. The timed-token MAC protocol can be used for both types of messages. However, most prior work focused on guaranteeing the transmission of synchronous messages.
Chapter 3

Preliminaries

This chapter describes the framework under which the proposed research has been conducted. In order to avoid confusion and to help understanding and comparing related works, the same framework is used as that adopted by many researchers [2, 3, 12, 29, 56, 90, 92, 94, 96] in their studies on synchronous bandwidth allocation with the timed-token protocol.

This chapter is organised as follows: Section 3.1 introduces the network model and the message model which are used in this research. A brief description of the timed-token MAC protocol is then given in Section 3.2. Section 3.3 states the basic ideas on synchronous bandwidth allocation (SBA) schemes, including definition, classification, constraint requirements, performance metric and some previously published schemes. The timing properties of the timed-token MAC protocol is finally discussed in Section 3.4.

3.1 The Network and Message Models

This section introduces the network and message models which have been widely used to simplify and facilitate the study and analysis without losing generality [3].

3.1.1 Network Models

The network is assumed to consist of \( n \) nodes being connected to form a logical ring without any hardware or software failure. A special bit pattern called the token rotates around the network, i.e., from node 1 to node \( n \) and back to node 1. Message transmission among these nodes is controlled by the timed-token protocol. A node is allowed to transmit its messages only when it holds the token for as long as the protocol permits.

Let \( \tau \) be the maximum possible amount of inevitable overheads involved, such as ring latency...
and other protocol/network dependent overheads during one complete traversal of the token around the ring and \( \alpha \) be the ratio of \( \tau \) to \( TTRT \), i.e., \( \alpha = \tau / TTRT \). Since \( \alpha \) represents the proportion of time unavailable for transmitting messages, \((1-\alpha) \cdot TTRT\) would be the maximum allocatable amount of time for message transmission during one complete token rotation.

### 3.1.2 Message Models

For simplicity, it is assumed, without losing generality, that each node \( i \) only has one synchronous message stream, denoted as \( S_i \). Each \( S_i \) can be characterised using three parameters: \( C_i, P_i, \) and \( D_i \), i.e., \( S_i = (C_i, P_i, D_i) \), where

- \( C_i \) is the maximum amount of time needed to transmit a message from stream \( S_i \);
- \( P_i \) is the inter-arrival period between any two consecutive messages from \( S_i \). That is, if the first message from \( S_i \) arrives at time \( t_0 \), then the \( k \)-th message will arrive at time \( t_0 + (k-1) \cdot P_i \), where the integer \( k \geq 2 \);
- \( D_i \) is the relative deadline of messages from \( S_i \). If a message from \( S_i \) arrives at time \( t \), then the message must be sent before \( t + D_i \). If the messages are scheduled in First In First Out (FIFO) order, and the first message arrives at time \( t_0 \), the \( k \)-th message must be sent before time \( t_0 + (k-1) \cdot P_i + D_i \).

For a network with \( n \) nodes, there are in total \( n \) synchronous message streams, denoted as \( S_1, S_2, \cdots, S_n \). Let \( M \) be the synchronous message set, i.e., \( M = \{S_i|i = 1, \cdots, n\} \), the utilisation of a synchronous message set \( M \), denoted as \( U(M) \), can be defined to be

\[
U(M) = \sum_{i=1}^{n} \frac{C_i}{P_i}
\]

which is the proportion of time required for synchronous traffic in the network.

### 3.2 Timed-Token MAC Protocol

The basic ideas of the timed-token protocol were first presented by Grow [25] in 1982. With this protocol, messages are grouped into two types: synchronous messages and asynchronous messages. Synchronous messages, such as voice or video traffic, are periodic messages which have delivery time constraints. Asynchronous messages are aperiodic and do not have any hard real-time deadline constraints. At network initialisation time, all nodes negotiate a common value for the Target Token Rotation Time \( (TTRT) \), which is an important protocol parameter. The \( TTRT \) should be chosen small enough to meet responsiveness requirements of all nodes, i.e.,
the negotiated value for \( TTRT \) should be small enough to satisfy the most stringent response-time requirements of all nodes. However, if the chosen \( TTRT \) becomes too small, the maximum network utilisation (of \( 1 - \frac{\tau}{TTRT} \)) will become low. Each node is then assigned a fraction of the \( TTRT \), known as its synchronous bandwidth (denoted as \( H_i \) for node \( i \)), which is the maximum time that the node is allowed to transmit its synchronous messages every time it receives the token [4, 12]. Whenever a node receives the token, it is permitted to transmit its synchronous messages, if any, for a time period no more than its allocated synchronous bandwidth. The asynchronous messages can only be transmitted if the token arrives earlier than expected since the token’s last visit to the node. That is, synchronous traffic is assigned a guaranteed bandwidth while the leftover bandwidth (unallocated, unused or both) is dynamically shared among all the nodes for asynchronous traffic.

More specifically, each node has two timers and one counter:

1. Token Rotation Timer of node \( i \) (\( TRT_i \)): This timer is initialised to \( TTRT \) and is always enabled. \( TRT_i \) counts down until it either expires (i.e., \( TRT_i = 0 \)) or the token is received early (i.e., earlier than expected since the token’s last arrival at the same node). In either situation, the \( TRT_i \) is reinitialised to \( TTRT \) and enabled again (starting the count down process).

2. Late Counter of node \( i \) (\( LC_i \)): This counter is initialised to zero and used to record the number of times that \( TRT_i \) has expired since the token last arrived at node \( i \). \( LC_i \) is incremented each time \( TRT_i \) expires and is reset to zero whenever node \( i \) receives the token. During the normal ring operation, \( LC_i \) should not exceed one. However, if \( LC_i \) exceeds one because of faults, the ring recovery process will be initialised [42]. The token is said to arrive early at node \( i \) if \( LC_i \) is zero when the token arrives at node \( i \). Otherwise, if \( LC_i \) is one, the token is considered to be late.

3. Token Holding Timer of node \( i \) (\( THT_i \)): This timer is set to the current value of \( TRT_i \) on the token’s arrival at node \( i \) (only if the token arrives early). This timer also counts down, but is enabled only during asynchronous transmission in order to control the amount of time for which the node \( i \) can transmit asynchronous messages. \( THT_i \) is set to zero and disabled at ring initialisation since no asynchronous messages are permitted to transmit during the ring initialisation period.

When the token arrives early at node \( i \), the current value of \( TRT_i \) is placed in \( THT_i \) and \( TRT_i \) is then reset to \( TTRT \). Synchronous frames, if any, can be transmitted for a time not exceeding its allocated synchronous bandwidth (\( H_i \)). The node may then transmit its asynchronous frames
(if any) until $THT_i$ or $TRT_i$ expires (i.e., as long as both $THT_i$ and $TRT_i$ are greater than zero). On the other hand, when the token is late on its arrival at node $i$ (i.e., $LC_i = 1$), the $LC_i$ is reset to zero and $TRT_i$ continues to count down. In this case, node $i$ is still permitted to transmit synchronous frames for a time not more than its allocated $H_i$ but no asynchronous frames are allowed to transmit. Note that $TRT_i$ is not reset to $TTRT$ as in the case when the token arrived late ( $LC_i = 1$).

The amount of time left, before the initiation of the ring recovery process by node $i$, can be expressed as a function of two parameters at that node: $TRT_i$ and $LC_i$ [13]. A single parameter $TRT_i'$, which captures the values of both $TRT_i$ and $LC_i$ within it, indicates the amount of time left before the initiation of the ring recovery process by node $i$. It is defined as below

$$ TRT'_i = TRT_i + (1 - LC_i) \cdot TTRT $$

Given that $0 \leq TRT_i \leq TTRT$ and $0 \leq LC_i \leq 1$, it is clear that $0 \leq TRT'_i \leq 2 \cdot TTRT$. The token is considered to arrive at node $i$ early if $TRT'_i > TTRT$ at the instant of its arrival at node $i$ because $TRT_i$ has not expired since the token’s last arrival at the node. Otherwise, the token is no earlier than expected or late if $TRT'_i \leq TTRT(LC_i = 1)$ at the instant of the token’s arrival (because $TRT_i$ has expired once since the last token arrival). In either case, $TRT'_i$ records the amount of time left before the ring recovery process is initiated by node $i$. The ring recovery process will be initialised when $TRT'_i = 0$.

A more detailed description of the timed-token protocol can be found in [39].

### 3.3 Synchronous Bandwidth Allocation Schemes

A synchronous bandwidth allocation (SBA) scheme is an algorithm that produces the value of the synchronous bandwidth $H_i$ to be allocated to node $i$ for transmission of synchronous messages. When the above message model is used, under the assumption that each node $i$ has only one synchronous message stream $S_i$, an SBA scheme can also be considered as an algorithm that produces $H_i$ to be allocated for $S_i$.

#### 3.3.1 Classification

SBA schemes can be classified into two categories [2]: local SBA schemes and global SBA schemes. The difference between these two categories is the type of information which is used in allocating synchronous bandwidth to a node. A local SBA scheme uses only node $i$’s local information, which includes the parameters of the stream $S_i$ (i.e., $C_i$, $P_i$ and $D_i$) residing on this node, $TTRT$ and $\tau$. Although $TTRT$ and $\tau$ are parameters of the network, they are still
regarded as the local information because they are known to all nodes. In contrast, a global SBA scheme will use global information, which includes all the information from all nodes in a network. Let \( f_L \) and \( f_G \) be a local SBA scheme and a global SBA scheme, respectively. Then a local SBA scheme can be expressed as

\[
H_i = f_L(C_i, P_i, D_i, TTRT, \tau) \quad (i = 1, 2, \ldots, n)
\]

while a global SBA scheme can be represented as

\[
\bar{H} = f_G(C_1, C_2, \ldots, C_n, P_1, P_2, \ldots, P_n, D_1, D_2, \ldots, D_n, TTRT, \tau)
\]

A global SBA scheme may perform better than a local one because it uses system-wide information. However, if any parameter of \( S_i \) changes, it may have to recompute the synchronous bandwidths at all nodes. This makes a global SBA scheme unsuitable for use in a dynamic environment compared to a local SBA scheme. When a local SBA scheme is used, any changes on \( S_i \) will only affect \( H_i \) without disturbing other nodes. Because a local scheme is more flexible and suitable for use in dynamic environments with its performance close or comparable to that of a global scheme, a local SBA scheme is more preferable from a network management perspective. This research only studies local SBA schemes. A novel local SBA scheme will be proposed in Chapter 4.

### 3.3.2 Constraints

Allocating a proper synchronous bandwidth to a node is important for timely transmission of synchronous messages. A node with inadequate synchronous bandwidth may not have enough time to transmit a synchronous message before its deadline. On the other hand, allocating too much synchronous bandwidth to a node may let the token arrive at other nodes too late. Thus, the messages at other nodes may miss their deadlines. In order to guarantee synchronous messages with arbitrary deadlines, the following three constraints must be satisfied [56].

- **Protocol Constraint**: The sum total of the synchronous bandwidth allocated to all nodes in the ring should not be greater than the available portion of the \( TTRT \), i.e.

\[
\sum_{i=1}^{n} H_i \leq TTRT - \tau \quad (3.2)
\]

- **Deadline Constraint**: Every synchronous message must be transmitted before its deadline. Formally, let \( s_{i,j} \) be the time that the transmission of the \( j \)-th message in stream \( S_i \) is completed. The deadline constraint implies that for \( i = 1, 2, \ldots, n \) and \( j = 1, 2, \ldots \).
\[ s_{i,j} \leq t_{i,j} + D_i \]  \hspace{1cm} (3.3)

where \( t_{i,j} \) is the arrival time of the \( j \)-th message of stream \( S_i \) residing on the node \( i \).

As a special case, for a synchronous message stream \( S_i \) with its deadline no larger than its period, any message in that stream must be sent out before the arrival of the next message because \( s_{i,j} \leq t_{i,j} + D_i \leq t_{i,j} + P_i = t_{i,j+1} \). Thus the deadline constraint implies that the minimum available time for node \( i \) to transmit its messages during the time interval \([t_{i,j}, t_{i,j} + D_i]\), denoted as \( X_i(D_i) \), must be no less than the message transmission time \( C_i \), i.e.

\[ X_i(D_i) \geq C_i \]  \hspace{1cm} (3.4)

- **Buffer Constraint:** This constraint states that the size of the buffer for outgoing synchronous messages at node \( i \) must be large enough to hold the maximum number of synchronous messages that may be awaiting transmission at node \( i \) at any time. This constraint is necessary to ensure that messages are not lost due to buffer overflow\(^1\).

These three constraints result from the requirements of the timed token protocol itself, hard real-time communication application and the limited memory in real systems, respectively.

A synchronous message set \( M \) can be guaranteed by a SBA scheme if an allocation, which satisfies the above three constraints, can be produced by the scheme. An allocation \( \vec{H} \) is feasible if it satisfies all the above three constraints. A synchronous message set is said to be schedulable if there exists at least one feasible allocation for the message set.

### 3.3.3 Performance Metric

A metric that has commonly been used in real-time processing and real-time communication, to evaluate and compare the effects of different SBA schemes on the performance of the network, is the Worst Case Achievable Utilisation (WCAU).

\(^1\)The buffer size is important in the area of real-time communication even with a fast speed processor and a large amount of memory. In this research, the buffer size reflects the length of waiting queue in the MAC layer. In the real-time systems where only the case of the deadline no more than the period is considered, the buffer size is not an issue. However, in the case where the deadline is larger than the period, the number of messages waiting in the queue may keep being accumulated. Thus, it is important to identify if there is a maximum length of such queue to prevent the overflow of the buffer. The overflow of the buffer may still happen without a proper selection of synchronous bandwidth. Thus the buffer constraint has to be satisfied for a hard real-time communication system.
Liu and Layland [51] addressed the issue of guaranteeing the deadlines of periodic tasks in a single processor environment in terms of the WCAU. Their work proposed a methodology that stresses the fundamental requirement of predictability and stability in any hard real-time environment. This research adopts the metric of the WCAU, which has been commonly used by many researchers [3, 4, 57, 56, 94] to judge if one SBA scheme performs better than another.

\( U \) is said to be an Achievable Utilisation of a SBA scheme if every synchronous message set whose utilisation factor is no more than \( U \) can be guaranteed by this scheme. The WCAU, denoted as \( U^* \), is the least upper bound of achievable utilisations (\( U_s \)). That is, as long as the utilisation factor of a synchronous message set is no more than \( U^* \), the message set can be guaranteed by this scheme.

The WCAU of a SBA scheme is a similar metric to the worst case utilisation of the rate monotonic algorithm for single processor scheduling [51], in the sense that it evaluates the predictability of a hard real-time communication system and gives a measure of the stability of the system, and therefore can be used to greatly simplify network management in practice [4]. In particular, there are three major advantages of this metric [3]:

1. evaluating the predictability of hard real-time communication systems, i.e., as long as the utilisation of a synchronous message set is no more than the bound specified by the WCAU metric, all synchronous messages in the set will meet their deadlines.

2. giving a measure of the stability; that is, as long as their total utilisation is kept within the bound, the parameters of synchronous messages can be freely modified.

3. being used to simplify network management considerably, in the sense that as long as the total utilisation of the time-critical synchronous message set is kept within the bound, the network manager can be certain that the message set will be transmitted without deadlines being missed.

The WCAU metric is useful for testing the schedulability of some synchronous message sets because it only offers a sufficient condition to testing the schedulability. That is, some message sets whose utilisations are larger than the WCAU may still be schedulable. However, the WCAU is still an important schedulability testing method because it does not need detailed knowledge of all real-time messages in the system and may be used at design time when only a rough estimate of the amount of real-time traffic is known. Schedulability tests developed based on the WCAU metric are called the utilisation-based tests [59]. In Chapter 5, the WCAU of the new SBA scheme proposed in this thesis will be studied.
3.3.4 Previously Published SBA Schemes

Four SBA schemes, including two local allocation schemes – FLA (Full Length Allocation, see Eq. (3.5)) and PA (Proportional Allocation, see Eq. (3.6)), and two global allocation schemes – EP (Equal Partition) and NP (Normalised Proportional), were proposed and analysed by Agrawal et al [3] in terms of their ability to satisfy the deadline constraint for a stream of synchronous messages with deadlines equal to periods. The WCAU metric has been adopted as a means to compare and evaluate different schemes. It became apparent that FLA and PA schemes present a weak guarantee ability and may fail to guarantee many synchronous message sets in practice due to their WCAU being as small as 0%. However, the WCAU of the NP scheme can achieve 33%, which is the highest of the four schemes analysed. Later, Agrawal et al [2] also proposed a local SBA scheme (LA_A scheme, see Eq. (3.7)) - for guaranteeing synchronous message sets with deadlines equal to periods. However, they showed that their new scheme can achieve the WCAU of 33% which can work much better than their previously reported local schemes FLA and PA. An optimal global SBA scheme was first proposed by Chen et al [12]. Later, Han et al [29] also proposed an optimal global SBA scheme with a polynomial-time worst-case complexity. These global schemes can only apply to message sets with deadlines equal to periods. Malcolm et al [56] generalised the local scheme proposed by Agrawal et al [2] and proposed a local SBA scheme (LA_M scheme, see Eq. (3.8)) for use in a general synchronous message set with arbitrary deadline constraints. This scheme was adopted by SAFENET because of its good performance with the WCAU of 33% and its applicability to general message sets[65]. Another local SBA scheme (LA_Z scheme, see Eq. (3.9)) for guaranteeing synchronous messages with arbitrary deadline constraints is developed by Zheng et al in [96] where the proposed local scheme is claimed to be optimal in some cases including the case of message sets with message periods equal to deadlines. However, the overall performance of this scheme is similar to that of LA_M. For the cases of deadlines being equal to periods, LA_M performs identically to LA_A.

Before Zhang and Burns [90] first pointed out that the message sets with $TTRT < D_{min} < 2 \cdot TTRT$ may also be schedulable with a proper setting of protocol parameters [91], none of the previously published schemes considered the message sets with $TTRT < D_{min} < 2 \cdot TTRT$. Zhang et al [90, 92, 94] also proposed a few enhanced/novel schemes applicable to message sets with $TTRT < D_{min} < 2 \cdot TTRT$. Although their proposed schemes can only be used for message sets with deadlines no more than periods, they generally perform very well with best achievable performance due to the adoption of most accurate results on protocol timing properties. The proposed NLA (New Local Allocation, see Eq. (3.10)) scheme showed the fact that the scheme proposed by Zheng et al is actually non-optimal for message sets with
periods equal to deadlines because the NLA scheme can guarantee message sets which cannot be guaranteed by LA scheme. Although the WCAU of the NLA scheme is zero (0%) due to its applicability to a message set with $TTRT < D_{\text{min}} < 2 \cdot TTRT$, it presents a larger worst-case utilisation value than that of LA (by taking into account of the actual number of nodes in the network) when NLA is limited for use with a message set with $D_{\text{min}} \geq 2 \cdot TTRT$.

- **Full Length Allocation (FLA) scheme** [3]:
  \[
  H_i = C_i
  \]  
  (3.5)

- **Proportional Allocation (PA) scheme** [3]:
  \[
  H_i = \frac{C_i}{P_i} \cdot (TTRT - \tau)
  \]  
  (3.6)

- **Local Allocation (LA) scheme proposed by Agrawal et al** [2]:
  \[
  H_i = \frac{C_i}{\lfloor \frac{P_i}{TTRT} \rfloor} - 1
  \]  
  (3.7)

- **Local Allocation (LM) scheme proposed by Malcolm et al** [56]:
  \[
  H_i = \max\left(\frac{TTRT}{P_i}, 1\right) \cdot C_i
  \]  
  (3.8)

- **Local Allocation (LAZ) scheme proposed by Zheng et al** [96]:
  \[
  H_i = a_i \cdot C_i
  \]  
  (3.9)

where $a_i$ is determined as below: let $x = \frac{D_i}{TTRT} - 1$ and $y = \frac{P_i}{TTRT}$, then,

\[
a_i = \begin{cases} 
  \frac{1}{|x|} & \text{if } y \geq |x| \geq 1 \\
  \frac{1}{y} & \text{if } y \leq 1 \text{ and } x \geq 2 
\end{cases}
\]

and

\[
a_i \leq \begin{cases} 
  1 + \frac{2-x}{y} & \text{if } y \leq 1 \text{ and } 1 \leq x < 2 \\
  \frac{1}{y} & \text{if } 1 < y < |x| 
\end{cases}
\]

- **New Local Allocation (NLA) scheme proposed by Zhang et al** [94]:
  \[
  H_i = \begin{cases} 
  C_i & \text{if } TTRT < D_i < 2 \cdot TTRT \\
  \frac{C_i}{q_i - 1} - \frac{1}{q_i} \max\{D_i - [(q_i + 1)TTRT - \frac{C_i}{q_i - 1}], 0\} & \text{if } D_i \geq 2 \cdot TTRT
\end{cases}
  \]  
  (3.10)

where $q_i = \lfloor \frac{D_i}{TTRT} \rfloor$

Table 3.1 describes six published local SBA schemes in more details.
Table 3.1: Six published local SBA schemes

<table>
<thead>
<tr>
<th>name</th>
<th>expression ((H_i =))</th>
<th>usage</th>
<th>descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLA</td>
<td>(C_i)</td>
<td>(P_i = D_i)</td>
<td>This scheme tries to allocate synchronous bandwidth as large as (C_i) in order for the whole message to be sent when the node receives the token without waiting for the next token arrival.</td>
</tr>
<tr>
<td>PA</td>
<td>(\frac{C_i}{P_i} \cdot (TTRT - \tau))</td>
<td>(P_i = D_i \geq 2 \cdot TTRT)</td>
<td>This scheme distributes the available portion of the token rotation time among the nodes according to the ratio of message lengths (C_i) to message periods (P_i).</td>
</tr>
<tr>
<td>LA(_A)</td>
<td>(\frac{C_i}{\lceil\frac{TTRT}{P_i}\rceil - 1})</td>
<td>(P_i = D_i \geq 2 \cdot TTRT)</td>
<td>The scheme’s expression shows its very reasonable allocation. In the worst case, the token can visit node (i) at least (\lceil\frac{P_i}{TTRT}\rceil - 1) times during the time interval of (P_i). Thus, in order to satisfy the deadline constraints, the required transmission time will be (\frac{C_i}{\lceil\frac{TTRT}{P_i}\rceil - 1}) every time when the token arrives at node (i).</td>
</tr>
<tr>
<td>LA(_M)</td>
<td>(\max\left(\frac{a_i \cdot C_i}{P_i - 1}\right))</td>
<td>(D_i \geq 2 \cdot TTRT)</td>
<td>This scheme is a generalised version of LA(_A), which can be used for a message set with arbitrary deadline constraints under (D_i \geq 2 \cdot TTRT).</td>
</tr>
<tr>
<td>LA(_Z)</td>
<td>(a_i \cdot C_i)</td>
<td>(D_i \geq 2 \cdot TTRT)</td>
<td>Apart from different expressions, both LA(_Z) and LA(_M) perform similarly.</td>
</tr>
<tr>
<td>NLA</td>
<td>[\begin{align*} &amp; C_i \quad \text{if } TTRT &lt; D_i &lt; 2 \cdot TTRT \ &amp; \frac{C_i}{q_{i-1}} - \frac{1}{q_i} \max{D_i - [(q_i + 1)TTRT - \frac{C_i}{q_{i-1}}], 0} \quad \text{if } D_i \geq 2 \cdot TTRT \end{align*}]</td>
<td>(P_i = D_i &gt; TTRT)</td>
<td>This scheme is the very first local SBA scheme that considers a synchronous message set with (D_i &gt; TTRT).</td>
</tr>
</tbody>
</table>
Table 3.2: The basic categories of SBA schemes

<table>
<thead>
<tr>
<th></th>
<th>global SBA schemes</th>
<th>local SBA schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>deadline ≥ 2 \cdot TTRT</td>
<td>deadline &gt; TTRT</td>
<td></td>
</tr>
<tr>
<td>deadline ≤ periods</td>
<td>arbitrary deadline constraints</td>
<td></td>
</tr>
</tbody>
</table>

SBA schemes can be categorised into six basic groups, as depicted in Table 3.2. Any of the above reviewed SBA schemes is the combination of three groups, which are chosen from one of the two columns in each row of the table. The SBA scheme proposed in this thesis is a local SBA scheme which can be used for any general synchronous message set of arbitrary deadline constraints with \( D_{\min} > TTRT \).

3.4 Protocol Timing Properties

One important timing property of the timed-token MAC protocol is its bounded token rotation time. The study on timing properties of the timed-token protocol has mainly focused on deriving an upper bound on the possible time elapsed during any given number of successive token rotations or a lower bound on the minimum available transmission time during any given length of time span. Let \( t_{l,i} \) be the token’s \( l \)-th arrival time at node \( i \). The following theorems give an upper bound on the time possibly elapsed during any one token rotation (Theorem 1) and any \((v-1)\) token rotations (Theorems 2 and 3).

**Theorem 1.** (Johnson and Sevcik’s Theorem [43, 68]): *For any integer \( l > 0 \) and any node \( i (1 \leq i \leq n) \), under the protocol constraint,*

\[
t_{l+1,i} - t_{l,i} \leq TTRT + \sum_{h=1,\ldots,n, \ h \neq i} H_h + \tau \leq 2 \cdot TTRT
\]  (3.11)

**Theorem 2.** (Generalised Johnson and Sevcik’s Theorem by Chen et al [12]): *For any integer \( l > 0, v \geq 2 \) and any node \( i (1 \leq i \leq n) \), under the protocol constraint,*

\[
t_{l+v-1,i} - t_{l,i} \leq (v-1) \cdot TTRT + \sum_{h=1,\ldots,n, \ h \neq i} H_h + \tau \leq v \cdot TTRT - H_i
\]  (3.12)

**Theorem 3.** (Generalised Johnson and Sevcik’s Theorem by Zhang and Burns [89]): *For any integer \( l > 0, v \geq 2 \) and any node \( i (1 \leq i \leq n) \), under the protocol constraint,*

\[
t_{l+v-1,i} - t_{l,i} \leq (v-1) \cdot TTRT + \sum_{h=1,\ldots,n, \ h \neq i} H_h + \tau - \left\lfloor \frac{v-1}{n+1} \right\rfloor \cdot \left( TTRT - \sum_{h=1}^{n} H_h - \tau \right)
\]  (3.13)

In the timed-token MAC protocol, a node can only send synchronous messages when it holds the token. The bounded timing property, as shown in Theorems 1, 2 and 3, provides the
necessary condition for synchronous messages to meet their deadline constraints because a node can have chances to capture and hold the token before the message deadline expires if network parameters are appropriately selected.

Theorem 1 shows that the time elapsed between any two consecutive token arrivals at the same node is bounded by $TTRT + \sum_{h=1,\ldots,n, \ h \neq i} H_h + \tau$. This is the special case of Theorem 2 when $v = 2$. Theorem 2 gives an upper bound on the possible time elapsed between any $v$ consecutive token arrivals at the same node. This result has been extensively used by Zhao et al [2, 3, 12, 56] in their studies and analysis of various SBA schemes. However, this result is not good enough for guaranteeing hard real-time traffic in the sense that the bound derived is not tight. Theorem 3 is another generalised version of Johnson and Sevcik's theorem, which is tighter than Theorem 2.

Compared with the upper bound derived by Chen et al [12], this bound is tighter when $v$ is no less than $n + 2$ under the condition of $\sum_{h=1}^{n} H_h < TTTR - \tau$. Actually, the condition $\sum_{h=1}^{n} H_h < TTTR - \tau$ is important. When allocating synchronous bandwidth for a given synchronous message set, the full allocation of $\sum_{h=1}^{n} H_h = TTTR - \tau$ is not always best and may even lead to no feasible allocations for some synchronous message sets to be guaranteed.

Getting a tighter upper bound of the token’s consecutive arrivals at a node is important to the further work on, for example, the development of synchronous bandwidth allocation schemes. Since a tighter upper bound means possibly more chances for a token to arrive at a certain node in a given time interval, and thus, the node may have more chances to transmit synchronous messages using its allocated synchronous bandwidth. This may lead to a smaller amount of synchronous bandwidth allocated to the nodes, and thus greater possibility to meet the protocol constraint while the deadline constraint remains satisfied.

All the above timing properties studied the bounded time elapsed between the token’s consecutive visits to the same node. Zhang et al also published [93] a generalised cycle time property of the timed-token MAC protocol, which gives an upper bound on the time possibly elapsed between the token’s $l$-th arrival at any node $i$ and the token’s $(l + v)$-th arrival at any node $k$ (where $v$ is a non-negative integer). However, when this timing property is used to study a local SBA scheme, it becomes the same as the timing property defined in Theorem 3.

The further study carried in this research is based on the best and exact timing property shown in Theorem 3. Adding $H_i$ to the both sides of (3.13), there is

$$t_{l+v-1,i} - t_{l,i} + H_i \leq (v - 1) \cdot TTTR + \sum_{h=1}^{h=n} H_h + \tau - \frac{v - 1}{n + 1} \cdot [TTTR - \sum_{h=1}^{n} H_h - \tau] \quad (3.14)$$

The value of $t_{l+v-1,i} - t_{l,i} + H_i$ represents the time elapsed before node $i$ uses up its next $(v - 1)$ allocated synchronous bandwidths ($H_i$). The above expression of (3.14) shows an upper
bound of this. Let \( I(v - 1) \) denote the upper bound, then

\[
I(v - 1) = (v - 1) \cdot TTRT + \sum_{h=1}^{\frac{v}{n} - 1} H_h + \tau - \left[ \frac{v - 1}{n + 1} \right] \cdot [TTRT - \sum_{h=1}^{n} H_h - \tau]
\]

and the following theorem is obvious.

**Theorem 4.** Let \( I(v) \) be the tight upper bound on the (maximum) time which could possibly elapse in the worst case before a node uses up its next \( v \) (where \( v \) is an integer no less than one) allocated synchronous bandwidths \( (H_i) \), then under the protocol constraint,

\[
I(v) = v \cdot TTRT + \sum_{k=1}^{n} H_h + \tau - \left[ \frac{v}{n + 1} \right] \cdot [TTRT - \left( \sum_{h=1}^{n} H_h + \tau \right)] \tag{3.15}
\]

As a special case of Theorem 4 when \( v = 1 \), there exists the following corollary.

**Corollary 1.** Under the protocol constraint, the maximum time possibly elapsed in the worst case, from the time node \( i \) starts waiting for its next turn to use its allocated synchronous bandwidth \( (H_i) \) till the time node \( i \) uses up its allocated synchronous bandwidth \( (H_i) \), is bounded by

\[
I(1) = TTRT + \sum_{h=1}^{n} H_h + \tau \tag{3.16}
\]

Also, when considering the minimum available transmission time during any given length of time span, the following corollary can be obtained.

**Corollary 2.** Under the protocol constraint, the minimum available time for node \( i \) to transmit synchronous messages during any time interval \( T \), when judged only on local information of node \( i \), is

\[
x_i(T) = (\lfloor \frac{T}{TTRT} \rfloor - 1) \cdot H_i + \max[0, T + H_i - (\lfloor \frac{T}{TTRT} \rfloor + 1)TTRT] \tag{3.17}
\]

The proof of Corollary 2 can be found in [94]. Theorem 4, Corollary 1 and Corollary 2 will be frequently used in the following chapter to derive the new local SBA scheme.

### 3.5 Summary

Necessary preliminaries, including network and message models, the timed-token MAC protocol, SBA schemes, and timing properties, have been introduced in this chapter. These preliminaries, together with some assumptions, such as error free network and one synchronous message stream for every node, establish the basic framework for this research.
There are many works concentrating on different aspects of the timed-token MAC protocol. Some main aspects include: (1) Timing property. This property is the basis of other work such as SBA schemes. The research presented here is based on the best timing property proposed by Zhang et al [93]. (2) Schedulability analysis. The schedulability analysis is usually based on different SBA schemes to test if a synchronous message set can be guaranteed by an SBA scheme. However, Lu et al [98] studied the different queueing models used for scheduling messages. Zhang et al [95] found that the orders of nodes was important for scheduling the real-time messages. (3) SBA schemes. As reviewed in Section 3.3.4, SBA schemes have been studied by many researchers. Works on SBA schemes can be categorised into two classes: effective local SBA schemes and optimal global SBA schemes. Although many local SBA schemes have been published, a new local SBA scheme, which performs better than any others in terms of guaranteeing hard real-time traffic, can be developed by using the best timing property. Compared with local SBA schemes, only a few global SBA schemes are published and none of them is really optimal. (4) Asynchronous message transmission. It is important to improve and even maximise the throughput of asynchronous messages while the transmission of the synchronous traffic is guaranteed. There were only a few works which tried to address this problem, as reviewed in Section 2.2.1. (5) Modification of the timed-token MAC protocol. Cobb et al [16] proposed the on-time timed-token protocol which adds additional information to the token to remove the token-lateness problem of the timed-token protocol. The BuST protocol designed by Franchino et al [22] improves the performance of the timed-token protocol by reducing the worst-case token rotation time to $TTRT$. (6) Usage of the timed-token MAC protocol. Some work has concentrated on how to use the timed-token MAC protocol to support real-time traffic. Section 2.3 has reviewed such use of the timed-token MAC protocol.

This research tries to address the problem of local SBA schemes. A new local scheme, which can outperform any other reviewed local SBA scheme, is developed in the next chapter.
Chapter 4

A Generalised Synchronous Bandwidth Allocation Scheme

In this chapter, a new local SBA scheme is proposed, based on the best result on the protocol timing property as described in Theorem 4. The result can be used to guarantee a synchronous message set with arbitrary deadline constraints and its minimum deadline larger than $TTRT$. The proposed scheme performs no worse than any previously published local SBA scheme. Based on whether or not the deadline is smaller than $2 \cdot TTRT$, two separate cases will be considered in the following discussion.

This chapter is organised as follows: Section 4.1 analyses the bandwidth allocation to those message streams whose deadlines are less than $2 \cdot TTRT$. The bandwidths allocated to the message streams with deadlines no less than $2 \cdot TTRT$ are then discussed in Section 4.2, using a different method than that used in Section 4.1. Section 4.3 will propose a new local SBA scheme, based on the results of the previous two sections, together with the schedulability analysis of this new local scheme. At the end of this section, this new SBA scheme will be compared with other previous published local SBA schemes to show its superiority.

Some notations that are frequently used in this chapter are first introduced below.

For $i = 1, 2, \cdots, n$, let $q_i = \lfloor \frac{D_i}{TTRT} \rfloor$, then $D_i$ can be expressed as $D_i = q_i \cdot TTRT + r_{D_i}$, where $0 \leq r_{D_i} < TTRT$. That is, $r_{D_i} = D_i - \lfloor \frac{D_i}{TTRT} \rfloor TTRT$. Similarly, $r_{P_i}$ can be expressed as $r_{P_i} = P_i - \lfloor \frac{P_i}{TTRT} \rfloor TTRT$.

$D_{min}$ is a lower bound on message deadlines. That is, for $i = 1, 2, \cdots, n$, $D_{min} \leq D_i$.

---

1 According to Corollary 1, the restriction of $D_{min} > TTRT$ is necessary for any message set to be guaranteed; or else, if $D_{min} \leq TTRT$, the node with $D_{min}$ cannot receive the token during $D_{min}$ even once, in the worst case.
Finally, $X_i(D_i)$ is the minimum available time for node $i$ to transmit its synchronous messages during the time interval $D_i$, while $x_i(D_i)$ is the minimum available synchronous message transmitting time for node $i$ during $D_i$ based only on node $i$’s local information.

4.1 Bandwidth Allocation to Message Streams with their Deadlines less than $2 \cdot TTRT$

This section discusses how to allocate the synchronous bandwidth to each synchronous message stream $S_i$ with $TTRT < D_i < 2 \cdot TTRT$ such that the deadline constraint can be satisfied\(^2\).

According to Corollary 1, in the worst case, under the case when $D_i < 2 \cdot TTRT$, at most one $H_i$ can be used by node $i$. To guarantee the message deadline, node $i$ should make use of $H_i$ for at least one time during the time interval $D_i$. According to Corollary 1, the minimum required time interval to ensure that a node can use up its allocated synchronous bandwidth once, is $TTRT + \sum_{h=1}^{n} H_h + \tau$. Thus, for any node $i$, in order to guarantee the deadline, there is

$$D_i \geq TTRT + \sum_{h=1}^{n} H_h + \tau > TTRT \quad (4.1)$$

Let $D_{min}$ be the minimum deadline of the synchronous message stream of a message set, i.e., $D_{min} = \min\{D_1, D_2, \cdots, D_n\}$. Then, (4.1) can be restated as

$$D_{min} \geq TTRT + \sum_{h=1}^{n} H_h + \tau > TTRT \quad (4.2)$$

That is

$$\sum_{h=1}^{n} H_h \leq D_{min} - TTRT - \tau \quad (4.3)$$

The above inequality (4.3) can be treated as a necessary condition for guaranteeing the deadlines of a synchronous message set. The violation of (4.3) will lead to missed deadline at least for the message stream with $D_{min}$. From (4.3), because $D_{min} < 2 \cdot TTRT$, the following inequality can be obtained.

$$\sum_{h=1}^{n} H_h \leq D_{min} - TTRT - \tau \quad < 2 \cdot TTRT - TTRT - \tau = TTRT - \tau$$

This inequality implies that as long as the inequality (4.3) holds, the protocol constraint (3.2) can be automatically satisfied. Thus, there is no need to check the protocol constraint separately for any schedulable synchronous message set with $D_{min} < 2 \cdot TTRT$.

\(^2\)This work has been published in the paper [87].
In the following discussion, three separate cases will be considered when allocating bandwidth to node $i$.

### 4.1.1 Message Streams with $D_i \leq P_i$

As discussed above, for the message stream $S_i$ with $D_i \leq P_i$, at most one $H_i$ can be used up within the time interval of $D_i$ in the worst case. According to (3.4), in order to guarantee the message deadline constraint, the synchronous bandwidth for node $i$ has to be allocated such that

$$H_i \geq C_i$$  \hspace{1cm} (4.4)

Because allocating more synchronous bandwidth than actually required makes no sense for meeting the message deadline but in contrast runs the risk of violating the protocol constraint, $H_i$ should be the minimum value which satisfy the above inequality, that is

$$H_i = C_i$$ \hspace{1cm} (4.5)

It is easy to check that under (4.3) and (4.5), any message from the stream $S_i$ with $D_i \leq P_i$ can be transmitted before its deadline. That is, for this case, the bandwidth allocated to $S_i$ based on (4.5) can meet the deadline constraint under (4.3).

### 4.1.2 Message Streams with $TTRT \leq P_i < D_i$

Assume that the first message from $S_i$ arrives on node $i$ at time $t_{i,0}$, then, the $k$-th ($k \geq 1$) message arrives at $t_{i,0} + (k-1) \cdot P_i$ and must be transmitted before $t_{i,0} + (k-1) \cdot P_i + D_i$ in order to meet its deadline. As all outgoing messages on node $i$ are queued in FIFO order, to meet the deadline, available bandwidth during the time interval $[t_{i,0}, t_{i,0} + (k-1) \cdot P_i + D_i]$ should be adequate enough for transmitting these $k$ messages. Let $X_i^k$ be the minimum time available for transmitting synchronous messages during $(k-1) \cdot P_i + D_i$, then, to guarantee the deadline, the following condition must be met.

$$X_i^k \geq k \cdot C_i$$ \hspace{1cm} (4.6)

According to Theorem 4,

$$I(v) = v \cdot TTRT + \sum_{k=1}^{n} H_h + \tau - \left\lfloor \frac{v}{n+1} \right\rfloor \cdot [TTRT - \left( \sum_{h=1}^{n} H_h + \tau \right)]$$

$$\leq v \cdot TTRT + \sum_{k=1}^{n} H_h + \tau$$ \hspace{1cm} (4.7)
Figure 4.1: The worst case situation when $TTRT \leq P_i < D_i$

Assume that during the time interval $(k-1) \cdot P_i + D_i$, node $i$ can use up $H_i$ at least $u$ times, then according to (4.7), this assumption can hold under the following condition

$$u \cdot TTTRT + \sum_{h=1}^{n} H_h + \tau \leq (k-1) \cdot P_i + D_i < (u+1) \cdot TTTRT + \sum_{h=1}^{n} H_h + \tau$$

From this condition, the result of $u$ is

$$u = \left\lfloor \frac{(k-1) \cdot P_i + D_i - \sum_{h=1}^{n} H_h - \tau}{TTTRT} \right\rfloor \tag{4.8}$$

Thus

$$X_i^k \geq u \cdot H_i$$

$$= \left\lfloor \frac{(k-1) \cdot P_i + D_i - \sum_{h=1}^{n} H_h - \tau}{TTTRT} \right\rfloor \cdot H_i \quad \text{(by (4.8))}$$

$$\geq \left\lfloor \frac{(k-1) \cdot P_i + TTTRT}{TTTRT} \right\rfloor \cdot H_i \quad \text{ (since (4.1))}$$

$$\geq \left\lfloor \frac{(k-1) \cdot TTTRT + TTTRT}{TTTRT} \right\rfloor \cdot H_i \quad \text{ (since $P_i \geq TTTRT$)}$$

$$= k \cdot H_i \tag{4.9}$$

From (4.9), (4.6) holds under

$$H_i = C_i \tag{4.10}$$

This result can also be derived from observation of Fig. 4.1, which shows the worst case situation under this case. Node $i$ has to wait for

$$t_{first} - t_0 - H_i = TTTRT + \sum_{h=1,\ldots,n,h\neq i} H_h + \tau$$

units of time in the worst case situation before it gets the first chance to use its synchronous bandwidth. It is easy to check that before the deadline of the first message, there is at most one chance to use up $H_i$. In order to guarantee the deadline, $H_i$ should be allocated no less than $C_i$. After the first message, every subsequent message can get a chance to use up at least...
Figure 4.2: The worst case situation when $0 < P_i < TTRT < D_i$

one $H_i$ before its deadline because after the time $t_{first}$, for every incremented time interval of $TTRT$, there is one more chance for node $i$ to use its synchronous bandwidth. Thus, the deadline constraint of the message stream $S_i$ can be satisfied when $H_i = C_i$.

### 4.1.3 Message Streams with $0 < P_i < TTRT < D_i$

Fig. 4.2 shows the worst case situation for this case. It is assumed that in the worst case situation, the $m$-th message is the last message which must be transmitted by sharing the allocated $H_i$ during the token’s first visit since $t_0$ if its deadline is not to be missed. That is, the deadline of the $m$-th message will miss if it is transmitted when node $i$ captures its next (second) token. That is, the $m$-th message has a deadline before the next $H_i$ becomes available. In order to meet the deadline constraint, enough synchronous bandwidth must be allocated for transmitting these first $m$ messages. The deadline constraint can certainly met if the synchronous bandwidth is allocated such that

$$H_i \geq m \cdot C_i \quad (4.11)$$

Since the absolute deadline of the $m$-th message is $t_0 + (m - 1) \cdot P_i + D_i$ and the second $H_i$ becomes available after $t_0 + (TTRT + \sum_{h=1}^{n} H_h + \tau) + TTRT - H_i$, there is

$$t_0 + (m - 1) \cdot P_i + D_i \leq t_0 + (TTRT + \sum_{h=1}^{n} H_h + \tau) + TTRT - H_i \quad (4.12)$$

Obviously, the earlier deadline is more difficult to satisfy. That is, in the worst case situation, only the earliest deadline needs to be considered. From (4.1) and (4.12), there is

$$(m - 1) \cdot P_i \leq TTRT - H_i \quad (4.13)$$

That is

$$m \leq \frac{TTRT - H_i}{P_i} + 1 \quad (4.14)$$

Because

$$\left\lceil \frac{TTRT - H_i}{P_i} \right\rceil + 1 \geq \frac{TTRT - H_i}{P_i} + 1 \geq m$$

50
In order to meet (4.11), \( H_i \) must be

\[
H_i \geq \left( \left\lceil \frac{TTRT - H_i}{P_i} \right\rceil + 1 \right) \cdot C_i \\
\geq \left( \frac{TTRT - H_i}{P_i} + 1 \right) \cdot C_i \\
\geq m \cdot C_i \tag{4.16}
\]

Since allocating more synchronous bandwidth than actually required reduces the chances of meeting the protocol constraint, the smallest \( H_i \) which satisfies the following inequality needs to be found.

\[
H_i \geq \left( \left\lceil \frac{TTRT}{P_i} \right\rceil + 1 \right) \cdot C_i \tag{4.17}
\]

\( H_i \) cannot be simply calculated from the above inequality because \( H_i \) sits on both sides of (4.17) and (4.17) is discontinues in terms of \( H_i \). The following two methods are proposed to obtain the minimum solution of \( H_i \). The first one is inaccurate but fast, and the second one is accurate but slow. The accuracy of the result is only from the perspective of satisfying (4.17). Chapter 6 will compare these two results in a more detailed way.

**Fast solution**

From (4.17), a simple solution can be approximated as

\[
H_i = \left( \frac{TTRT}{P_i} \right) + 1 \cdot C_i \tag{4.18}
\]

and the following theorem can be obtained.

**Theorem 5.** If the synchronous bandwidth is allocated according to (4.18) for a synchronous message stream \( S_i \) with \( P_i < TTRT < D_i < 2 \cdot TTRT \), then the deadline constraint is satisfied under

\[
TTRT + \sum_{h=1}^{n} H_h + \tau \leq D_i < 2 \cdot TTRT.
\]

A formal proof of this theorem can be found in Appendix A.

**Accurate solution**

It is first shown that there exists an accurate minimum solution \( H_{i_{\text{min}}} \) of inequality (4.17), then an algorithm is proposed to find such \( H_{i_{\text{min}}} \).

**Theorem 6.** Let \( f(H_i) = H_i - \left( \left\lceil \frac{TTRT - H_i}{P_i} \right\rceil + 1 \right) \cdot C_i \), then

(a) \( f(H_i) \) is an increasing function of \( H_i \);
Corollary 3. Inequality (4.17) has an accurate minimum solution, obvious result of Theorem 6.

(b) If \( \frac{TTRT - TTRT - P_i}{P_i + C_i} \in \mathbb{Z} \), then there is a minimum solution \( H_{\text{min}} = \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \) for \( f(H_i) \geq 0 \);

(c) If \( \frac{TTRT - TTRT + P_i}{P_i + C_i} \notin \mathbb{Z} \), then there is one minimum solution \( H_{\text{min}} \) for \( f(H_i) \geq 0 \), subject to \( \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \leq H_{\text{min}} < \frac{TTRT + 2P_i}{P_i + C_i} \cdot C_i \); where \( f\left(\frac{TTRT + P_i}{P_i + C_i} \cdot C_i\right) \leq 0 \) and \( f\left(\frac{TTRT + 2P_i}{P_i + C_i} \cdot C_i\right) > 0 \).

A formal proof of Theorem 6 can be found in Appendix B. The following corollary is an obvious result of Theorem 6.

**Corollary 3.** Inequality (4.17) has an accurate minimum solution \( H_{\text{min}} \), where

\[
\frac{TTRT + P_i}{P_i + C_i} \cdot C_i \leq H_{\text{min}} < \frac{TTRT + 2P_i}{P_i + C_i} \cdot C_i
\]

and

\[
f\left(\frac{TTRT + P_i}{P_i + C_i} \cdot C_i\right) \leq 0; f\left(\frac{TTRT + 2P_i}{P_i + C_i} \cdot C_i\right) > 0.
\]

From Theorem 6 and Corollary 3, \( H_i \) can be obtained based on the following two conditions.

- if \( f(H_1) = 0 \), the minimum solution \( H_{\text{min}} \) of (4.17) is \( H_1 \); and
- if \( f(H_1) < 0 \) and \( f(H_2) \geq 0 \), \( H_{\text{min}} \) must belong to the range \( [H_1, H_2] \).

Thus, the basic idea of the algorithm to implement the method is to find the range \( [H_1, H_2] \), which has the following property: \( f(H_1) = 0 \) or the length of \( [H_1, H_2] \), i.e., \( H_2 - H_1 \), tends to 0 while \( f(H_1) < 0 \). Corollary 3 gives the initial value of \( [H_1, H_2] \), i.e., \( \left[\frac{TTRT + P_i}{P_i + C_i} \cdot C_i, \frac{TTRT + 2P_i}{P_i + C_i} \cdot C_i\right] \).

The binary-search method can be used to shorten the length of \( [H_1, H_2] \) due to \( f(H_i) \) being the increasing function of \( H_i \). The following steps are obvious:

- Step 1: Initialising \( H_1 \) and \( H_2 \) by \( \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \) and \( \frac{TTRT + 2P_i}{P_i + C_i} \cdot C_i \) respectively and then calculating \( f(H_1) \) and \( f(H_2) \).

- Step 2: Looping the following sub-steps until \( f(H_1) \geq 0 \) or the length of \( [H_1, H_2] \) tends to zero (i.e., \( H_2 - H_1 < \varepsilon \)).

1. Calculating the middle point \( H_0 \) of \( [H_1, H_2] \) and \( f(H_0) \) where \( H_0 = \frac{H_1 + H_2}{2} \).

2. If \( f(H_0) \geq 0 \), let \( H_0 \) be the new value of \( H_2 \). Otherwise, let \( H_0 \) be the new value of \( H_1 \). After this, the length of the updated range of the new \( [H_1, H_2] \) is shortened to half of the old one and \( H_{\text{min}} \) falls into this new range.

---

3 Because of the accuracy of the CPU, \( \varepsilon \) is used to represent a value tiny enough to approach zero.

4 It is necessary to use \( f(H_1) \geq 0 \) instead of \( f(H_1) = 0 \) due to the accuracy of CPU.
Algorithm 1 Get\_H (INPUT: \(TTRT, P_i, C_i\), OUTPUT: \(H_{\text{min}}^i\))

1: \(H_1 \leftarrow \frac{TTRT \times P_i}{P_i + C_i} \cdot C_i, H_2 \leftarrow \frac{TTRT + 2 \times P_i}{P_i + C_i} \cdot C_i, f_1 \leftarrow f(H_1), f_2 \leftarrow f(H_2)\)

2: while \(H_2 - H_1 \geq \varepsilon\) do

3: if \(f_1 < 0\) and \(f_2 \geq 0\) then

4: \(H_0 \leftarrow \frac{H_1 + H_2}{2}\)

5: if \(f(H_0) \geq 0\) then

6: \(H_2 \leftarrow H_0, f_2 \leftarrow f(H_0)\)

7: else

8: \(H_1 \leftarrow H_0, f_1 \leftarrow f(H_0)\)

9: end if

10: else

11: BREAK WHILE

12: end if

13: end while

14: if \(f_1 \geq 0\) then

15: \(H_{\text{min}}^i \leftarrow H_1\)

16: else

17: \(H_{\text{min}}^i \leftarrow H_2\)

18: end if

19: return \(H_{\text{min}}^i\)

Figure 4.3: Pseudo code of algorithm Get\_H

- Step 3: After Step 2, the targeted range of \([H_1, H_2]\) is found. Because the length of this range is small enough, the value of \(H_{\text{min}}^i\) should be \(H_1\) when \(f(H_1) \geq 0\) or \(H_2\) when \(f(H_1) < 0\).

The above discussion also shows that the accuracy of this algorithm depends on the value of \(\varepsilon\). A smaller value of \(\varepsilon\) leads to a shorter \([H_1, H_2]\) and also brings a more accurate \(H_{\text{min}}^i\). However, the smaller \(\varepsilon\) could cause a longer running time of the algorithm. A suggested choice of the value of \(\varepsilon\) is one tenth of the required accuracy of \(H_{\text{min}}^i\).

The pseudo code of this algorithm, Get\_H, listed in Fig. 4.3, can be easily formed from the above analysis.

The Complexities of the Algorithm Get\_H The time and space complexities of the algorithm Get\_H must be examined for using it in practice.
1. **The time complexity**

To derive the time complexity of the algorithm \textbf{Get.H} in an asymptotic sense, it is only need to consider the \textbf{WHILE} statement here (i.e., for how many times the \textbf{WHILE} part will be executed). In the worst case, the \textbf{WHILE} statement will be executed until the difference between \( H_1 \) and \( H_2 \) is less than \( \varepsilon \). Since the value of \( H_2 - H_1 \) is halved for each iteration, after \( m \) times execution, the distance between \( H_2 - H_1 \) is given by

\[
I_m = \frac{H_2 - H_1}{2^m}
\]

Therefore, the terminating condition of the \textbf{WHILE} part can be expressed as

\[
I_m < \varepsilon
\]

Thus

\[
m > \log_2\left(\frac{H_2 - H_1}{\varepsilon}\right)
\]

Based on the above analysis, the time complexity of algorithm \textbf{Get.H} in an asymptotic sense is \( T(N) = O(\log_2 N) \) where \( N = \frac{H_2 - H_1}{\varepsilon}, H_1 = \frac{TT+RT+P_i}{P_i+C_i} \cdot C_i \) and \( H_2 = \frac{TT+RT+2 \cdot P_i}{P_i+C_i} \cdot C_i \).

The timing complexity is logarithmic and based on the value of \( N \). Since

\[
N = \frac{H_2 - H_1}{\varepsilon} = \frac{TT+RT+2 \cdot P_i}{P_i+C_i} \cdot C_i - \frac{TT+RT+P_i}{P_i+C_i} \cdot C_i
\]

\[
= \frac{P_i \cdot C_i}{P_i+C_i} \cdot \varepsilon
\]

\[
< \frac{C_i}{\varepsilon},
\]

the value of \( N \) is upper bounded by \( \frac{C_i}{\varepsilon} \). That is, for a given \( C_i \), a smaller \( \varepsilon \) will lead to a bigger \( N \), making the algorithm demand more time for execution. This should be obvious, since a smaller \( \varepsilon \) means a high requirement of accuracy on the result, which obviously demands more time. However, in practice, \( \varepsilon \) does not need to be as small as possible. The choice of \( \varepsilon \) is based on the accuracy of \( H_i \). When the accuracy of \( H_i \) is 0.01, \( \varepsilon \) only needs to be 0.001. Hence, the final time requirement should not be large since the number of iterations of the \textbf{WHILE} loop is only \( \log_2 N \).

2. **The space complexity**

At all times the algorithm \textbf{Get.H} only needs to remember five values: \( H_1, H_2, H_0, f_1 \) and \( f_2 \). Therefore, the space complexity is \( S(N) = O(1) \).

The above analysis shows that the time complexity of \textbf{Get.H} is logarithmic and the space complexity is constant. Thus, the algorithm can be implemented and used efficiently in practice.
4.1.4 Summary

Summarising the above analysis, for any synchronous message stream $S_i$ with $TTRT < D_i < 2 \cdot TTTR$, the bandwidth can be allocated as follows:

$$H_i = \begin{cases} C_i & \text{if } D_i \leq P_i \text{ or } TTTR \leq P_i < D_i \\ (\frac{TTRT}{P_i} + 1)C_i \text{ or a return from Get}_H & \text{if } P_i < TTTR < D_i \end{cases}$$ (4.19)

and the following theorem is obvious.

**Theorem 7.** When $TTTR < D_{\min} < 2 \cdot TTTR$, under the condition of

$$\sum_{h=1}^{n} H_h \leq D_{\min} - TTTR - \tau$$

the protocol constraint can be satisfied, and the deadline constraint can also be satisfied for any stream $S_i$ with $TTTR < D_i < 2 \cdot TTTR$ when the synchronous bandwidth is allocated to $S_i$ using the expression defined in (4.19).

4.2 Bandwidth Allocation to Message Streams with their Deadlines no less than $2 \cdot TTTR$

This section discusses how to allocate the synchronous bandwidth to each synchronous message stream $S_i$ with $D_i \geq 2 \cdot TTTR$ such that the protocol and deadline constraints can be satisfied. There are two cases to consider, which are discussed respectively in the following sub sections.

4.2.1 Message Streams with Deadlines no larger than Periods

For the message stream $S_i$ with $D_i \leq P_i$, in order to meet the deadline constraint, the synchronous bandwidth allocated to node $i$ must satisfy (3.4), which is restated as follow

$$X_i(D_i) \geq C_i$$ (4.20)

where $X_i(D_i)$ is the minimum available time for node $i$ to transmit its messages during the time interval $D_i$.

Because this study develops an efficient local SBA scheme, which uses only the locally available information when allocating synchronous bandwidth to node $i$, the minimum available transmission time does not rely on any information from other nodes. Otherwise, the resulting $H_i$ from (4.20) must have information from other nodes. Based on Corollary 2, it is known that the minimum available transmitting time for node $i$ during $D_i$, $x_i(D_i)$, based only on node $i$'s
local information, is
\[ x_i(D_i) = \left(\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1\right) \cdot H_i + \max\{0, D_i + H_i - \left(\left\lfloor \frac{D_i}{TTRT} \right\rfloor + 1\right)TTRT\} \]
\[ = (q_i - 1) \cdot H_i + \max\{0, D_i + H_i - (q_i + 1)TTRT\} \] (4.21)

where \( q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \).

From (4.20) and (4.21), there is
\[ (q_i - 1) \cdot H_i + \max\{0, D_i + H_i - (q_i + 1)TTRT\} \geq C_i \] (4.22)

Since allocating more synchronous bandwidth than actually needed does not increase the chance of guaranteeing the message deadline constraint but on the contrary may cause the protocol constraint to be violated, the minimum required \( H_i \) is tried to be allocated to node \( i \) under (4.22), which maximises the possibility of meeting the protocol constraint. It is easy to check that the left hand side of (4.22) is an increasing function of \( H_i \), so the minimum \( H_i \) exists and can be calculated from
\[ (q_i - 1) \cdot H_i + \max\{0, D_i + H_i - (q_i + 1) \cdot TTRT\} = C_i \] (4.23)

Deriving \( H_i \) from (4.23), there is
\[ H_i = \frac{C_i}{q_i - 1} - \frac{1}{q_i} \cdot \max\{D_i - [(q_i + 1)TTRT - \frac{C_i}{q_i - 1}], 0\} \] (4.24)

The detailed derivation process can be found in Appendix C.

It is obvious that when the allocation defined by (4.24) is used to any synchronous message stream \( S_i \) with \( 2 \cdot TTRT \leq D_i \leq P_i \), the deadline constraint can be satisfied under the protocol constraint.

### 4.2.2 Message Streams with Deadlines larger than Periods

There are only two local SBA schemes which can be used for synchronous message streams with \( D_i > P_i \) and \( D_i \geq 2 \cdot TTRT \): the \( LA_M \) scheme, as shown in (3.8), proposed by Malcolm et al [56] and a similar scheme, as shown in (3.9), proposed by Zheng et al [96]. Since the scheme proposed by Zheng et al is actually similar to the \( LA_M \) scheme which has been adopted in the SAFENET Standard [65], only the \( LA_M \) scheme is considered here. In order to compare the final SBA scheme with the \( LA_M \) scheme, a similar analysis, which is used in the deduction of the \( LA_M \) scheme, is adopted.

Due to the complexity of this section, for a clear description, the process of the derivation of the allocation scheme, which is composed with five steps, is generalised first.
Step 1: A possible result of $H_i$, which can be used for synchronous message streams with $D_i > P_i$ and $D_i \geq 2 \cdot TTRT$, is deduced based on the best timing property to date. In order to make the proposed allocation competent and comparable to that proposed by Malcolm et al [56], a similar analysis is adopted to that used by them. However, because the work is based on a better result of the timing property, a better allocation can be derived even if a similar analysis is used.

Step 2: This derived result of $H_i$ is used to test the deadline constraint under the protocol constraint. The $H_i$ is expected to guarantee the deadline constraint for a message stream $S_i$ with $D_i > P_i$ and $D_i \geq 2 \cdot TTRT$, if the protocol constraint can be satisfied. However, this derived allocation cannot achieve such a requirement. It fails for any stream $S_i$ with $\left\lfloor \frac{D_i}{TTRT} \right\rfloor = \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1$.

Step 3: An analysis is undertaken to identify the reasons why the proposed $H_i$, under the protocol constraint, fails to guarantee deadlines for all message streams given that this result is deduced based on a similar analysis to that developed by Malcolm et al whose result can satisfy the deadline constraint. A few errors in the development of $H_i$ are discussed fully.

Step 4: Further analysis is undertaken on which messages from the streams $S_i$ with $\left\lfloor \frac{D_i}{TTRT} \right\rfloor = \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1$ cannot be guaranteed by the proposed $H_i$.

Step 5: A value is calculated to complement the proposed $H_i$. The value is calculated based on the outcomes from Step 4, in order to meet the deadline constraint of those messages whose deadline cannot be guaranteed by the original $H_i$. The deadline constraint of message streams with $\left\lfloor \frac{D_i}{TTRT} \right\rfloor = \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1$ is then checked using this new $H_i$, under the protocol constraint. A further discussion about this new $H_i$ is presented.

Each of the above steps are explained in details in the following sub-sections.

**Deducing the Value of $H_i$ (Step 1)**

According to Corollary 2, when judged only by node $i$’s local information, under the protocol constraint, the minimum available time for transmitting synchronous messages during interval $D_i$ is

$$x_i(D_i) = (\left\lfloor \frac{D_i}{TTRT} \right\rfloor - 1) \cdot H_i + \max[0, D_i + H_i - (\left\lfloor \frac{D_i}{TTRT} \right\rfloor + 1) \cdot TTRT]$$

$$= (q_i - 1) \cdot H_i + \max[0, D_i + H_i - (q_i + 1) \cdot TTRT]$$

(4.25) 57
When considering only one message in the stream $S_i$, in order to meet the deadline constraint, the synchronous bandwidth needs to be allocated such that

$$x_i(D_i) \geq C_i$$ \hfill (4.26)

From (4.25) and (4.26), there is

$$(q_i - 1) \cdot H_i + \max[0, D_i + H_i - (q_i + 1) \cdot TTRT] \geq C_i$$ \hfill (4.27)

Similar to the analysis of (4.22), $H_i$ can be derived as follows

$$H_i = \frac{C_i}{q_i - 1} - \frac{1}{q_i} \max\{D_i - [(q_i + 1)TTRT - \frac{C_i}{q_i - 1}], 0\}$$ \hfill (4.28)

As shown in Section C.7 of Appendix C, the following two conditions are equivalent

$$D_i - (q_i + 1)TTRT + H_i > 0 \Leftrightarrow D_i - (q_i + 1)TTRT + \frac{C_i}{q_i - 1} > 0$$ \hfill (4.29)

when $H_i = \frac{C_i + (q_i + 1)TTRT - D_i}{q_i}$ and $q_i \geq 2$.

Equality (4.27) is only sufficient to meet the deadline constraint of the first message from $S_i$. Thus the derived $H_i$ defined by Eq. (4.28) can only sufficiently satisfy the first message’s deadline. To meet the deadline constraint of the following messages, several other aspects need to be considered.

From the other perspective, in the case of $D_i > P_i$, to meet the deadline constraint, the traffic flow at a node must be statistically balanced. That is, on average, the number of messages arriving at a node in a given time interval must be equal to the number of messages that the node can transmit in the same interval. So when the time length of $D_i$ is considered, the average number of messages\(^5\) arriving at node $i$ during this interval can be loosely regarded as $\frac{D_i}{P_i}$. For the perspective of the balance, $\frac{D_i}{P_i}$ messages must also be transmitted in every interval of length $D_i$. That is, in order to guarantee the messages from $S_i$, the time required within $D_i$ for transmitting synchronous traffic should be no less than $\frac{D_i}{P_i} \cdot C_i$. That is

$$x_i(D_i) \geq \frac{D_i}{P_i} \cdot C_i$$ \hfill (4.30)

With (4.25) and (4.30), there is

$$(q_i - 1) \cdot H_i + \max[0, D_i + H_i - (q_i + 1) \cdot TTRT] \geq \frac{D_i}{P_i} \cdot C_i$$ \hfill (4.31)

\(^5\)The concept of average number is first used by Malcolm et al [56, 55] when considering their allocation scheme to message stream $S_i$ with $D_i > P_i$. Although it is not accurate and not sufficient, it is very suitable for calculating $H_i$ based on the timing property and often leads to practically efficient allocation.
Due to the same consideration as deriving (4.23) from (4.22), there is, from (4.31)

\[(q_i - 1) \cdot H_i + \max[0, D_i + H_i - (q_i + 1) \cdot TTTRT] = \frac{D_i}{P_i} \cdot C_i \quad (4.32)\]

Based on equation (4.32), the following \(H_i\) expression can be obtained

\[H_i = \frac{q_i TTTRT C_i}{q_i - 1} - \frac{1}{q_i} \max\{D_i - [(q_i + 1)TTTRT - \frac{q_i TTTRT C_i}{q_i - 1}], 0\} \quad (4.33)\]

Appendix D shows how the above \(H_i\), given by (4.33), can be deduced from (4.32). It also gives the derivations of the following two equivalent conditions when \(H_i = \frac{q_i TTTRT C_i + (q_i + 1)TTTRT - D_i}{q_i}\)

and \(q_i \geq 2\)

\[D_i - (q_i + 1)TTTRT + H_i > 0 \iff D_i - (q_i + 1)TTTRT + \frac{q_i TTTRT C_i}{q_i - 1} > 0 \quad (4.34)\]

Combing the results from (4.28) and (4.33) gives the following local synchronous bandwidth allocation scheme when \(D_i \geq 2 \cdot TTTRT\) and \(D_i > P_i\)

\[H_i = \max\left(\frac{q_i TTTRT C_i}{q_i - 1}, 1\right) C_i - \frac{1}{q_i} \max\{D_i - [(q_i + 1)TTTRT - \frac{q_i TTTRT C_i}{q_i - 1}], 0\}, \quad (4.35)\]

Similar to (4.29) and (4.34), there exists the following equivalent conditions when \(q_i \geq 2\) and \(H_i = \frac{\max(q_i TTTRT C_i + (q_i + 1)TTTRT - D_i)}{q_i}\)

\[D_i - (q_i + 1)TTTRT + H_i > 0 \iff D_i - (q_i + 1)TTTRT + \frac{\max(q_i TTTRT C_i + (q_i + 1)TTTRT - D_i)}{q_i} > 0 \quad (4.36)\]

**Checking the Deadline Constraint (Step 2)**

It is relatively easy to check the protocol constraint (3.2). Actually, the protocol constraint is required to be satisfied when using Eq. (4.35) to allocate synchronous bandwidth to some streams of a message set because Eq. (4.35) is derived under the protocol constraint (see Corollary 2). However, checking the deadline constraint is difficult, especially the case of \(D_i > P_i\). Three considerations are examined below.

1. In order to test the deadline constraint, it is necessary to test if every synchronous message in a message set can meet its deadline constraint.

Let a busy interval at node \(i\) be a maximal time interval \([t_0, t_1]\) such that at all points of time between \(t_0\) and \(t_1\) inclusive, the buffer for outgoing synchronous messages at node \(i\) is non-empty.

59
A busy interval is always triggered by the arrival of a synchronous message. Considering the \( k \)-th message that arrives during a busy interval \([t_0, t_1]\) on node \( i \) \((k = 1, 2, \ldots, \lceil \frac{t_1 - t_0}{P_i} \rceil)\), in order to meet the deadline constraint, this message must be sent before its absolute deadline, i.e., \( t_0 + (k - 1) \cdot P_i + D_i \). Because all outgoing messages on node \( i \) are queued in FIFO order, enough time must be provided to transmit these \( k \) messages during the interval \([t_0, t_0 + (k - 1) \cdot P_i + D_i]\) in order to meet the \( k \)-th message’s deadline constraint.

Theorem 8 below gives the minimum amount of synchronous message transmitting time in such a time interval. The proof of this theorem can be found in Appendix E.

**Theorem 8.** Assuming that at time \( t_0 \), a synchronous message with period \( P_i \) and deadline \( D_i \) arrives at node \( i \). Then, under the protocol constraint, in time interval \([t_0, t_0 + (k - 1) \cdot P_i + D_i]\), the minimum amount of available synchronous transmitting time \( X^k_i(\vec{H}) \), based on the global information of the network, is given by

\[
X^k_i(\vec{H}) = (m_i - 1) \cdot H_i + \max \{0, (k - 1) \cdot P_i + D_i \} - \{m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - \left[ \frac{m_i}{n+1} \cdot \left[ TTRT - \left( \sum_{j=1}^{n} H_j + \tau \right) \right] - H_i \} \}
\]

(4.37)

where \( m_i \) is an integer \((m_i \geq 2)\) which makes the inequality of \( I(m_i - 1) \leq (k - 1) \cdot P_i + D_i < I(m_i) \) hold, and must be either

\[
m_i = \left\lfloor \frac{(k - 1) \cdot P_i + D_i \cdot (n + 1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor
\]

or

\[
m_i = \left\lfloor \frac{(k - 1) \cdot P_i + D_i \cdot (n + 1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor - 1
\]

Because transmitting these \( k \) messages needs at least \( k \cdot C_i \) available transmission time, the deadline constraint can be satisfied only when

\[
X^k_i(\vec{H}) \geq k \cdot C_i
\]

(4.38)

where \( i = 1, 2, \ldots, n \) and \( k = 1, 2, \ldots \).

2. The advantage of using (4.38) to test the deadline constraint of a message set is that the test such conducted is accurate. The testing itself is based on the most accurate result of the protocol timing property and takes the information from the whole network into consideration. However, it is impractical because it needs to test every message from a specific synchronous message stream and it is difficult to find out the maximum possible
value of \( k \) in a busy interval (although the value of \( k \) could be upper bounded). Moreover, when a new message stream joins into the message set or an existing stream drops out, the deadline constraint must be re-checked again for every stream because the testing is based on global information of all nodes in the network. This is a time consuming process and definitely cannot be accepted in practice. Thus, it is necessary to find an other relatively simpler but practical method to check the deadline constraint, even if it is not as accurate as (4.38). Here are several considerations:

- Since using global information to test the deadline constraint of a message set will easily result in a time consuming re-checking process of the whole message set when any change happens to it, it is better to restrict the information used in the deadline constraint testing to the local area. This way the testing of message stream \( S_i \) will not be affected by any changes from other message streams. Although such testing is only sufficient but not necessary, it is practical because there is no requirement of re-checking the deadline constraint for those message streams whose parameters stay unchanged. That is, when any change happens, the re-checking only needs to apply to those changed streams of the message set. Using the same busy interval concept but only considering the local information of node \( i \), according to Corollary 2, during the time interval \([t_0, t_0 + (k - 1) \cdot P_i + D_i]\), in the worst case, the minimum available transmission time based on the node \( i \)’s local information is

\[
x^k_i = \left\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \right\rfloor - 1 \cdot H_i + \max\{(k - 1)P_i + D_i + H_i - \left\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \right\rfloor + 1 \cdot TTRT, 0\}
\]

Thus, the following may help test the deadline constraint (if the deadline constraint is met)

\[
x^k_i \geq k \cdot C_i \quad \text{(where } i = 1, 2, ..., n \text{ and } k = 1, 2, ...\).
\]

(4.39)

When using the above inequality to test the deadline constraint of a message stream, the deadline constraint will be satisfied if it holds. However, if the testing fails, the deadline constraint may still be met since this testing is only sufficient.

- From the perspective of practice, the simplest possible method of testing the deadline constraint should be used. One such method is to test the deadline constraint through testing the protocol constraint. That is, when the protocol constraint is satisfied, the deadline constraint can be automatically satisfied. There is no need to test every single message from the stream. In order to achieve this, one possible way is to make Inequality (4.39) hold for any value of \( k \).
When a new SBA scheme is developed under the deadline constraint, only the protocol constraint needs testing. This simplifies the complex and time-consuming schedulability testing on the part of the deadline constraint. Although using Inequality (4.39) to test the deadline constraint only offers a sufficient test (because it only considers the node \(i\)'s local information and any positive integer value of \(k\), with the upper bound value of \(k\) ignored), it does help solve the time-consuming problem of testing the deadline constraint. Some actually schedulable message sets which can pass the test via Inequality (4.38) may fail to pass the test via (4.39). However, compared with Inequality (4.38), Inequality (4.39) is much easier to use and more practical. Thus, it is still a valuable and important method.

Moreover, compared with the deadline checking expression (of \(\lfloor (k-1)P_i+D_i \rfloor - 1 \cdot H_i \geq k \cdot C_i \)) used by Malcolm et al [56] in their Local Allocation (LAM) scheme, because there is more available transmitting time during time interval of \([t_0, t_0 + (k-1) \cdot P_i + D_i]\) when using Inequality (4.39), some schedulable message sets that cannot be guaranteed by a SBA scheme when judged by \(\lfloor (k-1)P_i+D_i \rfloor - 1 \cdot H_i \geq k \cdot C_i\) can now be guaranteed when judged by Inequality (4.39).

3. There are some relationships between Inequality (4.38) and Inequality (4.39). Firstly, Inequality (4.38) contains all the information used by Inequality (4.39), with the extra information from other nodes. Secondly, when the synchronous bandwidths allocated to a message set is a full allocation, i.e., \(\sum_{i=1}^{n} H_i = TT_RT - \tau\), testing the deadline constraint using (4.39) becomes both sufficient and necessary. That is, testing via (4.39) is equivalent to and as accurate as testing via (4.38). Such equivalency can be shown from the following derivations.

From Theorem 8, to test the deadline constraint of this message set, it needs to check

\[
X_k^i(\bar{H}) = (m_i - 1) \cdot H_i + \max(0, (k-1) \cdot P_i + D_i)
\]

\[
-\{m_i \cdot TT_RT + \sum_{j=1}^{n} H_j + \tau - \lfloor \frac{m_i}{n+1} \rfloor \cdot [TT_RT - (\sum_{j=1}^{n} H_j + \tau)] - H_i\}
\]

\[
= (m_i - 1) \cdot H_i + \max(0, (k-1) \cdot P_i + D_i - \{m_i \cdot TT_RT + TT_RT - H_i\})
\]

\[
= (m_i - 1) \cdot H_i + \max(0, (k-1) \cdot P_i + D_i - (m_i + 1) \cdot TT_RT + H_i]
\]

\[
\geq k \cdot C_i
\]
where
\[ m_i = \left\lfloor \frac{((k-1) \cdot P_i + D_i) \cdot (n+1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor \]

or
\[ m_i = \left\lfloor \frac{((k-1) \cdot P_i + D_i) \cdot (n+1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor - 1 \]

However, if \( m_i = \left\lfloor \frac{((k-1) \cdot P_i + D_i)}{TTRT} \right\rfloor - 1 \), there is
\[ I(m_i) = m_i \cdot TTRT + \sum_{k=1}^{n} H_h + \tau - \left\lfloor \frac{m_i}{n+1} \right\rfloor \left[ TTRT - \left( \sum_{h=1}^{n} H_h + \tau \right) \right] \]
\[ = \left\lfloor \frac{(k-1) \cdot P_i + D_i}{TTRT} \right\rfloor TTRT + \sum_{k=1}^{n} H_h + \tau \]
\[ = \frac{(k-1) \cdot P_i + D_i}{TTRT} \cdot TTRT \leq (k-1) \cdot P_i + D_i \]

and this result violates the requirement of \( I(m_i - 1) \leq (k-1) \cdot P_i + D_i < I(m_i) \). Therefore, \( m_i \) can only be \( \left\lfloor \frac{(k-1) \cdot P_i + D_i}{TTRT} \right\rfloor \). So, there is
\[ X^k_i(\vec{H}) = (m_i - 1) \cdot H_i + \max\{0, (k-1) \cdot P_i + D_i - (m_i + 1) \cdot TTRT + H_i \} \]
\[ = \left\lfloor \frac{(k-1) \cdot P_i + D_i}{TTRT} \right\rfloor - 1 \cdot H_i \]
\[ + \max\{(k-1)P_i + D_i + H_i - \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] + 1 \cdot TTRT, 0\} \]
\[ = x^k_i \]

The above analysis implies that under full allocation, testing deadline constraints using Inequality (4.39) is as accurate as using Inequality (4.38). The testing via Inequality (4.39) is not only sufficient but also necessary in the case of full allocation.

Based on the above discussion, the following two conclusions can be obtained.

1. Inequality (4.39) is useful for testing the deadline constraint although the testing is only sufficient and therefore it may fail to test the schedulability of some schedulable message sets. However, passing the test via Inequality (4.39) means the satisfaction of the deadline constraint.
2. When a message set fails to pass the test of Inequality (4.39), it is better to use Inequality (4.38) to find out if the deadline constraint of this considered synchronous message set really fails to be guaranteed.

These two conclusions are used to obtain the following theorem.

**Theorem 9.** Under the protocol constraint, the deadline constraint can be satisfied for any stream $S_i$ with $D_i \geq 2 \cdot TTRT$ and $D_i > P_i$ when the synchronous bandwidth is allocated to $S_i$ using the allocation defined in Eq. (4.35) except the case of $q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lceil \frac{P_i}{TTRT} \rceil + 1$.

**Proof.** Fig. 4.4 shows the framework of the proof to this theorem.
For the convenience of proof, the allocation of Eq. (4.35) is re-stated as follows:

\[ H_i = \frac{\max(\frac{q_i TTRT}{P_i}, 1) C_i}{q_i - 1} \]

\[ - \frac{1}{q_i} \max\{D_i - [(q_i + 1) TTRT - \frac{\max(\frac{q_i TTRT}{P_i}, 1) C_i}{q_i - 1}]0\}, \]

Because a different value of \(D_i - [(q_i + 1) TTRT - \frac{\max(\frac{q_i TTRT}{P_i}, 1) C_i}{q_i - 1}]\) (whether its value is larger than zero or no larger than zero) will lead to a different value of \(H_i\), two cases need to be considered.

**Case 1:** When \(D_i - [(q_i + 1) TTRT - \frac{\max(\frac{q_i TTRT}{P_i}, 1) C_i}{q_i - 1}] \leq 0\), according to the first conclusion above, it only needs to test if the Inequality (4.39) can hold for any \(k \geq 1\).

**Case 2:** When \(D_i - [(q_i + 1) TTRT - \frac{\max(\frac{q_i TTRT}{P_i}, 1) C_i}{q_i - 1}] > 0\), the situation becomes complex. The node can have more available synchronous bandwidth to transmit its synchronous messages because the max part of Inequality (4.39) is larger than zero. There are two sub-cases to consider: \(D_i \geq 2 TTRT + P_i\) and \(\max(2 TTRT, P_i) < D_i < 2 TTRT + P_i\). For the first sub-case, it needs to show that the deadline constraint can be satisfied (based on the first concluding point stated above). For the second sub-case, it is further divided into three small sub-cases: \(q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor\), \(q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1\) and \(q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 2\). These three small sub-cases cover all the possible synchronous message streams with \(\max(2 TTRT, P_i) < D_i < 2 TTRT + P_i\). For the first and third sub-cases, according to the first concluding point, a proof on the satisfaction of the deadline constraint can be given. For the second small sub-case, based on the second concluding point, it is shown that the deadline constraint may fail to be satisfied by an example. This example will show that when using Eq. (4.35) for a message stream with \(q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1\) for a given synchronous message set, under the protocol constraint, the deadline constraint cannot be met.

A complete proof to this theorem, which builds upon the proof flow chart in Fig. 4.4, can be found in Appendix F.

Listed below are a few notes on Theorem 9.

1. Theorem 9 refers to allocation of bandwidth to a message stream \(S_i\) with \(D_i \geq 2 \cdot TTRT\) and \(D_i > P_i\) of a general message set.

2. When using allocation Eq. (4.35) to allocate synchronous bandwidths to a message set, under the protocol constraint, if every message stream \(S_i\) \((i = 1, 2, \ldots, n)\) satisfies \(D_i \geq 2 \cdot TTRT, D_i > P_i\) and \(q_i = \lfloor \frac{D_i}{TTRT} \rfloor \neq \lfloor \frac{P_i}{TTRT} \rfloor + 1\), the deadline constraint must be satisfied for the message set considered.
3. When the message set contains a stream $S_i$ with $q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1$, the deadline constraint **MAY NOT** be satisfied for that stream.

**An Important Note on the Exception Case of Theorem 9 (Step 3)**

Theorem 9 covers the testing of the deadline constraint, which is based on node $i$’s local information. It shows that the synchronous bandwidth allocated by Eq. (4.35) may miss the deadline constraint of some message sets under the protocol constraint. This implies that the allocation scheme defined by Eq. (4.35) could be too small to satisfy the deadline constraints of some synchronous message streams.

The reasons why the allocation defined by (4.35) cannot guarantee all message streams whose bandwidths have been allocated through (4.35) (as described in Theorem 9) can be explained as follows:

1. It is stated at the begin of this section that this section uses a similar analytical method to that originally adopted by Malcolm *et al* [56] in developing their generalised SBA scheme. There are two inaccurate places when deriving $H_i$ for each message stream $S_i$ with $D_i > P_i$:

   - Firstly, (4.30) is based on the requirement of the deadline constraint. However, the average number of messages arriving at node $i$ during the time interval $D_i$ is considered, rather than the actual number of messages arriving at node $i$ during the time interval of $(k - 1) \cdot P_i + D_i$. This average number is *loosely* considered as $\frac{D_i}{P_i}$. That is, the sufficient and necessary condition should be

     $$(\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \rfloor)H_i + \max(0, (k - 1)P_i + D_i + H_i - (\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \rfloor + 1)TTRT) = \left\lfloor \frac{(k - 1)P_i + D_i}{P_i} \right\rfloor \cdot C_i,$$

     rather than

     $$(q_i - 1)H_i + \max(0, D_i + H_i - (q_i + 1)TTRT) = \frac{D_i}{P_i} \cdot C_i$$

     During the deadline constraint analysis, this leads to an error, $\delta_1$, which is the difference between the actual number of messages and the average number of messages arriving at node $i$ during a certain time interval.

   - Secondly, in Appendix D, when deriving $H_i$ from Eq. (4.32), $\frac{rD_i}{q_i-1}C_i$ in Eq. (D.4) is taken as redundant part and is ignored in order to get a smaller $H_i$ (A similar process was adopted by Malcolm *et al* [55]). Similarly, in Eq. (D.9), $\frac{rD_i}{q_i}C_i$ is
considered as an extra part. This may lead to a smaller available synchronous message transmitting time during a given time interval. The error here, $\delta_2$, is defined as the difference between the actually required synchronous bandwidth and the actually allocated synchronous bandwidth.

Malcolm et al’s work is based on an approximate timing property (i.e., (3.12)). In reality, during a given time interval, there is more available transmitting time than that derived via (3.12). Let $\delta$ be such a difference. Their proof, which shows their allocation scheme can ensure the satisfaction of any message’s deadline constraint under the protocol constraint, implies that the difference $\delta$ can compensate well for the errors $\delta_1$ and $\delta_2$. However, the result here is based on a more accurate result of protocol timing property (i.e., (3.13)). Thus, $\delta$ is not big enough to compensate for the lost time caused by errors $\delta_1$ and $\delta_2$.

Furthermore, consider the time interval $D_i$ (because Eq. (4.30) is considered in time interval $D_i$). When only judged by a node’s local information, based on (3.17), the difference of timing property (3.12) and (3.13) happens when $D_i - (q_i + 1) + H_i > 0$. That is, only when $D_i - (q_i + 1) + H_i > 0$, the timing property (3.13) enables more available transmitting time when compared with (3.12). Thus, the case where $\delta$ may not be big enough to compensate for the error of $\delta_1$ and $\delta_2$ can possibly happen only when $D_i - (q_i + 1) + H_i > 0$. That is, if there are some messages whose deadline constraints cannot be satisfied based on the allocation (4.35), these messages must come from the stream $S_i$ with $D_i - (q_i + 1) + H_i > 0$, i.e., $D_i - (q_i + 1)TTRT + \max(\frac{\frac{TTRT}{q_i-1}}{C_i}, 1)C_i > 0$ (because of (4.36)).

2. It is worthy to consider two situations for a specific synchronous message stream with fixed $C_i$ and $P_i$, in the time interval of $D_i$. Firstly, it is clear that under the case where $D_i - P_i = 0$, there is at most one message waiting in the queue to be transmitted. It is assumed that in this case, the deadline constraint can be satisfied. Then, if $D_i$ is slightly increase, there might be more than one message waiting in the queue. This requires more accumulated transmission time. However, although message’s $D_i$ increases, the messages following the first one waiting in the queue may not get a chance to obtain enough transmission time which is only available during the next token’s visit. Thus more synchronous bandwidth needs to be allocated to meet the deadline constraint. Secondly, when $D_i$ receives a big increment, although there might be more messages waiting in the queue, there could be more token visits to node $i$ before the last message’s deadline, and thus possibly more available accumulated transmitting time, which leads to a better chance to meet the deadline constraint even with a smaller $H_i$. One may find it easier to understand this when considering an infinite $D_i$. In this situation, $H_i$ can theoretically
tend to as small as zero.

As revealed by the above two situations, when \( D_i \) is increasing, the value of \( H_i \) is affected by the combined effects of both the increased number of arrived messages and the increased chances to use the allocated synchronous bandwidth. The increased number of arriving messages demands a bigger \( H_i \) while the increased chances to use the allocated synchronous bandwidth leads to the requirement of a smaller \( H_i \). The combination of these two effects leads to the following trend of \( H_i \): When \( D_i \) gradually increases, from the beginning, the additional required transmitting time to transmit those messages further produced is larger than the additional acquired transmitting time because of token more visits. Thus, a larger \( H_i \) will be needed. When \( D_i \) continues to increase, the situation is the inverse, and a smaller \( H_i \) is required.

Theorem 9 reflects the above two effects for the possible difference between \( D_i \) and \( P_i \).

From Theorem 9, when Eq. (4.35) is used to allocate synchronous bandwidth for a stream with \( D_i \geq 2 \cdot TTRT \) and \( D_i > P_i \), the deadline constraint of the stream with \( \lfloor \frac{D_i}{TTRT} \rfloor \neq \lfloor \frac{P_i}{TTRT} \rfloor + 1 \) can be guaranteed while the deadline constraint of the stream with \( \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \) may not be satisfied.

Messages from \( S_i \) with \( \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \) (Step 4)

From this step, a further check will be made to show which messages from message stream \( S_i \) with \( \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \) will miss their deadlines if Eq. (4.35) is used to allocate synchronous bandwidth, in order to get a suitable \( H_i \) which can be used for \( S_i \) but still is no worse than the result proposed by Malcolm et al.

The analysis is still based on node \( i \)'s local information to remain consistent with the above discussion. Theorem 10 below gives such conditions under which messages from \( S_i \) may miss their deadlines.

**Theorem 10.** When the synchronous bandwidth allocated to stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \), \( \lfloor \frac{P_i}{TTRT} \rfloor + 1 = \lfloor \frac{D_i}{TTRT} \rfloor = q_i \) and \( D_i - (q_i + 1)TTRT + H_i > 0 \) using Eq. (4.35), not all the messages from \( S_i \), which satisfy either of the following two groups of conditions, can be transmitted before their deadlines, when only judged by node \( i \)'s local information.
The first group of conditions:

\[
\begin{cases}
    \left\lfloor \frac{k r_i}{P_i} \right\rfloor TTRT < (k - 1)r_P + r_D, \text{ and} \\
    \left\lfloor \frac{k r_i}{P_i} \right\rfloor TTRT > (k - 1)r_P, \text{ and} \\
    H_i > TTRT - r_D, 
\end{cases}
\]  

(4.40)

The second group of conditions:

\[
\begin{cases}
    (k - 1)r_P + r_D < \left(\left\lfloor \frac{k r_i}{P_i} \right\rfloor + 1\right) TTRT \text{ and} \\
    \left\lfloor \frac{k r_i}{P_i} \right\rfloor TTRT > (k - 1)r_P, \text{ and} \\
    H_i \geq \left\lfloor \frac{k r_i}{P_i} \right\rfloor TTRT - (k - 1)r_P + TTRT - r_D, 
\end{cases}
\]  

(4.41)

where \( k \geq 2, r_{D_i} = D_i - q_i \cdot TTTRT \) and \( r_{P_i} = P_i - (q_i - 1) \cdot TTTRT \).

A proof of this theorem can be found in Appendix G.

Adjustment of the Value of \( H_i \) for \( S_i \) with \( \left\lfloor \frac{D_i}{P_i} \right\rfloor = \left\lfloor \frac{P_i}{TTTRT} \right\rfloor + 1 \) (Step 5)

As suggested by Theorem 10, when using \( H_i \) defined in Eq. (4.35), the deadline of some messages from message stream \( S_i \) with \( \left\lfloor \frac{D_i}{P_i} \right\rfloor = \left\lfloor \frac{P_i}{TTTRT} \right\rfloor + 1 \) will be missed. This implies that the total accumulated transmission time may be not enough for transmitting some synchronous messages. Thus, a new way must be found to allocate the \( H_i \) particularly to these streams.

The final result must be comparable to the \( H_i \) proposed by Malcolm et al. That is, in terms of guaranteeing the message deadlines, the proposed allocation must perform no worse than the \( H_i \) proposed by them, as re-stated in the (4.42) below:

\[
H_i \leq \frac{\alpha TTTRT C_i}{q_i - 1} \tag{4.42}
\]

In order to find such \( H_i \), an \( H_i \) can be defined as

\[
H_i = \frac{\alpha TTTRT C_i}{q_i - 1} - \Delta \tag{4.43}
\]

where \( \Delta \geq 0 \)

Again, the value of \( H_i \) should be as small as possible to earn a better chance to meet the protocol constraint. That is, the value of \( \Delta \) should be as big as possible under the satisfaction of the deadline constraint for messages from \( S_i \).

Listed below are a few ideas to find the value of \( \Delta \):
1. Theorem 10 considers messages from the second one (i.e., \( k \geq 2 \)) in the message stream \( S_i \) because the deadline constraint of the first message can always be guaranteed when using Eq. (4.35) to allocate the synchronous bandwidth. However, when deducing \( \Delta \), it is better to consider messages from the first one because \( H_i \) is allocated for a message stream \( S_i \) but not individual messages from that stream. That is, \( k \geq 1 \) should be considered.

2. The value of \( H_i \) defined by Eq. (4.35) must be adjusted to that defined by Eq. (4.43), in order to meet the deadline constraint of synchronous message stream \( S_i \) with \( \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \), according to Theorem 10. When deducing the value of \( \Delta \), the value of \( H_i \) contained in the conditions in Theorem 10 must be changed to that defined by Eq. (4.43). It is no longer that defined by Eq. (4.35).

3. Theorem 10 shows the conditions under which the messages’ deadlines may be missed. Thus, one way to calculate the value of \( \Delta \) is to find a value that can be used in Eq. (4.43) to make those messages matching the conditions defined by Theorem 10 meet their deadlines. However, because the \( H_i \) used in Theorem 10 is no longer that originally defined by Eq. (4.35) but by Eq. (4.43), other messages excluded by Theorem 10 may miss their deadlines under the new allocation of (4.43), although originally their deadline constraint can be satisfied when using Eq. (4.35). Thus, it is necessary to validate whether this resulted \( H_i \), i.e., Eq. (4.43), can be used for any messages from message stream \( S_i \) with \( \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \).

Appendix H shows a process of deduction of \( \Delta \), based on the above ideas. \( \Delta \) is derived as follows

\[ \Delta = \frac{(q_i TTRT - P_i)C_i}{P_i(q_i - 1)} \quad (4.44) \]

From Eq. (4.43), there is

\[
H_i = \frac{q_i TTRT}{q_i - 1} \cdot C_i - \Delta \\
= \frac{q_i TTRT}{q_i - 1} \cdot C_i - \frac{(q_i TTRT - P_i)C_i}{P_i(q_i - 1)} \quad \text{(by (4.44))} \\
= \frac{C_i}{q_i - 1} \quad (4.45)
\]
Using this $H_i$, there is

$$x_i^k = \left[\frac{(k-1)P_i + D_i}{TTTRT}\right] - 1 \cdot H_i$$

$$+ \max\{(k-1)P_i + D_i + H_i - \left[\frac{(k-1)P_i + D_i}{TTTRT}\right] + 1\} \cdot TTTRT, 0\}$$

$$\geq \left[\frac{(k-1)P_i + D_i}{TTTRT}\right] - 1 \cdot H_i$$

$$= \left[\frac{(k-1)(q_i - 1)TTTRT + r_P} + q_i TTTRT + r_D_i}]{TTTRT} - 1\right] \cdot H_i$$

$$= [(k-1)(q_i - 1) + q_i + \left[\frac{(k-1)r_P + r_D}{TTTRT}\right] - 1] \cdot H_i$$

$$\geq k(q_i - 1)H_i$$

$$= k(q_i - 1)\frac{C_i}{q_i - 1}$$

$$= k \cdot C_i$$

Based on the discussion in Step 2, the above inequality implies that when using $H_i = \frac{C_i}{q_i - 1}$ to allocate synchronous bandwidths to $S_i$ with $D_i \geq 2 \cdot TTTRT$ and $\left\lfloor \frac{D_i}{TTTRT}\right\rfloor = \left\lfloor \frac{P_i}{TTTRT}\right\rfloor + 1$, the deadline constraint can always be satisfied. Although the $H_i$ is deduced under the condition of $D_i - (q_i + 1)TTTRT + H_i > 0$, the result can be used for any messages from these message streams.

This new $H_i$ defined by Eq. (4.45) cannot be simply compared with $H_i$ defined by Eq. (4.35) (i.e., the value of this new $H_i$ may be bigger or smaller), although it is derived based on Theorem 10. Theorem 10 only gives the statement that the transmission of those messages may not be guaranteed, thus, $H_i$ defined by Eq. (4.35) may not be sufficient. However, $H_i$ defined by Eq. (4.35) is possibly big enough to satisfy the deadline constraint of some messages with the conditions described in Theorem 10. Moreover, when $H_i$ changes, the condition $D_i - (q_i + 1)TTTRT + H_i > 0$ changes too. This implies that the range of message streams which $H_i$ can be applied to also changes. However, $H_i$ defined by Eq. (4.45) is definitely smaller than that proposed by Malcolm et al.
4.2.3 Summary

Summarising the above discussion, when \( D_i \geq 2 \cdot TTRT \), combining (4.24), (4.35) and (4.45) together gives the following result

\[
H_i = \begin{cases} 
\frac{C_i}{q_i-1} & \text{if } q_i = \left\lceil \frac{P_i}{TTRT} \right\rceil + 1 \\
\frac{1}{q_i} \max\{D_i - (q_i + 1)TTRT + \frac{\max(\frac{q_i TTRT}{P_i}, 1)C_i}{q_i-1}, 0\} & \text{if } q_i \neq \left\lceil \frac{P_i}{TTRT} \right\rceil + 1 \\
\text{max}(q_iTTRT, 1)C_i & \text{if } q_i = \left\lceil \frac{P_i}{TTRT} \right\rceil + 1 \\
\end{cases}
\]

(4.46)

where \( q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \).

The following corollary is obvious.

**Corollary 4.** When \( D_{\text{min}} \geq 2 \cdot TTRT \), under the protocol constraint, the deadline constraint must be satisfied when the synchronous bandwidth is allocated to \( S_i \) using the expression defined in Eq. (4.46).

4.3 A New Generalised SBA Scheme

Based on the discussion in Sections 4.1 and 4.2, a new SBA scheme, which can be used to guarantee a general synchronous message set with arbitrary deadline constraints and its minimum deadline larger than \( TTRT \), is presented as follows:

\[
H_i = \begin{cases} 
C_i & \text{if } TTRT < D_i \leq P_i \\
(TTRT \cdot P_i + 1)C_i & \text{or } TTRT \leq P_i < D_i \\
\text{max}(\frac{q_i TTRT}{P_i}, 1)C_i & \text{if } 0 < P_i < TTRT < D_i \\
\frac{C_i}{q_i-1} & \text{if } q_i = \left\lceil \frac{P_i}{TTRT} \right\rceil + 1 \\
\max(\frac{q_i TTRT}{P_i}, 1)C_i & \text{if } q_i \neq \left\lceil \frac{P_i}{TTRT} \right\rceil + 1 \\
\end{cases}
\]

(4.47)
where \( q_i = \lfloor \frac{D_i}{TTRT} \rfloor \).

### 4.3.1 A Few Notes on the New Local SBA Scheme

Here are a few notes on the new local SBA scheme defined by Eq. (4.47).

- The upper half of Eq. (4.47) is for message streams with \( TTRT < D_i < 2 \cdot TTRT \) while the lower half is used for the message streams with \( D_i \geq 2 \cdot TTRT \). These two different parts come from the different considerations when deducing the new \( H_i \). A message stream \( S_i \) with \( TTRT < D_i < 2 \cdot TTRT \) is only worth considering when using the most advanced timing property.

The upper half of Eq. (4.47) is further divided into two cases. One is for \( D_i \leq P_i \) while the other is for \( D_i > P_i \) which includes two sub-cases: \( P_i \geq TTRT \) and \( P_i < TTRT \). Obviously, these cases cover all the possibilities of \( TTRT < D_i < 2 \cdot TTRT \).

The lower half of Eq. (4.47) is also further divided into two sub cases. One if for \( q_i = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \) while the other is for \( q_i \neq \lfloor \frac{P_i}{TTRT} \rfloor + 1 \). These two cases covers the whole range of \( D_i \geq 2 \cdot TTRT \).

- Eq. (4.47) improves the performance of allocating synchronous bandwidth from two perspectives. Firstly, it relaxes the constraint on the deadline of message streams from \( D_i \geq 2 \cdot TTRT \) to \( D_i > TTRT \), which covers all the legitimate deadline range in practice. Secondly, it enhances the allocation for the message streams with \( D_i \geq 2 \cdot TTRT \) by reducing the value of \( H_i \). The deadline constraint and the buffer constraint can still be satisfied (see below in Section 4.3.2) while the possibility of meeting the protocol constraint is improved because of the reduced \( H_i \). These two advantages of the new SBA scheme benefit from the adoption of the best results on the protocol timing property.

- Allocation scheme Eq. (4.47) is a local SBA scheme. This means Eq. (4.47) is not an optimal scheme. That is, when a message set is not schedulable when using Eq. (4.47) to allocate synchronous bandwidth, there may still exist a feasible allocation for the message set. However, as a local SBA scheme with its good performance, Eq. (4.47) is flexible and suitable for use in dynamic environments from a network management perspective.

For convenience in the later discussion, the new local SBA scheme proposed here will be referred as the Generalised Local Allocation (GLA) scheme.

Actually, by carefully studying the proof of Theorem 9, the performance of GLA can be further improved for the message streams \( S_i \) with \( q_i \geq \lfloor \frac{P_i}{TTRT} \rfloor + 3 \). The allocated synchronous
bandwidth \( H_i \) can be further reduced to \( \frac{TTRT}{P_i} C_i \) for these streams, which can be explained as follows:

Because \( q_i \geq \left\lceil \frac{P_i}{TTRT} \right\rceil + 3 \), there is

\[
D_i - P_i \geq q_i TTRT + r_{D_i} - [(q_i - 3)TTRT + r_{P_i}] = 3 \cdot TTRT + r_{D_i} - r_{P_i} > 2 \cdot TTRT
\]

where \( 0 \leq r_{D_i} < TTRT \), \( 0 \leq r_{P_i} < TTRT \).

Thus

\[
(k - 1)P_i + D_i > (k - 1)P_i + P_i + 2TTRT = kP_i + 2TTRT
\]

Based on this inequality, Fig. 4.5 can be derived to show the amount of available transmission time in the worst case situation in the interval of \( (k - 1)P_i + D_i \), which can be expressed as follows

\[
x_i^k \geq \left( \frac{kP_i}{TTRT} \right) - 1 + 2 \cdot H_i
\]

\[
= \left( \frac{kP_i}{TTRT} \right) + 1 \cdot H_i
\]

\[
> \cdot \frac{kP_i}{TTRT} \cdot H_i
\]

In order to satisfy the deadline constraint, it only needs \( x_i^k \geq kC_i \). That is

\[
\frac{kP_i}{TTRT} \cdot H_i \geq kC_i
\]

\[
\iff H_i \geq \frac{TTRT}{P_i} C_i
\]

With the above discussion, an improved version of the GLA scheme, the IGLA scheme, can be obtained, as expressed below.\(^6\)

\(^6\)The GLA scheme is still kept in this research because the deducing of the GLA scheme is well organised while the IGLA scheme is actually obtained from the informal observation of the deriving process of GLA, although it is explained by Fig.4.5. However, the IGLA scheme does perform better than the GLA scheme because of the
\[
H_i = \begin{cases} 
C_i & \text{if } TTRT < D_i \leq P_i \\
\frac{(TTRT/P_i + 1)C_i}{q_i - 1} & \text{if } 0 < P_i < TTRT < D_i \\
\frac{\max\left(\frac{q_i \cdot TTRT}{P_i - 1}, 1\right)C_i}{q_i - 1} - \frac{1}{q_i} \cdot \max[D_i - (q_i + 1)TTRT + \frac{\max\left(\frac{q_i \cdot TTRT}{P_i - 1}, 1\right)C_i}{q_i - 1}, 0] & \text{if } q_i \leq \left\lfloor \frac{P_i}{TTRT} \right\rfloor \text{ or } q_i = \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 2 \\
\frac{TTRT}{P_i}C_i & \text{if } q_i \geq \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 3
\end{cases}
\]

where \( q_i = \left\lfloor \frac{D}{TTRT} \right\rfloor \).

It is easy to verify that Theorem 11 and Theorem 12 (see below in Section 4.3.2), without any modification, can be used for the IGGA scheme for the schedulability testing, including the protocol constraint, the deadline constraint and the buffer constraint.

### 4.3.2 Feasibility Analysis

This section discusses the feasibility of the proposed GLA scheme defined in Eq. (4.47). As mentioned in Section 3.3.2, for a given synchronous message set to become schedulable, three constraints (on protocol, deadline and buffer) must be satisfied.

Theorem 11 below shows the condition under which the protocol and deadline constraints can be met when the SBA scheme defined by Eq. (4.47) is adopted to allocate synchronous bandwidths to any message set with \( D_{\text{min}} > TTRT \).

smaller \( H_i \) allocated to stream \( S_i \) with \( q_i \geq \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 3 \). In Chapter 6, simulations and numerical examples will be used to show the difference between the IGGA scheme and the GLA scheme. Another reason to retain the GLA scheme is that this scheme will be used to find out the WCAU of the new schemes in Chapter 5. (The IGGA is an improved version of the GLA and have the same WCAU as that of GLA. However, to derive the WCAU of the GLA is simpler compared with the IGGA because the GLA has a simpler format.)
Theorem 11. If the synchronous bandwidth is allocated according to Eq. (4.47) for a synchronous message set with \( D_{\text{min}} > \text{TTRT} \), then the protocol and deadline constraints of this message set can be satisfied under

\[
\sum_{h=1}^{n} H_h \leq \min\{\text{TTRT} - \tau, D_{\text{min}} - \text{TTRT} - \tau\}
\]

(4.49)

Proof. Based upon the discussion in Section 4.1, the allocation \( \bar{H} \) must satisfy the inequality (4.3), for guaranteeing the deadline constraint. Since the protocol constraint (i.e., inequality (3.2)) must also be met if the message set is schedulable, combining (4.3) and (3.2) together, the inequality (4.49) can be obtained. In fact, when \( D_{\text{min}} \geq 2 \cdot \text{TTRT} \), inequality (4.49) is equivalent to (3.2). When \( 2 \cdot \text{TTRT} > D_{\text{min}} > \text{TTRT} \), (4.49) is equivalent to (4.3). Thus, the violation of (4.49) implies either protocol or deadline constraints can not be satisfied.

To the buffer constraint, the following theorem can be obtained.

Theorem 12. Let the synchronous bandwidth be allocated using Eq. (4.47). Let \( b_i \) be the maximum number of bytes in each synchronous message from stream \( S_i \). If the deadline constraint is satisfied, then the buffer needed at node \( i \) is no more than

\[
b_i \cdot \max\{\left\lceil \frac{2 \cdot \text{TTRT}}{P_i} + 1 \right\rceil, 3\}
\]

bytes.

A proof to the above theorem can be found in Appendix I.

Theorem 12 indicates that the maximum queue length is limited and depends only on the network polling speed (i.e., \( \text{TTRT} \)) and the message inter-arrival time (i.e., \( P_i \)). It is independent of message deadlines (i.e., \( D_i \)). Even when message deadlines are very large, the throughput of synchronous messages will be sufficient to prevent a large buildup of queued messages. This gives significant benefits for system design. An application designer can choose message deadlines freely without concern over potential buffer overflow. For example, a designer can choose deadlines for a voice transmission application from several hundred microseconds to half a second, while keeping the buffer size unchanged.

4.3.3 Comparison with Some Other Local SBA Schemes

In this section, the GLA scheme is briefly compared with some other local SBA schemes. It will become clear that the GLA scheme (also, the IGLA due to its better performance than GLA) can perform better than any of these local schemes. In fact, GLA performs no worse than any
previously published local SBA schemes. Chapter 6 will further demonstrate the superiority of GLA over other local schemes through simulation results and numerical examples.

For convenience of discussion, let \( H_i^{SCHEME} \) denote the \( H_i \) produced by a local allocation scheme called \( SCHEME \). For example, \( H_i^{GLA} \) represents the synchronous bandwidth allocated to node \( i \) by the GLA scheme.

**Comparing with the Local Allocation (\( LA_M \)) Scheme**

The GLA scheme is first compared with the \( LA_M \) scheme proposed by Malcolm et al [56]. The \( LA_M \) scheme is defined as follows:

\[
H_i^{LA_M} = \max\left(\frac{(q_i \cdot TTRT)}{P_i}, 1\right)C_i \quad \text{where } q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \geq 2
\]

The \( LA_M \) scheme is also a local SBA scheme that can be used for a message set with arbitrary deadline constraint, but it shows a weaker guarantee ability than the GLA scheme due to being developed from a relatively less accurate result of the protocol timing property, as shown from the following two observations:

- The \( LA_M \) scheme cannot guarantee any synchronous message sets with \( TTRT < D_{\min} < 2 \cdot TTRT \) due to the inapplicability caused by the restriction of \( D_{\min} \geq 2 \cdot TTRT \). However, the GLA scheme can be used for such message sets.

- For synchronous message sets with \( D_{\min} \geq 2 \cdot TTRT \), the allocation \( \vec{H} \) produced by either the \( LA_M \) scheme or the GLA scheme can meet the deadline constraint. Thus, an allocation \( \vec{H} \) becomes feasible only if it satisfies the protocol constraint at the same time.

From the definition of these two schemes, there is (when \( D_{\min} \geq 2 \cdot TTRT \)),

\[
\text{when } q_i \neq \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1,

H_i^{LA_M} = \max\left(\frac{(q_i \cdot TTRT)}{P_i}, 1\right)C_i

\geq \frac{\max\left(\frac{(q_i \cdot TTRT)}{P_i}, 1\right) \cdot C_i}{q_i - 1} - \frac{1}{q_i} \cdot \max[D_i - (q_i + 1)TTRT + \frac{\max\left(\frac{(q_i \cdot TTRT)}{P_i}, 1\right) \cdot C_i}{q_i - 1}, 0] = H_i^{GLA}
\]

\(^7\)Other local SBA schemes proposed by Zhao’s group are not compared with the GLA scheme here because the \( LA_M \) scheme is the best scheme among those proposed by that group.
and when $q_i = \left\lceil \frac{P_i}{TTRT} \right\rceil + 1$,

$$H_i^{LAM} = \max\left(\frac{q_i TTRT}{P_i}, 1\right)C_i \frac{q_i - 1}{q_i}$$

That is, $H_i^{LAM} \geq H_i^{GLA}$. Thus, $\sum_{i=1}^{n} H_i^{LAM} \geq \sum_{i=1}^{n} H_i^{GLA}$. This means that the allocation $\hat{H}$ produced by scheme GLA stands a better chance of satisfying the protocol constraint. In this sense, the GLA scheme is of course superior to the $LAM$ scheme when $D_{\min} \geq 2TTRT$.

**Comparing with the New Local Allocation (NLA) Scheme**

The NLA scheme proposed by Zhang et al [94] is defined as

$$H_i^{NLA} = \begin{cases} 
C_i & \text{if } TTRT < D_i < 2 \cdot TTRT \\
\frac{C_i}{q_i} - \frac{1}{q_i} \max\{D_i - \left(\frac{q_i + 1}{q_i} TTRT - \frac{C_i}{q_i - 1}\right), 0\} & \text{if } D_i \geq 2 \cdot TTRT 
\end{cases}$$

where $q_i = \left\lceil \frac{D_i}{TTRT} \right\rceil$.

It is easy to check that the NLA scheme is a special case of the GLA scheme. GLA will produce exactly the same amount of $H_i$ as NLA when allocating synchronous bandwidths to a synchronous message set with $D_i \leq P_i (i = 1, 2, \ldots, n)$. This is because these two schemes were developed based on the same protocol timing property. Although both schemes can be used for a message set with $D_{\min} > TTRT$, the NLA is restricted to message sets with $D_i \leq P_i$ while the GLA can be used for message sets with arbitrary deadline constraint. From this point of view, the GLA scheme definitely works better than the NLA scheme.

**4.4 Summary**

In this chapter, a new local SBA scheme, the GLA scheme is proposed. This scheme can be used to allocate synchronous bandwidth to any theoretical synchronous message set with arbitrary deadline constraints and minimum deadline larger than $TTRT$. The feasibility of this scheme, including the protocol constraint, the deadline constraint, and the buffer constraint, is also discussed. The proposed scheme is compared with some other published local SBA schemes to
show its superiority on guaranteeing the transmission of synchronous messages. An improved version of the GLA scheme, the IGLA scheme, is also proposed in this chapter.
Chapter 5

The Worst Case Achievable Utilisation

This chapter will focus on the Worst Case Achievable Utilisation (WCAU) of the proposed GLA SBA scheme (including the improved version – the IGLA scheme).

The utilisation of a synchronous message set $M$ with arbitrary deadline constraints is defined as

$$U_e(M) = \frac{\sum_{i=1}^{n} C_i}{\min(P_i, D_i)}$$

$U_e(M)$ can be regarded as the proportion of time required for synchronous traffic in the network. The traditional definition of the utilisation is $U(M) = \sum_{i=1}^{n} \frac{C_i}{P_i}$, as introduced in Chapter 3, which actually fails to represent the effective demand of synchronous message traffic in the case of $D_i < P_i$ [55]. When $D_i < P_i$, because a message from $S_i$ must be sent within $D_i$ time units of its arrival in order to meet its deadline and there is at most one message waiting in the queue, the ratio $\frac{C_i}{D_i}$ should be used to present the real traffic demand of $S_i$.

In [55], $U_e(M)$ is used to denote the effective utilisation of a synchronous message set $M$. Corresponding to this notation, a network (with a given setting of its parameters) has an effective achievable utilisation $U_e$ if it can meet the deadline of any synchronous message set $M$ with $U_e(M)$ no more than $U_e$. The worst case effective achievable utilisation $U^*_e$ of a network is the least upper bound of its effective achievable utilisations.

Because the proposed GLA and IGLA schemes can be used for any synchronous message set including those with $D_i < P_i$, the concepts of $U_e(M)$, $U_e$ and $U^*_e$ will be adopted. However, in this research, these concepts are taken to be identical with the traditional concepts of the utilisation of a synchronous message set $M$, the achievable utilisation of a network and the
WCAU of a network, correspondingly. That is, the word *effective* will be ignored.

The WCAU of the IGLA scheme should be the same as the GLA scheme, because the IGLA scheme is a slightly improved version of the original GLA scheme for any synchronous message stream \( S_i \) with \( q_i = \lfloor \frac{P_i}{TTRT} \rfloor \geq 3 \). In the rest of this chapter, only the WCAU of the GLA scheme is studied, but all the results discussed in this chapter remain true for the IGLA scheme without any modifications.

The rest of this chapter is organised as follows: In Section 5.1, an achievable utilisation of GLA is derived. It is shown that this derived achievable utilisation is actually the WCAU of GLA in Section 5.2. Also, the result of the WCAU of GLA is compared with the WCAUs of other local schemes in this section. Finally, in Section 5.3, the impact of \( TTRT \) on the WCAU is assessed.

### 5.1 An Achievable Utilisation of the GLA Scheme

This section is going to find an achievable utilisation of the new scheme GLA. The following three lemmas, which can be found in [94], are needed in the following discussion.

**Lemma 1.** Let \( A > B \geq 0, \ E > 0 \) and \( r \geq 0 \). Define the function \( f(r) \) such that
\[
f(r) = \frac{B + \max(0,r - E)}{A + r},
\]
then \( f(r) \) reaches its minimum value when \( r = E \), i.e., \( f(r) \geq f(E) = \frac{B}{A + E} \).

**Lemma 2.** If the function \( f(q_i) \) is defined as follows (where \( q_i \) is an integer no less than 2):
\[
f(q_i) = \frac{(q_i - 1)H_i}{(q_i + 1)TTRT - H_i},
\]
then \( f(q_i) \) is an increasing function of \( q_i \).

**Lemma 3.** Under the assumption of \( \sum_{i=1}^{n} H_i = TTRT - \tau \), \( q_{min} = \lfloor \frac{P_i}{TTRT} \rfloor \geq 2 \) and \( 0 \leq H_i \leq TTRT - \tau \), the function \( F(\vec{H}) \), defined as below,
\[
F(\vec{H}) = \sum_{i=1}^{n} \frac{(q_{min} - 1)H_i}{(q_{min} + 1)TTRT - H_i},
\]
reaches its minimum value when \( H_1 = H_2 = \cdots = H_n = \frac{TTRT - \tau}{n} \).

The following theorem shows that the protocol constraint will be satisfied provided that the utilisation of the given set of synchronous messages falls within a certain bound.

**Theorem 13.** For any synchronous message set \( M \) with its utilisation
\[
U_e(M) \leq U_e = \frac{q_{min} - 1}{q_{min} + 1 - \frac{\alpha}{n}}(1 - \alpha) \quad \text{where} \quad q_{min} = \lfloor \frac{D_{min}}{TTRT} \rfloor,
\]
if the synchronous bandwidths are allocated using the scheme defined by Eq. (4.47), then the protocol constraint will be satisfied.

Proof. This theorem is proved by considering the following two cases:

Case 1: For synchronous message sets with $D_{\min} < 2 \cdot TTRT$.

For this case, Zhang et al [94] showed that for any synchronous message set with $D_{\min} < 2 \cdot TTRT$, the WCAU of any SBA scheme with the timed-token protocol can asymptotically approach 0%. Thus, for a message set with $D_{\min} < 2 \cdot TTRT$, the achievable utilisation of the GLA scheme defined by Eq. (4.47) should be 0. On the other hand, because $D_{\min} < 2 \cdot TTRT$, there is

$$q_{\min} = \left\lfloor \frac{D_{\min}}{TTRT} \right\rfloor = 1 \quad (5.2)$$

Replacing $q_{\min}$ in (5.1) by Eq. (5.2), there is, $U_e = 0$, an expected result which matches this case. Therefore, (5.1) holds in this case.

Case 2: For synchronous message sets with $D_{\min} \geq 2 \cdot TTRT$.

Under this case, this theorem is proved by contradiction. Assume that there exists one synchronous message set $M$ with its utilisation factor $U_e(M)$ such that

$$U_e(M) = \sum_{i=1}^{n} \frac{C_i}{\min(P_i, D_i)} \leq \frac{q_{\min} - 1}{q_{\min} + 1 - \frac{\alpha}{n}} (1 - \alpha) \quad (5.3)$$

But, the local scheme defined by Eq. (4.47) cannot guarantee the message set $M$, i.e., the allocation $\vec{H}_i$ produced by Eq. (4.47) is infeasible.

According to the allocation scheme defined by Eq. (4.47), the message streams from the message set $M$ are relabelled and this message set is divided into two groups: Group $G_1$ contains all the message streams with $q_i \neq \lfloor \frac{P_i}{TTRT} \rfloor + 1$ while Group $G_2$ contains all the message streams with $q_i = \lfloor \frac{P_i}{TTRT} \rfloor + 1$, based on the different allocations to these two groups.

The message streams from $G_1$ are labelled $S_1, S_2, \ldots, S_m$ where $1 \leq m \leq n$. Let $f(H_i)$ be

$$f(H_i) = (q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i]) \quad (5.4)$$

then, from Eq. (4.47), the $H_i$ allocated to this group can be considered as the solution to the following function in terms of $H_i$:

$$f(H_i) = \max\left(\frac{q_i TTRT}{P_i}, 1\right)C_i \quad (5.5)$$

Similar, the message streams from $G_2$ are labelled to be $S_{m+1}, S_{m+2}, \ldots, S_n$. Let $g(H_i)$ be

$$g(H_i) = (q_i - 1)H_i \quad (5.6)$$
then, the $H_i$ allocated to this group can be considered as the solution to the following function in terms of $H_i$

$$g(H_i) = C_i$$ (5.7)

When using the allocation scheme Eq. (4.47) to allocate synchronous bandwidth to any message set, the deadline constraint can be satisfied. Thus, the only possible reason for the allocation $\tilde{H}_i$ produced by Eq. (4.47) to be infeasible is the violation of the protocol constraint. That is, when using Eq. (4.47) to allocate synchronous bandwidths, there is

$$\sum_{i=1}^{n} H_i > TTRT - \tau$$

The above discussion shows that the sum of the solutions of Eq. (5.5) or Eq. (5.7) are larger than $TTRT - \tau$.

It is easy to check that both $f(H_i)$ and $g(H_i)$ are increasing functions of $H_i$. Thus, if one or more $H_i$s are reduced to make $\sum_{i=1}^{n} H_i = TTRT - \tau$ holds, then under $\sum_{i=1}^{n} H_i = TTRT - \tau$, there is

$$f(H_i) = (q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i]) \leq \max(\frac{q_i TTRT}{P_i}, 1)C_i$$ (5.8)

and

$$g(H_i) = (q_i - 1)H_i \leq C_i$$ (5.9)

and there must be at least one $H_i$ which makes

$$f(H_i) = (q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i]) < \max(\frac{q_i TTRT}{P_i}, 1)C_i$$ (5.10)

or

$$g(H_i) = (q_i - 1)H_i < C_i$$ (5.11)

The following derivations now can be obtained

$$U_c(M) = \sum_{i=1}^{n} \frac{C_i}{\min(P_i, D_i)}$$

$$= \sum_{i=1}^{n} \frac{C_i}{\min(P_i, D_i)} + \sum_{i=m+1}^{n} \frac{C_i}{\min(P_i, D_i)}$$ (5.12)

Considering the second part of (5.12), i.e., $\sum_{i=m+1}^{n} \frac{C_i}{\min(P_i, D_i)}$, because $q_i = \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1$, there is

$$P_i = (q_i - 1)TTRT + rP_i < q_i TTRT \leq D_i$$

83
Thus
\[
\sum_{i=m+1 \text{ to } n} \frac{C_i}{\min(P_i, D_i)} = \sum_{i=m+1 \text{ to } n} \frac{C_i}{P_i} \geq \sum_{i=m+1 \text{ to } n} \frac{(q_i - 1)H_i}{P_i} = \sum_{i=m+1 \text{ to } n} \frac{(q_i - 1)H_i}{\frac{(q_i - 1)TTRT + r_p}{P_i}} > \sum_{i=m+1 \text{ to } n} \frac{(q_i - 1)H_i}{\frac{(q_i + 1)TTRT - H_i}{P_i}}
\]

(5.13)

Considering the first part of (5.12), i.e., \(\sum_{i=1 \text{ to } m} C_i = \sum_{i=m+1 \text{ to } n} C_i\min(P_i, D_i)\), three sub-cases need to be considered.

- **Sub-case 1: when \(\frac{q_iTTRT}{P_i} \geq 1\)**

For this sub-case, because \(D_i \geq q_iTTRT \geq P_i\), thus
\[
\sum_{i=1 \text{ to } m} \frac{C_i}{\min(P_i, D_i)} = \sum_{i=1 \text{ to } m} \frac{C_i}{P_i} \geq \sum_{i=1 \text{ to } m} \frac{(q_i - 1)H_i}{P_i} = \sum_{i=1 \text{ to } m} \frac{(q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i])}{P_i}
\]

(5.14)

- **Sub-case 2: when \(\frac{q_iTTRT}{P_i} < 1\) and \(D_i \geq P_i\)**

For this sub-case, there is
\[
q_iTTRT < P_i \leq D_i
\]
\[
\Rightarrow q_i \leq \frac{P_i}{TTRT} \leq \frac{D_i}{TTRT}
\]
\[
\Rightarrow q_i \leq \left\lfloor \frac{P_i}{TTRT} \right\rfloor \leq \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i
\]
\[
\Rightarrow \left\lfloor \frac{P_i}{TTRT} \right\rfloor = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i
\]
Thus

\[
\sum_{i=1}^{m} \frac{C_i}{\min(P_i, D_i)} = \sum_{i=1}^{m} \frac{C_i}{P_i}
\]

\[
\geq \sum_{i=1}^{m} \frac{(q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i])}{P_i}
\]

\[
\geq \sum_{i=1}^{m} \frac{(q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i])}{D_i}
\]

\[
= \sum_{i=1}^{m} \frac{(q_i - 1)H_i + \max(0, rD_i - [TTRT - H_i])}{q_iTTRT + rD_i}
\]

\[
\geq \sum_{i=1}^{m} \frac{(q_i - 1)H_i}{(q_i + 1)TTRT - H_i}
\]

(Since \(P_i \leq D_i\))

(5.15)

(by Lemma 1 when \(A = q_iTTRT\), \(B = (q_i - 1)H_i\), \(E = TTRT - H_i\) and \(r = rD_i\))

- **Sub-case 3:** when \(\frac{q_iTTRT}{P_i} < 1\) and \(D_i < P_i\)

Under this sub-case, there is

\[
\sum_{i=1}^{m} \frac{C_i}{\min(P_i, D_i)} = \sum_{i=1}^{m} \frac{C_i}{D_i}
\]

\[
\geq \sum_{i=1}^{m} \frac{(q_i - 1)H_i + \max(0, D_i - [(q_i + 1)TTRT - H_i])}{D_i}
\]

(Similar as in sub-case 2) (5.16)

From (5.14), (5.15) and (5.16), there is

\[
\sum_{i=1}^{m} \frac{C_i}{\min(P_i, D_i)} \geq \sum_{i=1}^{m} \frac{(q_i - 1)H_i}{(q_i + 1)TTRT - H_i}
\]

(5.17)
Thus, based on (5.13) and (5.17), from (5.12), there is

\[ U_e(M) = \sum_{i=1}^{n} \frac{C_i}{\min(P_i, D_i)} \]

\[ > \sum_{i=1 \text{ to } n} \frac{(q_i - 1)H_i}{(q_i + 1)TTTR - H_i} + \sum_{q_i \neq \lfloor \frac{1}{TTTR} \rfloor + 1}^{m} \frac{(q_i - 1)H_i}{(q_i + 1)TTTR - H_i} \]

(The equality part no longer exists because of (5.10) and (5.11))

\[ = \sum_{i=1 \text{ to } n} \frac{(q_i - 1)H_i}{(q_i + 1)TTTR - H_i} \]

\[ \geq \sum_{i=1 \text{ to } n} \frac{(q_{\min} - 1)H_i}{(q_{\min} + 1)TTTR - H_i} \quad \text{(by Lemma 2)} \]

\[ \geq \sum_{i=1 \text{ to } n} \frac{(q_{\min} - 1)\frac{TTTR}{n} - \tau}{(q_{\min} + 1)TTTR - \frac{TTTR}{n}} \quad \text{(by Lemma 3)} \]

\[ = \frac{q_{\min} - 1}{q_{\min} + 1 - \frac{1-\alpha}{n}} (1 - \alpha) \]

However, the above inequality violates the assumption of (5.3). Thus, the assumption cannot hold and the theorem is established. □

Obviously, with Theorems 13, 11 and 12, when Eq. (4.47) is used to allocate synchronous bandwidths for any synchronous message set with its utilisation \( U_e(M) \) no more than \( U_e \) where

\[ U_e = \frac{q_{\min} - 1}{q_{\min} + 1 - \frac{1-\alpha}{n}} (1 - \alpha) \quad (5.18) \]

then, the protocol, deadline and buffer constraints will be satisfied. Thus, \( U_e \) defined by Eq. (5.18) is an achievable utilisation.

### 5.2 The WCAU of the GLA Scheme

This section shows that the achievable utilisation defined by Eq. (5.18) is also the WCAU of scheme GLA, i.e., \( U_e = U^*_e \). The following theorem defines the WCAU of GLA.

**Theorem 14.** If the synchronous bandwidths are allocated using the scheme in Eq. (4.47), then the worst case achievable utilisation, \( U^*_e \), of the network is given by

\[ U^*_e = \frac{q_{\min} - 1}{q_{\min} + 1 - \frac{1-\alpha}{n}} (1 - \alpha) \quad (5.19) \]

**Proof.** As already proved in the last section, \( U^*_e \) defined by Eq. (5.19) is an achievable utilisation. Thus, in order to show that it is also the WCAU, it only needs to show that for any given \( \epsilon > 0 \), there exists a synchronous message set \( M \) with utilisation \( U_e(M) \) such that \( U^*_e < U_e(M) \leq U^*_e + \epsilon \) and the protocol constraint cannot be satisfied for this set of messages.
Such a synchronous message set $M$ can be constructed with $n$ message streams as follows:
for $i = 1, 2, \cdots, n$,

$$C_i = \frac{(q_i - 1)(TTRT - \tau)}{n} + \left(\frac{(q_i + 1)TTRT}{n} - \frac{TTRT - \tau}{n^2}\right) \cdot \epsilon$$

$$P_i = D_i = (q_i + 1)TTRT - \frac{TTRT - \tau}{n}$$

where

$q_1 = q_2 = \cdots = q_n = q_{\text{min}} = \lceil \frac{D_{\text{min}}}{TTRT} \rceil \geq 2$.

It is easy to check that the utilisation of this message set $M$, $U_{\epsilon}(M)$, can satisfy the condition $U_{\epsilon}^* < U_{\epsilon}(M) \leq U_{\epsilon}^* + \epsilon$, as follows

$$U_{\epsilon}(M) = \sum_{i=1}^{n} \frac{C_i}{\text{min}(P_i, D_i)} = \sum_{i=1}^{n} \frac{C_i}{P_i}$$

$$= \sum_{i=1}^{n} \frac{(q_{\text{min}}-1)(TTRT-\tau)}{n} + \left(\frac{(q_{\text{min}}+1)TTRT}{n} - \frac{TTRT-\tau}{n^2}\right) \cdot \epsilon$$

$$= n \cdot \frac{(q_{\text{min}}-1)(TTRT-\tau)}{n} + \left(\frac{(q_{\text{min}}+1)TTRT}{n} - \frac{TTRT-\tau}{n^2}\right) \cdot \epsilon$$

$$= \frac{q_{\text{min}} - 1}{q_{\text{min}} + 1 - \frac{1-\alpha}{n}} (1 - \alpha) + \epsilon$$

It is now shown that this message set $M$ cannot be guaranteed by the local SBA scheme defined by Eq. (4.47). Because $q_i \geq 2 \Rightarrow D_i \geq 2 \cdot TTTR, D_i = P_i \Rightarrow q_i = \lceil \frac{P_i}{TTTR} \rceil$ and $D_i = P_i \Rightarrow q_i TTTR \leq D_i = P_i$, there is, from Eq. (4.47)

$$H_i = \frac{C_i}{q_i - 1} \cdot \frac{1}{q_i} \cdot \text{max}[D_i - (q_i + 1)TTRT + \frac{C_i}{q_i - 1}, 0]$$

$$= \frac{C_i}{q_i - 1} \cdot \frac{1}{q_i} \cdot \text{max}[(q_i + 1)TTRT - \frac{TTRT - \tau}{n}, 0] - (q_i + 1)TTRT + \frac{(q_i - 1)(TTRT - \tau)}{n} + \left(\frac{(q_i + 1)TTRT}{n} - \frac{TTRT - \tau}{n^2}\right) \cdot \epsilon$$

$$= \frac{C_i}{q_i - 1} \cdot \frac{1}{q_i} \cdot \text{max}[(q_i + 1)TTRT - \frac{TTRT - \tau}{n}, 0] - \frac{1}{q_i - 1} \cdot \frac{TTRT}{(q_i - 1)n}(q_i + 1 - \frac{1-\alpha}{n})$$

$$= \frac{TTRT - \tau}{n} + \frac{\epsilon \cdot TTTR}{(q_i - 1)n}(q_i + 1 - \frac{1-\alpha}{n}) - \frac{1}{q_i} \cdot \frac{\epsilon \cdot TTTR}{(q_i - 1)n}(q_i + 1 - \frac{1-\alpha}{n})$$

$$= \frac{TTRT - \tau}{n} + \frac{\epsilon \cdot TTTR}{q_i \cdot (q_i - 1)n}(q_i + 1 - \frac{1-\alpha}{n})$$

$$= \frac{TTRT - \tau}{n} + \frac{\epsilon \cdot TTTR}{q_{\text{min}} \cdot n}(q_{\text{min}} + 1 - \frac{1-\alpha}{n})$$
Thus,

\[ \sum_{i=1}^{n} H_i = \sum_{i=1}^{n} \left( \frac{TTRT - \tau}{n} + \frac{\epsilon \cdot TTRT}{q_{\min} \cdot n} \left( q_{\min} + 1 - \frac{1 - \alpha}{n} \right) \right) \]

\[ = n \cdot \left( \frac{TTRT - \tau}{n} + \frac{\epsilon \cdot TTRT}{q_{\min} \cdot n} \left( q_{\min} + 1 - \frac{1 - \alpha}{n} \right) \right) \]

\[ = TTRT - \tau + \frac{\epsilon \cdot TTRT}{q_{\min}} \left( q_{\min} + 1 - \frac{1 - \alpha}{n} \right) \]

\[ > TTRT - \tau \]

This shows that the protocol constraint for message set \( M \) cannot be satisfied. Therefore, \( U_e^* \) is the WCAU.

The WCAU of the GLA scheme is same as that of the NLA scheme proposed by Zhang et al [94]. However, their WCAU of NLA was derived for message sets with \( D_i = P_i \) \( (i = 1, 2, \cdots, n) \). From the above discussion, it has been proved that this WCAU is also applicable to any synchronous message set with arbitrary deadline constraints when using the proposed allocation scheme GLA.

Note that by taking the number of nodes in the network into account, the WCAU expression given by Eq. (5.19) can achieve a higher value than the WCAU expression derived by Malcolm et al [56], as expressed below, for their proposed local allocation (LA\(_M\)) scheme.

\[ U_e^* = \frac{q_{\min} - 1}{q_{\min} + 1} (1 - \alpha) \]

Since Agrawal et al [3] have shown that the WCAU of both the full length allocation (FLA) scheme and the proportional allocation (PA) scheme can asymptotically approach 0%, it is clear that the WCAU of the GLA scheme is no lower than that of any previously published local SBA scheme.

From Eq. (5.19), the WCAU of the GLA scheme is a function of \( D_{\min} \), \( TTRT \) and \( n \), and the following observations can be obtained:

- When \( TTRT < D_{\min} < 2 \cdot TTRT \), \( U_e^* \) becomes 0. That is, for message sets with \( TTRT < D_{\min} < 2 \cdot TTRT \), the WCAU of GLA is 0.
- For the fixed values of \( TTRT \) and \( n \), \( U_e^* \) increases as \( D_{\min} \) increases. \( U_e^* \) approaches \( 1 - \alpha = 1 - \frac{\tau}{TTRT} \), the network’s available utilisation, when \( D_{\min} \) tends to infinity.
- For the fixed values of \( D_{\min} \) and \( n \), \( U_e^* \) may increase as the \( TTRT \) decreases. However, a smaller value of \( TTRT \) makes a larger value of \( \alpha \) and a smaller value of \( 1 - \alpha \) which, in turn, may eventually cause a smaller value of \( U_e^* \) (when the \( TTRT \) is small enough).
Thus, there exists an optimal value which maximises $U^*_e$, given $D_{\text{min}}$ and $n$. This optimal value of $TTRT$ is further evaluated in the next section.

- For the fixed values of $D_{\text{min}}$ and $TTRT$, $U^*_e$ decreases as $n$ increases. When $n$ tends to infinity, $U^*_e$ tends to $\frac{q_{\text{min}}-1}{q_{\text{min}}+1}(1-\alpha)$. Thus, the WCAU of allocation scheme $LA_M$ is the special case of the WCAU of GLA when $n$ tends to infinity.

To test the schedulability of a synchronous message set when using allocation scheme GLA, one can use the WCAU defined in Eq. (5.19), especially at the stage of system design when only a rough estimate of the amount of real-time traffic may be known. However, some message sets whose utilisations are larger than the WCAU defined in Eq. (5.19) can still be scheduled by using the allocation scheme GLA. Thus, in order to get a relatively accurate feasibility testing result, Theorem 11 should be used.

5.3 Selection of $TTRT$

$TTRT$ is an important protocol parameter. As indicated in the last section, there exists an optimal value of $TTRT$ which can maximise the WCAU of $U^*_e$. This section will formally discuss how to choose such a $TTRT$ to maximise $U^*_e$.

It has already been shown that when $TTRT < D_{\text{min}} < 2 \cdot TTRT$, the WCAU $U^*_e$ is always zero no matter how the $TTRT$ is chosen. Thus, in the following discussion on the selection of $TTRT$, only the case of $D_{\text{min}} \geq 2 \cdot TTRT$ is considered.

**Lemma 4.** When $D_{\text{min}} \geq 2 \cdot TTRT$ and Eq. (4.47) is to be used for allocating synchronous bandwidths, the maximum value of the worst case achievable utilisation

$$U^*_e = \frac{q_{\text{min}}-1}{q_{\text{min}}+1}(1-\alpha) = \frac{\left\lfloor \frac{D_{\text{min}}}{TTRT} \right\rfloor - 1}{\left\lfloor \frac{D_{\text{min}}}{TTRT} \right\rfloor + 1 - \frac{1}{n}}(1-\frac{\tau}{TTRT})$$

occurs when

$$\frac{D_{\text{min}}}{TTRT} = \left[\frac{-n + 2 + \frac{\tau}{D_{\text{min}}} + \sqrt{9n^2 + 8n^2 \frac{D_{\text{min}}}{\tau} + 6n \frac{\tau}{D_{\text{min}}} + \frac{\tau^2}{D_{\text{min}}}} - 4n \frac{D_{\text{min}}}{\tau}}{2n + 2 \frac{\tau}{D_{\text{min}}}}\right] = a$$

or

$$\frac{D_{\text{min}}}{TTRT} = \left[\frac{-3n + 2 - \frac{\tau}{D_{\text{min}}} + \sqrt{9n^2 + 8n^2 \frac{D_{\text{min}}}{\tau} + 6n \frac{\tau}{D_{\text{min}}} + \frac{\tau^2}{D_{\text{min}}}} - 4n \frac{D_{\text{min}}}{\tau}}{2n + 2 \frac{\tau}{D_{\text{min}}}}\right] = b$$

where $0 \leq a - b \leq 1$ and $a \geq 2$.

A proof of this lemma can be found in Appendix J. Most calculations are made using MATLAB due to the complex expressions. The result given by Lemma 4 implies two cases: the
case of \(a = b\), in which the maximum value of \(U_e^*\) happens when \(\frac{D_{\min}}{TTRT}\) has a unique value, i.e., \(\frac{D_{\min}}{TTRT} = a = b\); and the case of \(a = b + 1\), where both \(\frac{D_{\min}}{TTRT} = a\) and \(\frac{D_{\min}}{TTRT} = b\) can maximise \(U_e^*\).

Based on this lemma, the following theorem, which gives the value of \(TTRT\) which can maximise \(U_e^*\), can be obtained.

**Theorem 15.** When \(D_{\min} \geq 2 \cdot TTRT\), the Worst Case Achievable Utilisation

\[
U_e^* = \frac{q_{\min} - 1}{q_{\min} + 1 - \frac{\tau}{\min}} (1 - \alpha)
\]

for the local synchronous bandwidth allocation scheme defined by Eq. (4.47) is maximised if

\[
TTRT = \frac{D_{\min}}{-3n + 2 - \frac{\tau}{\min} + \sqrt{9n^2 + 8n^2 \frac{D_{\min}}{\tau} + 6n \frac{\tau}{\min} + \frac{\tau^2}{D_{\min}} - 4n \frac{\tau}{\min}}} + 2n + 2 \frac{\tau}{\min}
\]

(5.20)

**Proof.** By Lemma 4, the maximum value of \(U_e^*\) occurs when

\[
\frac{D_{\min}}{TTRT} = \left[\frac{-n + 2 + \frac{\tau}{\min} + \sqrt{9n^2 + 8n^2 \frac{D_{\min}}{\tau} + 6n \frac{\tau}{\min} + \frac{\tau^2}{D_{\min}} - 4n \frac{\tau}{\min}}}{2n + 2 \frac{\tau}{\min}}\right]
\]

or

\[
\frac{D_{\min}}{TTRT} = \left[\frac{-3n + 2 - \frac{\tau}{\min} + \sqrt{9n^2 + 8n^2 \frac{D_{\min}}{\tau} + 6n \frac{\tau}{\min} + \frac{\tau^2}{D_{\min}} - 4n \frac{\tau}{\min}}}{2n + 2 \frac{\tau}{\min}}\right]
\]

Thus, the value of \(TTRT\) that maximises \(U_e^*\) is either

\[
TTRT = \frac{D_{\min}}{-n + 2 + \frac{\tau}{\min} + \sqrt{9n^2 + 8n^2 \frac{D_{\min}}{\tau} + 6n \frac{\tau}{\min} + \frac{\tau^2}{D_{\min}} - 4n \frac{\tau}{\min}}} + 2n + 2 \frac{\tau}{\min}
\]

or

\[
TTRT = \frac{D_{\min}}{-3n + 2 - \frac{\tau}{\min} + \sqrt{9n^2 + 8n^2 \frac{D_{\min}}{\tau} + 6n \frac{\tau}{\min} + \frac{\tau^2}{D_{\min}} - 4n \frac{\tau}{\min}}} + 2n + 2 \frac{\tau}{\min}
\]

It is better to choose a larger value for \(TTRT\). This gives a larger available utilisation \(1 - \frac{\tau}{TTRT}\). Hence \(TTRT\) should be chosen as in Eq. (5.20).

Because the WCAU of allocation scheme \(LA_M\) is a special case of the WCAU of allocation scheme \(GLA\) when \(n\) tends to infinity, as described in Section 5.2, the value of \(TTRT\) which maximises the WCAU of scheme \(LA_M\) must be a special case of the value of \(TTRT\) which maximises the WCAU of scheme \(GLA\) when \(n\) tends to infinity. This can be shown as follows.
As shown in [55], the value of $TTRT$ which can maximise the WCAU of scheme $LM$ is

\[
\frac{D_{\text{min}}}{\frac{-3+\sqrt{9+\frac{2D_{\text{min}}}{\tau}}}{2}}
\]

(5.21)

When \( n \) tends to zero, there is, from Eq. (5.20)

\[
\lim_{n \to \infty} \frac{D_{\text{min}}}{\frac{-3+\sqrt{9n^2+8n^2D_{\text{min}}+6nD_{\text{min}}+\frac{\tau}{D_{\text{min}}}+4nD_{\text{min}}}}{2n+\frac{\tau}{D_{\text{min}}}}}
\]

\[= \lim_{n \to \infty} \frac{D_{\text{min}}}{\frac{-3n+\sqrt{9n^2+8n^2D_{\text{min}}}}{2n}}
\]

\[= \frac{D_{\text{min}}}{\frac{-3+\sqrt{9+\frac{2D_{\text{min}}}{\tau}}}{2}}
\]

which is exactly the same as (5.21).

The impact of different $TTRT$ values on the WCAU of $U_c^*$ can be demonstrated simply through the following example.

Consider the case where $D_{\text{min}} = 4$, $\tau = 0.05$, $n = 10$. By Eq. (5.20), when $TTRT = 4/12 = 0.33$, $U_c^*$ can be maximised to be 72.40%. However, when $TTRT$ is too small, say $TTRT = 0.1$, $U_c^*$ can be as low as 47.62%. Also, when $TTRT$ is too large, say $TTRT = 2$, $U_c^*$ can be as low as 33.59%. Thus, it is important to select an appropriate value of $TTRT$ to achieve a higher WCAU.

### 5.4 Summary

The WCAU can be used to test the schedulability of a synchronous message set, especially at the stage of system design when only a rough estimate of the amount of real-time traffic may be known. This chapter derives the WCAU of the proposed new SBA schemes. As a performance metric, this derived WCAU is compared with those of other published local SBA schemes. The result also reveals the superiority of the proposed schemes on guaranteeing the transmission of synchronous messages. This chapter also assesses the impact of $TTRT$ on the WCAU, and an optimal value of $TTRT$, which can maximise the WCAU, is proposed.

\[\text{Here, it also verifies that when } \frac{D_{\text{min}}}{TTRT} = 12 \text{ is an integer, } U_c^* \text{ can be maximised.}\]
Chapter 6

Simulation and Numerical Examples

In this chapter, the new local SBA scheme of GLA/IGLA is compared with other published local SBA schemes, through simulation and numerical examples.

This chapter is organised as follows: In Section 6.1 different local SBA schemes are compared from the perspective of their performance in guaranteeing synchronous message sets. The comparison is made in two groups, based on whether $D_{\text{min}}$ of the studied synchronous message sets is smaller than $2 \cdot TTRT$ or no less than $2 \cdot TTRT$. The WCAUs of these schemes are also compared in Section 6.2.

6.1 Guaranteeing Synchronous Message Sets

This section focuses on the capability of local SBA schemes to guarantee synchronous message sets. For the convenience of discussion and easy reference, the local SBA schemes are listed using their short-hand notations\(^1\). According to their usage to synchronous message sets, these schemes are grouped into three categories: (1) local SBA schemes applicable to synchronous message sets with $D_i = P_i$, (2) local SBA schemes applicable to synchronous message sets of arbitrary deadline constraint with $D_i \geq 2 \cdot TTRT$, and (3) local SBA schemes applicable to any theoretical synchronous message set. The synchronous bandwidths allocated by different schemes are respectively calculated through their corresponding definition expressions.

An allocation scheme is said to be able to guarantee a message set if it can produce an allocation $\vec{H}$ which meets the protocol constraint, the deadline constraint and the buffer constraint.

\(^1\)The definition of each local allocation scheme is also provided for an easy understanding of this chapter.
Different allocation schemes may use different feasibility testing methods, which are listed along with their definitions for easy reference. There is no need to test the buffer constraint because every SBA scheme listed below can satisfy the buffer constraint.

Local SBA schemes applicable to synchronous message sets with $D_i = P_i$

- **FLA** (Full Length Allocation scheme [3]):

  \[ H_i = C_i \]

  **Schedulability test:** For a message set with $D_{min} \geq TTRT$, test the protocol constraint using $\sum_{i=1}^{n} H_i \leq TTRT - \tau$ and complement the testing of the deadline constraint using $\sum_{i=1}^{n} H_i \leq D_{min} - TTRT - \tau$.

- **PA** (Proportional Allocation scheme [3]):

  \[ H_i = \frac{C_i}{P_i} (TTRT - \tau) \]

  **Schedulability test:** For a message set with $D_{min} \geq 2 \cdot TTRT$, test the protocol constraint using $\sum_{i=1}^{n} H_i \leq TTRT - \tau$, and test the deadline constraint using (4.38) (when $k = 1$), i.e., (6.1).

  \[
  (m_i - 1) \cdot H_i + \max\{0, D_i \} - \{ m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - \left\lfloor \frac{m_i}{n+1} \right\rfloor \cdot [TTRT - (\sum_{j=1}^{n} H_j + \tau)] - H_i \} \geq C_i
  \]

  (6.1)

  where $m_i$ is an integer ($m_i \geq 2$) which makes the inequality of $I(m_i - 1) \leq D_i < I(m_i)$ hold, and must be either

  \[
  m_i = \left\lfloor \frac{D_i \cdot (n + 1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor
  \]

  or

  \[
  m_i = \left\lfloor \frac{D_i \cdot (n + 1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor - 1
  \]

- **LA** (Local Allocation scheme [2]):

  \[ H_i = \frac{C_i}{q_i - 1} \quad \text{where} \quad q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \]

  **Schedulability test:** For a message set with $D_{min} \geq 2 \cdot TTRT$, test the protocol constraint using $\sum_{i=1}^{n} H_i \leq TTRT - \tau$. There is no need to test the deadline constraint as it holds automatically under the protocol constraint.
• **NLA** (New Local Allocation scheme [94]):

\[
H_i = \begin{cases} 
C_i & \text{if } TTRT < D_i < 2 \cdot TTRT \\
\frac{C_i}{q_i - 1} - \frac{1}{q_i} \max \{D_i - [(q_i + 1)TTRT - \frac{C_i}{q_i}],0\} & \text{if } D_i \geq 2 \cdot TTRT \\
\end{cases}
\]

where \( q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \)

**Schedulability test:** Same as the FLA scheme.

Local SBA schemes applicable to synchronous message sets of arbitrary deadline constraints with \( D_{min} \geq 2 \cdot TTRT \)

• **LA_M** (Local Allocation scheme [56]):

\[
H_i = \max \left( \frac{2 \cdot TTRT}{P_i} , 1 \right) \cdot C_i \\
\text{where } q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor 
\]

(6.2)

**Schedulability test:** Same as the \( LA_A \) scheme.

• **LA_Z** (Local Allocation scheme, proposed by Zheng et al [96]):

\[
H_i = a_i \cdot C_i
\]

where \( a_i \) is determined as below: let \( x = \frac{D_i}{TTRT} - 1 \) and \( y = \frac{P_i}{TTRT} \), then,

\[
a_i = \begin{cases} 
\frac{1}{|x|} & \text{if } y \geq |x| \geq 1 \\
\frac{1}{y} & \text{if } y \leq 1 \text{ and } x \geq 2 \\
\end{cases}
\]

and

\[
a_i \leq \begin{cases} 
1 + \frac{2 - x}{y} & \text{if } y \leq 1 \text{ and } 1 \leq x < 2 \\
\frac{1}{\lfloor y \rfloor} & \text{if } 1 < y < |x| \\
\end{cases}
\]

**Schedulability test:** Same as the \( LA_A \) scheme.
Local SBA schemes applicable to any theoretical synchronous message sets

- **GLA** (Generalised Local Allocation scheme, proposed in this research):

\[
H_i = \begin{cases} 
  C_i & \text{if } TTRT < D_i \leq P_i \\
  (\frac{TTRT}{P_i} + 1)C_i & \text{or } TTRT \leq P_i < D_i \\
  \text{or returned allocation from Get_H} & \\
  \frac{C_i}{q_i-1} & \text{if } 0 < P_i < TTRT < D_i \\
  \frac{\max(\frac{q_i TTRT}{P_i} - 1) C_i}{q_i - 1} - \frac{1}{q_i} \cdot \max[D_i - (q_i + 1)TTRT + \frac{\max(\frac{q_i TTRT}{P_i} - 1) C_i}{q_i - 1}, 0] & \text{if } q_i = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \\
  0 & \text{if } q_i \neq \lfloor \frac{P_i}{TTRT} \rfloor + 1 \\
\end{cases} \\
\]

where \( q_i = \lfloor \frac{D_i}{TTRT} \rfloor \).

**Schedulability test:** Same as the FLA scheme.
IGLA (Improved Generalised Local Allocation scheme, proposed in this research):

\[
H_i = \begin{cases} 
C_i & \text{if } TTRT < D_i \leq P_i \\
(D_i + 1)C_i & \text{or } TTRT \leq P_i < D_i \\
\text{or returned allocation from Get}_i & \\
\frac{C_i}{q_i - 1} & \text{if } 0 < P_i < TTRT < D_i
\end{cases}
\]

where \( q_i = \lfloor \frac{P_i}{TTRT} \rfloor \).

Schedulability test: Same as the FLA scheme.

The performance of SBA schemes is demonstrated and compared by simulation and numerical examples. One typical way to measure the performance of a guarantee test is to randomly generate a huge number of synchronous message sets and then find out the percentage of schedulable message sets which pass the test [8]. In this research, the percentage of infeasible sets is tested. The "infeasible sets" means the synchronous message sets which fail to pass the test and therefore are deemed as "unschedulable" under the scheme considered. Let \( P \) be such a percentage, i.e.,

\[
P = \frac{\text{The number of infeasible synchronous message sets}}{\text{The total number of tested synchronous message sets}} \times 100\%
\]

and a synchronous message set with the utilisation \( U_e(M) \) can be generated using the following method. To every synchronous message stream \( S_i = (C_i, P_i, D_i) \) from this message set, it will

- generate utilisation \( U_i \) (\( U_i \) is defined as \( \frac{C_i}{\min(C_i, P_i, D_i)} \)) randomly with a uniform distribution using the method proposed in [8]. (The method also makes sure that \( \sum_{i=1}^{n} U_i = U_e(M) \).)
function vectU = UtilisationGen(n, Ue)
    vectU = zeros(1, n);
    sumU = Ue;
    for i=1:n-1,
        nextSumU = sumU.*rand^(1/(n-i));
        vectU(i) = sumU - nextSumU;
        sumU = nextSumU;
    end
    vectU(n) = sumU;

function vectP = PeriodGen(n, Pi_min, Pi_max)
    vectP = rand(1, n)*(Pi_max - Pi_min) + Pi_min;

function vectD = DeadlineGen(n, Di_min, Di_max)
    vectD = rand(1, n)*(Di_max - Di_min) + D_min;

function vectC=ComputationTimeGen(vectP, vectD, vectU)
    minScale = min(vectP, vectD);
    vectC = vectU.*minScale;

Figure 6.1: Matlab code for generating a synchronous message set with \( U_e(M) \)

- generate period \( P_i \) randomly with a uniform distribution in the interval \([P_{i\text{min}}, P_{i\text{max}}]\).
- generate relative deadline \( D_i \) randomly with a uniform distribution in the interval \([D_{i\text{min}}, D_{i\text{max}}]\). For the special case of \( D_i = P_i \), there is no need for this step.
- compute transmission time \( C_i \) as \( C_i = U_i \times \min(P_i, D_i) \). For the special case of \( D_i = P_i \), use \( C_i = U_i \times P_i \) to generate \( C_i \).

With this method, the generated synchronous message sets are uniformly distributed and thus can provide a reasonable evaluation of the performance of SBA schemes. Fig. 6.1 lists the relative Matlab code.

Now, one simulation for a SBA scheme can be designed to measure the percentage of infeasible sets with a varying utilisation of \( U_e(M) \) from \( U_e(M)_{\text{min}} \) to \( U_e(M)_{\text{max}} \) for a step of \( U_e(M)_{\text{step}} \). For each value of \( U_e(M) \), a large number \( N (N \geq 10000) \) of synchronous message sets will be generated and tested to gain a reasonable result.

The other commonly employed way to measure and compare the performance of SBA schemes is to use numerical examples, which is also adopted in this study, and will be discussed later in this chapter.
6.1.1 Synchronous Message Sets with $TTRT < D_{\text{min}} < 2 \cdot TTRT$

The FLA scheme, the NLA scheme, the GLA scheme and its improved version IGLA scheme can all be used for this type of message set. The FLA scheme and the NLA scheme can only be used for message sets with $D_i = P_i$ while the GLA scheme and the IGLA scheme are both suitable for message sets with an arbitrary deadline constraint. In the case of $D_i = P_i$ ($i = 1, 2, \cdots, n$), NLA, GLA and IGLA are exactly the same. In the case of $TTRT < D_i = P_i < 2 \cdot TTRT$ ($i = 1, 2, \cdots, n$), the above four schemes are exactly the same.

Fig. 6.2 shows that there do exist synchronous message sets which are schedulable under $D_{\text{min}} < 2 \cdot TTRT$. The percentage of infeasible message sets of the GLA scheme is less than that of the NLA scheme or the FLA scheme because it can be used for message sets with arbitrary deadline constraints including those with $D_i = P_i$. Table 6.1 shows a message set with $TTRT < D_{\text{min}} < 2 \cdot TTRT$ which can be guaranteed by schemes GLA, IGLA, NLA and FLA while Table 6.2 gives a message set which can only be guaranteed by schemes GLA and IGLA.

Two methods are used in the GLA scheme to allocate synchronous bandwidth $H_i$ to message stream $S_i$ with $TTRT < D_{\text{min}} < 2 \cdot TTRT$ and $0 < P_i < TTRT < D_i$. One is through calculating $(\frac{TTRT}{P_i} + 1)C_i$ and the other is by executing algorithm Get_H. However, as depicted in Fig. 6.3, the percentage of infeasible message sets when using $(\frac{TTRT}{P_i} + 1)C_i$ winds around that when using Get_H. A smaller $H_i$ can be obtained by using $(\frac{TTRT}{P_i} + 1)C_i$ for some message streams, but by using Get_H for other message streams. The numerical examples to show these situations are given in Table 6.3 and Table 6.4. The message set in Table 6.3 can be guaranteed by using $(\frac{TTRT}{P_i} + 1)C_i$ but not Get_H while the message set in Table 6.4 can be guaranteed by using Get_H but not $(\frac{TTRT}{P_i} + 1)C_i$.

The reasons behind the differences are now discussed.

As discussed in subsection 4.1.3, the synchronous bandwidth $H_i$ allocated to message stream $S_i$ with $TTRT < D_{\text{min}} < 2 \cdot TTRT$ and $0 < P_i < TTRT < D_i$ is deduced from

$$H_i \geq \left\lceil \frac{TTRT - H_i}{P_i} \right\rceil + 1 \cdot C_i$$

which is only a sufficient condition to satisfy the deadline constraint. Thus, even by using Get_H to obtain a minimum value of $H_i$ to satisfy the above inequality, some message sets may still be unschedulable because of the violation of $\sum_{i=1}^{n} H_i \leq D_{\text{min}} - TTRT - \tau$. A smaller feasible $H_i$, which can meet the deadline constraint for $S_i$, may still exist.

On the other side, $H_i = (\frac{TTRT}{P_i} + 1)C_i$ is a result obtained through observation of the form of the above inequality and is proved to satisfy the deadline constraint. It has no relationship

\(^2\)Also the IGLA scheme because it is same as the GLA scheme for the message stream $S_i$ with $TTRT < D_i < 2 \cdot TTRT$.
with the above inequality and it is also a sufficient result to meet the deadline constraint. Thus, it has the same problem with \( \text{Get}_H \).

\[ H_i = \left( \frac{TTRT}{P_i} + 1 \right) C_i \] and \( \text{Get}_H \) are independent from each other and are both sufficient conditions to satisfy the deadline constraint. Thus, \( H_i \) obtained by \( \left( \frac{TTRT}{P_i} + 1 \right) C_i \) may be bigger or smaller than that obtained by \( \text{Get}_H \).

However, Fig. 6.3 also suggests that the difference between these two methods is slight. Thus, in practice, users can first employ \( \left( \frac{TTRT}{P_i} + 1 \right) C_i \) for an easy and quick calculation of the synchronous bandwidth. If the result cannot meet the condition \( \sum_{i=1}^{n} H_i \leq D_{\text{min}} - TTRT - \tau \), then \( \text{Get}_H \) can be employed to check if a feasible result can be obtained. That is, \( \min \left\{ \left( \frac{TTRT}{P_i} + 1 \right) C_i, \text{Get}_H \right\} \) will be used instead, except in Fig. 6.3, Table 6.3 and Table 6.4.

### 6.1.2 Synchronous Message Sets with \( D_{\text{min}} \geq 2 \cdot TTRT \)

For the synchronous message sets with \( D_{\text{min}} \geq 2 \cdot TTRT \), first, allocations to the message sets with \( D_i = P_i \) \((i = 1, 2, \cdots, n)\) by different local SBA schemes will be compared. Then, performances of different local SBA schemes for the message sets with arbitrary deadline constraint will be compared.

Fig. 6.4 shows the different percentage of infeasible message sets when using different allocation schemes to allocate synchronous bandwidths to message sets with \( D_i = P_i \) \((i = 1, 2, \cdots, n)\). Several observations can be made from Fig. 6.4:

- Among all the local SBA scheme under consideration, the GLA scheme\(^3\) performs best, with the lowest percentage of infeasible message sets. The \( LA_M \) scheme (also the \( LA_A \) scheme and the \( LA_Z \) scheme) has a similar trend as the GLA scheme but a higher percentage of infeasible message sets. The FLA scheme and the PA scheme, especially the PA scheme, can hardly be used in practice due to their high percentages of infeasible message sets and the close-to-zero worst case achievable utilisation.

- The WCAU of the \( LA_M \) scheme, in this simulation, is \( U_e^* = \frac{q_{\text{min}} - 1}{q_{\text{min}} + 1} (1 - \alpha) = \frac{2 - 1}{2 + 1} (1 - 0) = 33.3\% \). Fig. 6.4 shows that all message sets with \( U_e(M) \leq 33.3\% \) can be scheduled by the \( LA_M \) scheme. Similarly, the WCAU of GLA scheme here is \( U_e^* = \frac{q_{\text{min}} - 1}{q_{\text{min}} + 1} \left( 1 - \frac{2 - 1}{2 + 1} \right) (1 - 0) = 35.7\% \). As illustrated by Fig. 6.4, all message sets with \( U_e(M) \leq 35.7\% \) can be guaranteed by the GLA scheme. Fig. 6.4 also confirms that the WCAU of the \( LA_M \) scheme is not much worse than that of the GLA scheme.

---

\(^3\)Also the IGLA scheme and the NLA scheme because they are the same for synchronous message streams with \( D_i \geq 2 \cdot TTRT \) and \( D_i = P_i \) \((i = 1, 2, \cdots, n)\)
The percentage of infeasible message sets increases as the utilisation of synchronous message sets increases. This situation can be explained from two perspectives. First, if there are not many nodes in the network, the high utilisation implies that some message streams have large transmission time $C_i$. The large transmission time demands more synchronous bandwidth to be allocated, which in turn may lead to a large possibility of violating the protocol constraint. Secondly, if there are many nodes in the network, though every stream may require only a small amount of synchronous bandwidth, the total bandwidth required by the entire network may still be very large. Thus, the message set may be unschedulable because of this large required bandwidth.

Tables 6.5 to 6.11 give some numerical examples to compare the performance of different local SBA schemes used for message sets with $D_{min} \geq 2 \cdot TTRT$ and $D_i = P_i$ ($i = 1, 2, \cdots, n$). Table 6.5 gives an example where the message set can be guaranteed by all the local allocation schemes considered here. In general, the $LA_M$ scheme (as well as schemes $LA_A$ and $LA_Z$) and the $GLA$ scheme (as well as schemes $IGLA$ and $NLA$) are superior to schemes $FLA$ and $PA$, as demonstrated in Table 6.6 and Table 6.7. However, there do exist some synchronous message sets which can be guaranteed even by the $PA$ scheme but fail to be guaranteed by the $LA_M$ scheme. Table 6.8 gives such an example. However, this message set can be guaranteed by the $GLA$ scheme. Table 6.9 and Table 6.10 give two synchronous message sets which can only be guaranteed by $GLA$, $IGLA$ or $NLA$. Some message sets may fail to be guaranteed by any of the local SBA schemes considered here, as shown in Table 6.11.

The next comparison is made between the $GLA$ scheme, the $IGLA$ scheme, the $LA_M$ scheme and the $LA_Z$ scheme. All of these local schemes can be used for message sets with arbitrary deadline constraints under $D_{min} \geq 2 \cdot TTRT$.

Fig. 6.5 illustrates four simulation results of the comparison between the $GLA$ scheme, the $LA_M$ scheme and the $LA_Z$ scheme, with the different configurations of period and deadline generation ranges. This group of simulations suggests that the overall performance of the $GLA$ scheme is superior than either the $LA_M$ scheme or the $LA_Z$ scheme. The $GLA$ scheme is always superior to the $LA_M$ scheme. The comparison with the $LA_Z$ scheme is more complex. Although the $LA_Z$ scheme shows a slightly better performance, this is only when deadlines of message streams are relative small or when the utilisation of the message set is big. If the simulated deadline generation range becomes wider, this slightly better performance becomes unnoticeable or an even worse result is observed. As confirmed by Fig. 6.5(b), the $LA_Z$ scheme and the $LA_M$ scheme both perform quite similarly except when the utilisation of the message set approaches one. As shown from Fig. 6.5(a) to Fig. 6.5(d), except when the utilisation is
near one, the performance of the $LAM$ scheme is first weaker and then better than that of the $LAZ$ scheme. Compared with the GLA scheme, the $LAZ$ shows a weaker performance in most situations. Only when the deadlines are relative small and when the utilisation is big, does the $LAZ$ show a slightly better performance, as illustrated in Fig. 6.5(a). However, by using the improved version of the GLA scheme, i.e., the IGLA scheme, the observation of this tiny performance improvement becomes difficult, as shown in Fig. 6.6. Fig. 6.6 shows four simulation results using the same configurations as those used in Fig. 6.5. The same simulation results of the $LAM$ scheme and the $LAZ$ scheme from these two groups of figures also show that the simulation method used here can accurately evaluate performances of different schemes even if message sets are randomly generated.

More numerical examples are given in Tables 6.12 to 6.18 to compare the performance of schemes $LAM$, $LAZ$, GLA and IGLA on guaranteeing synchronous message sets with $D_{min} \geq 2 \cdot TTRT$ and arbitrary deadline constraint. Table 6.12 gives a synchronous message set which can be guaranteed by all four schemes. Table 6.13 and Table 6.14 give two examples that cannot be guaranteed by the $LAZ$ scheme. However, as suggested by the simulation result, in a few cases, the $LAZ$ scheme can guarantee a message set which cannot be scheduled by scheme $LAM$ or GLA. Such an example is given in Table 6.15. Although the GLA scheme fails to offer a feasible allocation $\vec{H}$ for this synchronous message set, the IGLA scheme can. Table 6.16 shows a message set that cannot be guaranteed by the schemes $LAM$ and $LAZ$ but can be guaranteed by the schemes GLA and IGLA. The message set list in Table 6.17 can only be guaranteed by the IGLA scheme. For the message set in Table 6.18, none of these four local SBA schemes can offer a feasible $\vec{H}$ because of the violation of the protocol constraint.

As a conclusion to this comparison, the performances of the GLA scheme, the IGLA scheme, the $LAM$ scheme and the $LAZ$ scheme when used for synchronous message sets with $D_{min} > TTRT$ but arbitrary deadline constraint are compared. Fig.6.7 shows the expected results. The $LAM$ scheme and the $LAZ$ scheme both perform similarly to each other. In general, the GLA scheme and the IGLA scheme have a better performance than the $LAM$ scheme and the $LAZ$ scheme, especially when the utilisation of a message set is not too big, with the IGLA scheme showing an even more outstanding performance. Table 6.19 gives an example message set with $D_{min} > TTRT$ which can only be scheduled by schemes GLA and IGLA. The message set in Table 6.20, however, cannot be guaranteed by any of the listed schemes, including GLA and IGLA, because of the violation of the deadline constraint.
6.2 The Worst Case Achievable Utilisation

In this section, the Worst Case Achievable Utilisation of different schemes is compared. The WCAU of the GLA/IGLA scheme is the same as that of the NLA scheme, but can be used for any general synchronous message set with arbitrary deadline constraints. This section only compares the WCAU of the new schemes to that of the LA$_M$ scheme because the WCAU of the LA$_M$ scheme is the best result reported in literature and the LA$_M$ scheme can be used for a general message set with $D_{\text{min}} \geq 2 \cdot TTRT$. The WCAU of the LA$_M$ scheme is

$$U_e^* = \frac{q_{\text{min}} - 1}{q_{\text{min}} + 1} (1 - \alpha) \quad \text{where } q_{\text{min}} \geq 2$$

and the WCAU of the GLA/IGLA scheme is

$$U_e^* = \frac{q_{\text{min}} - 1}{q_{\text{min}} + 1 - \frac{1}{n}} (1 - \alpha) \quad \text{where } q_{\text{min}} \geq 1$$

The difference between the WCAU of the LA$_M$ scheme and that of the GLA/IGLA scheme is caused by the number of nodes $n$ in the network. Fig. 6.8 shows such a difference. The WCAU of the GLA/IGLA scheme is always larger than that of the LA$_M$ scheme. This gives the GLA/IGLA scheme a better chance of guaranteeing more synchronous message sets when judged only by the WCAU. However, this difference gets smaller as $n$ increases.

Fig. 6.9 shows the relationship between the TTRT and the WCAU of the LA$_M$ scheme and the GLA/IGLA scheme, for several different values of $D_{\text{min}}$. From Fig. 6.9, the following observations can be made:

- The relationships between the TTRT and the WCAU of the GLA/IGLA scheme and LA$_M$ scheme are similar to each other. Both have the optimal value of TTRT which can maximise the WCAU. As $D_{\text{min}}$ increases, the optimal TTRT increases for both schemes.

- The WCAU of the GLA/IGLA scheme is always larger than that of the LA$_M$ scheme, given specific values of $D_{\text{min}}$ and TTRT. Also, the maximum value of the WCAU of the GLA/IGLA scheme is larger than that of the LA$_M$ scheme when $D_{\text{min}}$ is fixed.
Figure 6.2: Percentage of infeasible message sets ($TTRT < D_{\text{min}} < 2 \cdot TTRT$): $n = 5$, $TTRT = 10.0$, $\tau = 0.0$, $[P_{1,\text{min}}, P_{1,\text{max}}] = [10.0, 20.0]$, $[D_{1,\text{min}}, D_{1,\text{max}}] = [10.5, 19.5]$, $[U_{e(M),\text{min}}, U_{e(M),\text{max}}] = [0.1, 1.0]$, $U_{e(M),\text{step}} = 0.025$, $N = 10000$.

Table 6.1: Synchronous message set A guaranteed by GLA, IGLA, NLA and FLA ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
<th>FLA</th>
<th>NLA</th>
<th>GLA</th>
<th>IGLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$ $C_i$ $D_i$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>30.00 $7.50$ $30.00$</td>
<td>7.50</td>
<td>3.75</td>
<td>3.70</td>
<td>3.70</td>
</tr>
<tr>
<td>2</td>
<td>19.50 $1.00$ $19.50$</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Protocol constraint met?</td>
<td>Yes Yes Yes Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadline constraint met?</td>
<td>Yes Yes Yes Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message set guaranteed?</td>
<td>Yes Yes Yes Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

103
Figure 6.3: Percentage of infeasible message sets \((TTRT < D_{\text{min}} < 2 \cdot TTTR)\): \(n = 3\), \(TTRT = 10.0\), \(\tau = 0.0\), \([P_{i,\text{min}}, P_{i,\text{max}}] = [5.0, 30.0]\), \([D_{i,\text{min}}, D_{i,\text{max}}] = [10.5, 19.5]\), \([U_e(M)_{\text{min}}, U_e(M)_{\text{max}}] = [0.3, 0.6]\), \(U_e(M)_{\text{step}} = 0.025\), \(N = 10000\).

Table 6.2: Synchronous message set B guaranteed by GLA and IGLA \((TTRT = 10.0\) and \(\tau = 0.0)\)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth (H_i) allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) (P_i) (C_i) (D_i) (FLA) (NLA) (GLA) (IGLA)</td>
<td></td>
</tr>
<tr>
<td>1 17.00 1.60 17.00 1.60 1.60 1.60</td>
<td></td>
</tr>
<tr>
<td>2 5.00 1.00 18.00 N/A N/A 3.00 3.00</td>
<td></td>
</tr>
<tr>
<td>3 50.00 3.00 60.00 N/A N/A 0.60 0.60</td>
<td></td>
</tr>
<tr>
<td>Protocol constraint met?</td>
<td>N/A N/A Yes Yes</td>
</tr>
<tr>
<td>Deadline constraint met?</td>
<td>N/A N/A Yes Yes</td>
</tr>
<tr>
<td>Message set guaranteed?</td>
<td>No No Yes Yes</td>
</tr>
</tbody>
</table>
Figure 6.4: Percentage of infeasible message sets ($D_{\text{min}} \geq 2 \cdot TTIRT$ and $D_i = P_i$): $n = 5$, $TTIRT = 10.0$, $\tau = 0.0$, $[D_{\text{min}}, D_{\text{max}}] = [20.5, 100.0]$, $[U_e(M)_{\text{min}}, U_e(M)_{\text{max}}] = [0.1, 1.0]$, $U_e(M)_{\text{step}} = 0.025$, $N = 10000$.

Table 6.3: Synchronous message set $C$ guaranteed by $GLA((\frac{TTIRT}{P_i} + 1)C_i)$ ($TTIRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>$i$</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$D_i$</th>
<th>$GLA((\frac{TTIRT}{P_i} + 1)C_i)$</th>
<th>$GLA(\text{Get}_H)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22.00</td>
<td>4.70</td>
<td>18.80</td>
<td>4.70</td>
<td>4.70</td>
</tr>
<tr>
<td>2</td>
<td>6.50</td>
<td>1.05</td>
<td>17.80</td>
<td>2.67</td>
<td>3.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Protocol constraint met?</strong></td>
<td><strong>Yes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Deadline constraint met?</strong></td>
<td><strong>Yes</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Message set guaranteed?</strong></td>
<td><strong>Yes</strong></td>
</tr>
</tbody>
</table>

105
Figure 6.5: Percentage of infeasible message sets ($D_{min} \geq 2 \cdot TTRT$): $n = 5$, $TTRT = 10.0$, $	au = 0.0$, $[U_e(M)_{min}, U_e(M)_{max}] = [0.1, 1.0]$, $U_e(M)_{step} = 0.025$, $N = 10000$.

Table 6.4: Synchronous message set D guaranteed by $GLA(\text{Get}_H)$ ($TTRT = 10.0$ and $	au = 0.0$)

<table>
<thead>
<tr>
<th>i</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$D_i$</th>
<th>$GLA((\frac{TTRT}{P_i} + 1)C_i)$</th>
<th>$GLA(\text{Get}_H)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14.00</td>
<td>3.50</td>
<td>16.00</td>
<td>3.50</td>
<td>3.50</td>
</tr>
<tr>
<td>2</td>
<td>7.90</td>
<td>1.20</td>
<td>18.00</td>
<td>2.72</td>
<td>2.41</td>
</tr>
</tbody>
</table>

Protocol constraint met? | Yes | Yes |
Deadline constraint met? | No | Yes |
Message set guaranteed? | No | Yes |
Figure 6.6: Percentage of infeasible message sets ($D_{\min} \geq 2 \cdot TTRT$): $n = 5$, $TTRT = 10.0$, $\tau = 0.0$, $[U_e(M)_{\min}, U_e(M)_{\max}] = [0.1, 1.0]$, $U_e(M)_{\text{step}} = 0.025$, $N = 10000$.

Table 6.5: Synchronous message set E with $P_i = D_i \geq 2 \cdot TTRT$ guaranteed by all the local SBA schemes ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>35.00</td>
</tr>
<tr>
<td>2</td>
<td>60.00</td>
</tr>
<tr>
<td>3</td>
<td>45.00</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

107
Figure 6.7: Percentage of infeasible message sets ($D_{\min} > TTRT$): where $n = 10$, $TTRT = 10.0$, $\tau = 0.02$, $[P_{\omega \min}, P_{\omega \max}] = [10.5, 100.0]$, $[D_{\omega \min}, D_{\omega \max}] = [10.5, 100.0]$, $[U_e(M)_{\min}, U_e(M)_{\max}] = [0.1, 1.0]$, $U_e(M)_{\text{step}} = 0.025$, $N = 10000$.

Figure 6.8: Comparison of $U^*_e (\alpha = 0)$. 
Figure 6.9: $U_e^*$ versus TTRT ($n = 3$, $\tau = 0.05$).

Table 6.6: Synchronous message set $F$ with $P_i = D_i \geq 2 \cdot TTRT$ guaranteed by $LA_A/LA_M/LA_Z$ and $GLA/IGLA/NLA$ ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>185.00</td>
</tr>
<tr>
<td>2</td>
<td>70.00</td>
</tr>
<tr>
<td>3</td>
<td>30.00</td>
</tr>
</tbody>
</table>

Protocol constraint met? No Yes Yes Yes
Deadline constraint met? Yes No Yes Yes
Message set guaranteed? No No Yes Yes
Table 6.7: The other example synchronous message set G with $P_i = D_i \geq 2 \cdot TTRT$ guaranteed by $LAA/LAM/LAZ$ and $GLA/IGLA/NLA$ ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>60.00</td>
</tr>
<tr>
<td>2</td>
<td>200.00</td>
</tr>
<tr>
<td>3</td>
<td>120.00</td>
</tr>
<tr>
<td>Protocol constraint met?</td>
<td>No</td>
</tr>
<tr>
<td>Deadline constraint met?</td>
<td>Yes</td>
</tr>
<tr>
<td>Message set guaranteed?</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 6.8: Example synchronous message set H with $P_i = D_i \geq 2 \cdot TTRT$ unscheduled by $LAA/LAM/LAZ$ but guaranteed by $PA$ ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>84.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
</tr>
<tr>
<td>Protocol constraint met?</td>
<td>No</td>
</tr>
<tr>
<td>Deadline constraint met?</td>
<td>Yes</td>
</tr>
<tr>
<td>Message set guaranteed?</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 6.9: Synchronous message set I with $P_i = D_i \geq 2 \cdot TTRT$ only guaranteed by \textit{GLA/IGLA/NLA} ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
</tr>
<tr>
<td>2</td>
<td>150.00</td>
</tr>
<tr>
<td>3</td>
<td>65.00</td>
</tr>
</tbody>
</table>

Protocol constraint met? No Yes No Yes  
Deadline constraint met? Yes No Yes Yes  
Message set guaranteed? No No No Yes

Table 6.10: The other example synchronous message set J with $P_i = D_i \geq 2 \cdot TTRT$ only guaranteed by \textit{GLA/IGLA/NLA} ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>27.50</td>
</tr>
<tr>
<td>2</td>
<td>125.00</td>
</tr>
<tr>
<td>3</td>
<td>195.00</td>
</tr>
<tr>
<td>4</td>
<td>140.00</td>
</tr>
</tbody>
</table>

Protocol constraint met? No Yes No Yes  
Deadline constraint met? Yes No Yes Yes  
Message set guaranteed? No No No Yes
Table 6.11: Example synchronous message set K unscheduled by any local SBA scheme ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>28.50</td>
</tr>
<tr>
<td>2</td>
<td>103.50</td>
</tr>
<tr>
<td>3</td>
<td>22.50</td>
</tr>
<tr>
<td>Protocol constraint met?</td>
<td>No</td>
</tr>
<tr>
<td>Deadline constraint met?</td>
<td>Yes</td>
</tr>
<tr>
<td>Message set guaranteed?</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 6.12: Synchronous message set L with $D_i \geq 2 \cdot TTRT$ guaranteed by $LA_M$, $LA_Z$, $GLA$ and $IGLA$ ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>85.00</td>
</tr>
<tr>
<td>2</td>
<td>160.00</td>
</tr>
<tr>
<td>3</td>
<td>188.00</td>
</tr>
<tr>
<td>Protocol constraint met?</td>
<td>Yes</td>
</tr>
<tr>
<td>Deadline constraint met?</td>
<td>Yes</td>
</tr>
<tr>
<td>Message set guaranteed?</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Table 6.13: Synchronous message set M with $D_i \geq 2 \cdot TTRT$ unscheduled by $LA_Z$ ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>$i$</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$D_i$</th>
<th>$LA_M$</th>
<th>$LA_Z$</th>
<th>GLA</th>
<th>IGLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>150.00</td>
<td>5.50</td>
<td>26.50</td>
<td>5.50</td>
<td>5.50</td>
<td>4.50</td>
<td>4.50</td>
</tr>
<tr>
<td>2</td>
<td>92.50</td>
<td>4.80</td>
<td>38.00</td>
<td>2.40</td>
<td>2.40</td>
<td>2.27</td>
<td>2.27</td>
</tr>
<tr>
<td>3</td>
<td>24.00</td>
<td>4.50</td>
<td>185.00</td>
<td>1.99</td>
<td>2.25</td>
<td>1.99</td>
<td>1.88</td>
</tr>
</tbody>
</table>

Protocol constraint met? Yes No Yes Yes
Deadline constraint met? Yes Yes Yes Yes
Message set guaranteed? Yes No Yes Yes

Table 6.14: The other example synchronous message set N with $D_i \geq 2 \cdot TTRT$ unscheduled by $LA_Z$ ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>$i$</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$D_i$</th>
<th>$LA_M$</th>
<th>$LA_Z$</th>
<th>GLA</th>
<th>IGLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>127.00</td>
<td>3.20</td>
<td>100.00</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
</tr>
<tr>
<td>2</td>
<td>66.00</td>
<td>7.00</td>
<td>195.00</td>
<td>1.12</td>
<td>1.17</td>
<td>1.12</td>
<td>1.07</td>
</tr>
<tr>
<td>3</td>
<td>13.50</td>
<td>2.00</td>
<td>132.00</td>
<td>1.61</td>
<td>2.00</td>
<td>1.61</td>
<td>1.49</td>
</tr>
<tr>
<td>4</td>
<td>40.00</td>
<td>3.50</td>
<td>27.50</td>
<td>3.50</td>
<td>3.50</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>5</td>
<td>20.00</td>
<td>6.00</td>
<td>110.00</td>
<td>3.30</td>
<td>3.00</td>
<td>3.30</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Protocol constraint met? Yes No Yes Yes
Deadline constraint met? Yes Yes Yes Yes
Message set guaranteed? Yes No Yes Yes
Table 6.15: Synchronous message set O with $D_i \geq 2 \cdot TTRT$ guaranteed by LAZ and IGLA ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>60.00</td>
</tr>
<tr>
<td>2</td>
<td>95.00</td>
</tr>
<tr>
<td>3</td>
<td>150.00</td>
</tr>
</tbody>
</table>

Protocol constraint met? | No | Yes | No | Yes |
Deadline constraint met? | Yes | Yes | Yes | Yes |
Message set guaranteed? | No | Yes | No | Yes |

Table 6.16: Synchronous message set P with $D_i \geq 2 \cdot TTRT$ guaranteed by GLA and IGLA ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>47.50</td>
</tr>
<tr>
<td>2</td>
<td>50.00</td>
</tr>
<tr>
<td>3</td>
<td>38.50</td>
</tr>
</tbody>
</table>

Protocol constraint met? | No | No | Yes | Yes |
Deadline constraint met? | Yes | Yes | Yes | Yes |
Message set guaranteed? | No | No | Yes | Yes |

114
Table 6.17: Synchronous message set Q with $D_i \geq 2 \cdot TTRT$ only guaranteed by IGLA ($TTTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>i</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$D_i$</th>
<th>$LA_M$</th>
<th>$LA_Z$</th>
<th>GLA</th>
<th>IGLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45.50</td>
<td>9.60</td>
<td>38.00</td>
<td>4.80</td>
<td>4.80</td>
<td>3.87</td>
<td>3.87</td>
</tr>
<tr>
<td>2</td>
<td>37.00</td>
<td>2.50</td>
<td>105.00</td>
<td>0.76</td>
<td>0.84</td>
<td>0.76</td>
<td>0.68</td>
</tr>
<tr>
<td>3</td>
<td>9.50</td>
<td>4.60</td>
<td>95.00</td>
<td>5.45</td>
<td>4.85</td>
<td>5.40</td>
<td>4.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 6.18: Example synchronous message set R with $D_i \geq 2 \cdot TTTRT$ unscheduled by any local SBA scheme ($TTTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>i</th>
<th>$P_i$</th>
<th>$C_i$</th>
<th>$D_i$</th>
<th>$LA_M$</th>
<th>$LA_Z$</th>
<th>GLA</th>
<th>IGLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45.00</td>
<td>5.80</td>
<td>25.00</td>
<td>5.80</td>
<td>5.80</td>
<td>5.40</td>
<td>5.40</td>
</tr>
<tr>
<td>2</td>
<td>38.50</td>
<td>2.00</td>
<td>20.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
<td>2.00</td>
</tr>
<tr>
<td>3</td>
<td>78.00</td>
<td>6.50</td>
<td>90.00</td>
<td>0.94</td>
<td>0.93</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>4</td>
<td>95.00</td>
<td>18.00</td>
<td>140.00</td>
<td>2.05</td>
<td>2.00</td>
<td>2.05</td>
<td>1.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Table 6.19: Synchronous message set $S$ with $D_i > TTRT$ guaranteed by GLA and IGLA ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>60.00</td>
</tr>
<tr>
<td>2</td>
<td>62.00</td>
</tr>
<tr>
<td>3</td>
<td>95.00</td>
</tr>
</tbody>
</table>

| Protocol constraint met? | N/A | N/A | Yes | Yes |
| Deadline constraint met? | N/A | N/A | Yes | Yes |
| Message set guaranteed? | No  | No  | Yes | Yes |

Table 6.20: Example synchronous message set $T$ with $D_i > TTRT$ unscheduled by any local SBA scheme ($TTRT = 10.0$ and $\tau = 0.0$)

<table>
<thead>
<tr>
<th>Message parameters</th>
<th>Synchronous bandwidth $H_i$ allocated by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i$</td>
<td>$P_i$</td>
</tr>
<tr>
<td>1</td>
<td>30.00</td>
</tr>
<tr>
<td>2</td>
<td>45.00</td>
</tr>
<tr>
<td>3</td>
<td>20.00</td>
</tr>
</tbody>
</table>

| Protocol constraint met? | N/A | N/A | Yes | Yes |
| Deadline constraint met? | N/A | N/A | No  | No  |
| Message set guaranteed? | No  | No  | No  | No  |
6.3 Summary

Simulation and numerical examples are used to compare the proposed local SBA schemes with other published local SBA schemes, from the perspective of guaranteeing synchronous message sets and WCAUs, respectively. The results show that the GLA/IGLA scheme can perform better than any previously published local SBA scheme, with the IGLA performing best and achieving the highest performance on guaranteeing synchronous message sets.
Chapter 7

Conclusions

In a distributed system for hard real-time applications, tasks usually reside on different nodes and need to communicate with one another to accomplish a common goal. The communication is achieved through not only the correct but also the timely delivery of inter-task messages (e.g., synchronous messages studied in this research). It is important to select a proper communication network with a proper MAC protocol that can support such delivery. The main focus of this research is to address some important issues related to guaranteeing deadlines of synchronous messages in a timed-token network. A guaranteed message will always be correctly transmitted before its deadline.

This research has concentrated on the timed-token MAC protocol, which is one of the most suitable candidates for distributed hard real-time communications because of its inherent timing property of bounded access time, which provides a necessary condition to guarantee deadlines of synchronous messages. Because of this important property, the timed-token MAC protocol has been incorporated into a number of network standards, including the IEEE 802.4 token bus, the Fiber Distributed Data Interface (FDDI), and the Survivable Adaptive Fiber-Optic Embedded Network (SAFENET), to support hard real-time communications. Also, some work has been undertaken to integrate the timed-token MAC protocol into Ethernet and even wireless networks to enable these networks to support the timely delivery of time-sensitive messages.

To guarantee deadlines of synchronous messages, network parameters, such as the synchronous bandwidth, the $TTRT$, and the buffer size, must be carefully chosen. Among all of these three parameters, the synchronous bandwidth is the most critical. Synchronous messages can be guaranteed by a proper allocation of synchronous bandwidths, which is not offered by the protocol standard itself. This research studied the problem of allocating synchronous bandwidth to guarantee synchronous messages with arbitrary deadline constraints under $D_{\min} > TTRT$. 

118
In addition, the problems of selection of $TTRT$ and allocation of buffer size are also studied. In the areas where the timed-token MAC protocol is adopted, the results obtained in this research can be used to achieve good performance in supporting hard real-time transmission.

## 7.1 Contributions

It is stated in Section 1.2 that the aim of this research is to develop an effective and efficient approach to guarantee the timely transmission of hard real-time traffic in a timed-token ring network. This section examines the aim and objectives listed in Section 1.2 and summarises the main contributions of this research, as follows.

1. **A new Generalised Local Allocation (GLA) SBA scheme**: As detailed in Chapter 4, a new local SBA scheme, named GLA, together with its improved version, named IGLA (Improved GLA), has been developed. Based on the best result of the timing property of the timed-token MAC protocol, the new local SBA scheme developed can be used for synchronous message sets with arbitrary deadline constraints under $D_{min} > TTRT$. That is, the scheme is applicable to any synchronous message set with legitimate deadline constraints. Synchronous message sets which were deemed to be unschedulable because of its $D_{min}$ less than $2 \cdot TTRT$ may now have the chance to be scheduled by using the GLA or IGLA scheme. This new local SBA scheme also improves the performance of the synchronous bandwidth allocation to message sets with $D_{min} \geq 2 \cdot TTRT$ by taking into account the difference between $P_i$ and $D_i$. This results in a few complex solutions for the proposed new scheme, but its performance is better than any of the previously published local SBA schemes.

2. **Schedulability testing**: The two commonly used methods of schedulability testing for a synchronous message set when using the proposed allocation scheme are discussed. In Chapter 4, the schedulability testing by testing the protocol constraint, the deadline constraint and the buffer constraint of the new local SBA scheme is studied. As a result, a condition under which the protocol constraint and the deadline constraint can be satisfied is derived. It is also shown that the buffer constraint can always be met. Chapter 5 studies the schedulability testing and evaluates the Worst Case Achievable Utilisation (WCAU) of the proposed scheme. The derived WCAU result takes into account the number of nodes in the network and is no less than the WCAU of any previously published local SBA scheme.

3. **Optimal selection of $TTRT$**: Guaranteeing synchronous message transmission relies not
only on a proper allocation of synchronous bandwidths but also on an appropriate selection of TTRT. In Chapter 5, an optimal TTRT which can maximise the WCAU under the given settings of other protocol parameters and of message parameters is presented.

7.2 Future Work

There are several directions in which the work presented here can be continued or extended. This section briefly describes some interesting research topics which are worthwhile investigating further.

1. Optimal global SBA scheme for guaranteeing synchronous message sets with arbitrary deadline constraints

The local allocation SBA scheme, although more preferred from a network management perspective due to its flexible and good performance, can never be optimal. This means that a synchronous message set may still be schedulable even if a non-optimal scheme fails to offer a feasible allocation. An optimal SBA scheme must be a global scheme. How to find such an optimal global SBA scheme for guaranteeing synchronous message sets with arbitrary deadline constraints is an open and challenging problem. Most existing work has focused on guaranteeing synchronous message sets with message periods equal to deadlines. Up to now, nobody has proposed an optimal scheme for guaranteeing general message sets with arbitrary deadlines. This problem is directly related to that of schedulability of synchronous message sets with arbitrary message deadline constraints. Currently, how to effectively determine the schedulability of any given synchronous message set is still an unsolved problem. For many general synchronous message sets it is very difficult, if not impossible, to determine their schedulability (since their schedulability may never be determined by any of the methods known so far). The development of such an optimal SBA scheme will provide a solution to this problem.

2. Incorporating the timed-token MAC protocol into Ethernet and wireless local area network (WLAN) to support real-time traffic

The timed-token MAC protocol is one of the most suitable protocols to support real-time communication. Ethernet is the most widely used network in the wired area and WLAN has become more and more popular. However, because of the use of the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) protocol, Ethernet itself cannot support real-time traffic well. The wireless channel is unstable and unpredictable for real-time traffic. Furthermore, the use of the CSMA/CA (Carrier Sense Multiple Access/Collision
Avoidance) protocol, makes the WLAN is even less able to support real-time transmission. Thus, how to integrate the timed-token MAC protocol into Ethernet and WLAN, without changing any hardware structure (in order to offer an efficient way to solve the real-time communication problem in Ethernet and WLAN) is another interesting and challenging problem. One possible way is to build an upper layer over the original MAC layer of the Ethernet and WLAN, using the timed-token MAC protocol, to coordinate the real-time traffic among the different nodes in a network. However, due to the error-prone and unstable nature of wireless channels, there remain some challenging problems to be solved.
References


[38] IEEE. Token-passing bus access method and physical layer specifications. ANSI standard 802.4-1985, 1985.


126


Appendix A

Proof of Theorem 5

**Theorem 5.** If the synchronous bandwidth is allocated according to (4.18) for a synchronous message stream $S_i$ with $P_i < TTRT < D_i < 2 \cdot TTRT$, then the deadline constraint is satisfied under

$$TTRT + \sum_{h=1}^{n} H_h + \tau \leq D_i < 2 \cdot TTRT.$$ 

**Proof.** In order to prove this theorem, we only need to prove that before the deadline of the $k$-th message, there is enough available bandwidth for transmitting these $k$ messages. That is, during the time interval $(k-1) \cdot P_i + D_i$, the available time for transmitting the synchronous messages, $X^k_i$, should be no less than $k \cdot C_i$.

According to Theorem 4 and (4.7), we know that during the time interval $(k-1) \cdot P_i + D_i$, node $i$ can use up $H_i$ at least $\left\lfloor \frac{(k-1) \cdot P_i + D_i - \sum_{h=1}^{n} H_h - \tau}{TTRT} \right\rfloor$ times. So we have:

$$X^k_i \geq \left\lfloor \frac{(k-1) \cdot P_i + D_i - \sum_{h=1}^{n} H_h - \tau}{TTRT} \right\rfloor \cdot H_i$$

$$\geq \left\lfloor \frac{(k-1) \cdot P_i + TTRT}{TTRT} \right\rfloor \cdot H_i \quad \text{(Since } D_i \geq TTRT + \sum_{h=1}^{n} H_h + \tau \text{)}$$

$$= \left( \left\lfloor \frac{(k-1) \cdot P_i}{TTRT} \right\rfloor + 1 \right) \cdot \left( \frac{TTRT}{P_i} + 1 \right) \cdot C_i$$

There are two cases to consider.
Case 1: \((k-1) \cdot P_i < TTRT\)

In this case, we have,

\[
X^k_i \geq \left(\left\lfloor \frac{(k-1) \cdot P_i}{TTRT} \right\rfloor + 1 \right) \cdot \left(\frac{TTRT}{P_i} + 1\right) \cdot C_i \\
= \left(\frac{TTRT}{P_i} + 1\right) \cdot C_i \quad \text{(Since } (k-1) \cdot P_i < TTRT) \\
> \left(\frac{(k-1) \cdot P_i}{P_i} + 1\right) \cdot C_i \quad \text{(Since } (k-1) \cdot P_i < TTRT) \\
= k \cdot C_i
\]

Case 2: \((k-1) \cdot P_i \geq TTRT\)

Under this case, we have

\[
X^k_i \geq \left(\left\lceil \frac{(k-1) \cdot P_i}{TTRT} \right\rceil + 1 \right) \cdot \left(\frac{TTRT}{P_i} + 1\right) \cdot C_i \\
\geq \left(\left\lceil \frac{(k-1) \cdot P_i}{TTRT} \right\rceil \right) \cdot \left(\frac{TTRT}{P_i} + 1\right) \cdot C_i \\
\geq \left(\frac{(k-1) \cdot P_i}{TTRT} \right) \cdot \left(\frac{TTRT}{P_i} + 1\right) \cdot C_i \\
= \left(\frac{(k-1) \cdot P_i}{TTRT} \right) \cdot C_i + \left(\frac{(k-1) \cdot P_i}{TTRT} \right) \cdot C_i \\
= (k-1) \cdot C_i + \left(\frac{(k-1) \cdot P_i}{TTRT} \right) \cdot C_i \\
\geq (k-1) \cdot C_i + 1 \cdot C_i \quad \text{(Since } (k-1) \cdot P_i \geq TTRT) \\
= k \cdot C_i
\]

Thus, the deadline constraint is satisfied. □
Appendix B

Proof of Theorem 6

**Theorem 6.** Let \( f(H_i) = H_i - \left(\frac{TTRT - H_i}{P_i}\right) + 1\) \cdot C_i, then

(a) \( f(H_i) \) is an increasing function of \( H_i \);

(b) If \( \frac{TTRT - TTR + P_i}{P_i + C_i} \in \mathbb{Z} \), then there is a minimum solution \( H_{min} = \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \) for \( f(H_i) \geq 0 \);

(c) If \( \frac{TTRT - TTR + P_i}{P_i + C_i} \notin \mathbb{Z} \), then there is one minimum solution \( H_{min} \) for \( f(H_i) \geq 0 \), subject to \( TTRT + P_i \cdot P_i + C_i \cdot C_i \leq H_{min} < TTRT + 2 \cdot P_i \cdot P_i + C_i \cdot C_i \); where \( f(\frac{TTRT + P_i}{P_i + C_i} \cdot C_i) \leq 0 \) and \( f(\frac{TTRT + 2 \cdot P_i}{P_i + C_i} \cdot C_i) > 0 \).

**Proof.** We prove these three points listed in this theorem one by one as follows:

1. Let \( H_1 > H_2 \), we have

\[
\begin{align*}
f(H_1) - f(H_2) &= [H_1 - \left(\frac{TTRT - H_1}{P_i}\right) + 1]C_i - [H_2 - \left(\frac{TTRT - H_2}{P_i}\right) + 1]C_i \\
&= H_1 - H_2 + \left(\frac{TTRT - H_2}{P_i}\right)C_i - \left(\frac{TTRT - H_1}{P_i}\right)C_i \\
&= (H_1 - H_2) + \left(\frac{TTRT - H_2}{P_i}\right)C_i - \left(\frac{TTRT - H_1}{P_i}\right)C_i
\end{align*}
\]

That is, when \( H_1 > H_2 \), we have \( \frac{TTRT - H_2}{P_i} \geq \frac{TTRT - H_1}{P_i} \). Thus

\[
f(H_1) - f(H_2) > 0
\]

This implies that \( f(H_i) \) is an increasing function of \( H_i \).
2. Let \( H_0 = \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \). Since \( \frac{TTRT - \frac{TTRT + P_i}{P_i + C_i} \cdot C_i}{P_i} \in \mathbb{Z} \), we have

\[
\frac{TTRT - H_0}{P_i} = \frac{TTRT - \frac{TTRT + P_i}{P_i + C_i} \cdot C_i}{P_i} \in \mathbb{Z}.
\]

Thus,

\[
f(H_0) = H_0 - \left( \left\lfloor \frac{TTRT}{P_i} \right\rfloor + 1 \right) C_i
\]

\[
= \frac{TTRT + P_i}{P_i + C_i} \cdot C_i - \left( \left\lfloor \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \right\rfloor + 1 \right) \cdot C_i
\]

\[
= \frac{TTRT + P_i}{P_i + C_i} \cdot C_i - \left( \frac{TTRT + P_i}{P_i + C_i} \cdot C_i + 1 \right) \cdot C_i
\]

\[
= (\frac{TTRT + P_i}{P_i + C_i} - \frac{TTRT + P_i}{P_i + C_i} \cdot C_i - 1) \cdot C_i \tag{B.1}
\]

\[
= P_i(TTTR + P_i) - (P_i + C_i)(TTRT - \frac{TTRT + P_i}{P_i + C_i} \cdot C_i) - (P_i + C_i)P_iC_i
\]

\[
= \frac{P_i(TTTR + P_i)^2 - P_iTTRT - C_iTTTR + C_iTTTR + P_iC_i - P_i^2 - P_iC_iC_i}{(P_i + C_i)P_i}
\]

\[
= 0 \tag{B.2}
\]

This means that \( H_0 = \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \) is a solution of \( f(H_i) \geq 0 \). We now show that \( H_0 \) is the minimum solution of \( f(H_i) \geq 0 \). Assume that \( H_0 \) is not the minimum solution of \( f(H_i) \geq 0 \). That is, there exists \( H' \) (where \( H' < H_0 \)) which is also a solution of \( f(H_i) \geq 0 \), i.e., \( f(H') \geq 0 \). Because \( f(H_i) \) is an increasing function of \( H_i \), we have \( f(H') < f(H_0) = 0 \), this is violation of condition \( f(H') \geq 0 \). Therefore, there is no \( H' \) (where \( H' < H_0 \)) which can satisfy \( f(H_i) \geq 0 \). Thus \( H_0 \) is the minimum solution of \( f(H_i) \geq 0 \).

3. For this case, we have:

\[
f\left( \frac{TTTR + P_i}{P_i + C_i} \cdot C_i \right) = \frac{TTTR + P_i}{P_i + C_i} \cdot C_i - \left( \left\lfloor \frac{TTTR + P_i}{P_i + C_i} \cdot C_i \right\rfloor + 1 \right) \cdot C_i
\]

\[
< \frac{TTTR + P_i}{P_i + C_i} \cdot C_i - \left( \frac{TTTR + P_i}{P_i + C_i} \cdot C_i + 1 \right) \cdot C_i
\]

\[
= \left( \frac{TTTR + P_i}{P_i + C_i} - \frac{TTTR + P_i}{P_i + C_i} \cdot C_i - 1 \right) \cdot C_i
\]

\[
= 0 \quad \text{(Same as the derivations from (B.1) to (B.2))}
\]

\[134\]
Since the ceiling function is not a continuance function, \( f(H_i) \) has points of discontinuity where \( \left\lceil \frac{TTRT - H_i}{P_i + C_i} \right\rceil \in \mathbb{Z} \). In order to show that the minimum accurate solution exists for \( f(H_i) \geq 0 \), we only need to show the following situation is impossible.

Fig. B.1 shows such an impossible situation where there is not an accurate solution for \( f(H') \geq 0 \) because when \( \left\lceil \frac{TTRT - H'}{P_i} \right\rceil \in \mathbb{Z} \), \( f(H') < 0 \). Thus,
\[ f(H') = H' - \left( \frac{TTRT - H'}{P_i} \right) + 1 \cdot C_i \]
\[ = H' - \left( \frac{TTRT - H'}{P_i} + 1 \right) \cdot C_i \]
\[ < 0 \quad \text{(B.3)} \]

(B.3) implies that \( H' < \frac{TTRT + P_i}{P_i + C_i} \cdot C_i = H_1 \) because \( f(H') \) is an increasing function of \( H' \) and \( f(H')|_{H' = \frac{TTRT + P_i}{P_i + C_i} \cdot C_i} = 0 \). This violates Fig. B.1. Thus, Fig. B.1 is impossible. Recall that \( f\left( \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \right) \leq 0 \), \( f\left( \frac{TTRT + 2 \cdot P_i}{P_i + C_i} \cdot C_i \right) > 0 \) and the fact of \( f(H_i) \) being an increasing function of \( H_i \), there must exit the minimum solution of \( f(H_i) \geq 0 \), denoted as \( H_{i_{\min}} \), subject to \( \frac{TTRT + P_i}{P_i + C_i} \cdot C_i \leq H_{i_{\min}} < \frac{TTRT + 2 \cdot P_i}{P_i + C_i} \cdot C_i \).
Appendix C

Deducing $H_i$ from Eq. (4.23)

For convenience, Eq. (4.23) is re-stated as follows:

\[(q_i - 1) \cdot H_i + \max[0, D_i + H_i - (q_i + 1) \cdot TTRT] = C_i \quad \text{(C.1)}\]

We need to derive $H_i$ from (C.1) under the condition $D_i \geq 2 \cdot TTRT$. There are two sub cases to consider here.

**Case 1:** $D_i + H_i - (q_i + 1) \cdot TTRT \leq 0$

Under this condition, the max portion of (C.1) becomes 0, and thus we have

\[(q_i - 1) \cdot H_i = C_i \quad \text{(C.2)}\]

Because $q_i = \lfloor \frac{D_i}{TTRT} \rfloor \geq 2$ (when $D_i \geq 2 \cdot TTRT$), we have $q_i - 1 \geq 1 > 0$. From (C.2), we have

\[H_i = \frac{C_i}{q_i - 1} \quad \text{(C.3)}\]

The condition of this case, $D_i + H_i - (q_i + 1) \cdot TTRT \leq 0$, can now be transformed to an equivalent condition without $H_i$.

\[D_i + (C_i \cdot \frac{1}{q_i - 1} - (q_i + 1) \cdot TTRT \leq 0 \quad \text{(C.4)}\]

**Case 2:** $D_i + H_i - (q_i + 1) \cdot TTRT > 0$

The max portion in (C.1) under this condition is $D_i + H_i - (q_i + 1) \cdot TTRT$ and from (C.1) we have

\[(q_i - 1) \cdot H_i + D_i + H_i - (q_i + 1) \cdot TTRT = C_i \quad \text{(C.5)}\]
Thus
\[ H_i = \frac{C_i + (q_i + 1) \cdot TTRT - D_i}{q_i} \]  
(C.6)

Similar to the transformation in Case 1, (C.6) can be used to get an equivalent condition of 
\[ D_i + H_i - (q_i + 1) \cdot TTRT > 0 \]
as follows:

\[
\begin{align*}
D_i + H_i - (q_i + 1) TTRT & > 0 \\
\iff & \quad \frac{D_i + C_i + (q_i + 1) TTRT - D_i}{q_i} - (q_i + 1) TTRT > 0 \\
\iff & \quad q_i D_i + C_i + (q_i + 1) TTRT - D_i - q_i (q_i + 1) TTRT > 0 \\
\iff & \quad (q_i - 1) D_i - (q_i - 1)(q_i + 1) TTRT + (q_i - 1) \frac{C_i}{q_i - 1} > 0 \\
\iff & \quad \left( \frac{q_i - 1}{q_i} \right) \cdot \left[ D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} \right] > 0 \\
\iff & \quad \left( 1 - \frac{1}{q_i} \right) \cdot \left[ D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} \right] > 0
\end{align*}
\]

Because \( 1 - \frac{1}{q_i} \geq 1 - \frac{1}{2} > 0 \) (since \( q_i \geq 2 \)), we have

\[
\left( 1 - \frac{1}{q_i} \right) \cdot \left[ D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} \right] > 0
\]

\[ \iff D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} > 0 \]  
(C.7)

Based on the above discussion, we have

\[
H_i = \begin{cases} 
\frac{C_i}{q_i - 1} & \text{if } D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} \leq 0 \\
\frac{C_i + (q_i + 1) TTRT - D_i}{q_i} & \text{if } D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} > 0 
\end{cases}
\]

(C.8)

Notice that \( \frac{C_i + (q_i + 1) TTRT - D_i}{q_i} \) can be transformed to \( \frac{C_i}{q_i - 1} - \frac{1}{q_i} \cdot \left[ D_i - (q_i + 1) TTRT + \frac{C_i}{q_i - 1} \right] \),

thus (C.8) can be simplified as the following uniformed equation:

\[
H_i = \frac{C_i}{q_i - 1} - \frac{1}{q_i} \max \{ D_i - [(q_i + 1) TTRT - \frac{C_i}{q_i - 1}], 0 \}
\]
Appendix D

Deducing $H_i$ from Eq. (4.32)

For the convenience of derivation, we re-state Eq. (4.32) as follows:

$$(q_i - 1) \cdot H_i + \max[0, D_i + H_i - (q_i + 1) \cdot TTRT] = \frac{D_i}{P_i} \cdot C_i$$

We want to derive $H_i$ from (D.1) under the condition $D_i \geq 2 \cdot TTRT$. There are two sub cases to consider here.

**Case 1:** $D_i + H_i - (q_i + 1) \cdot TTRT \leq 0$

Under this condition, the max portion of (D.1) becomes 0, and thus we have

$$(q_i - 1) \cdot H_i = \frac{D_i}{P_i} \cdot C_i$$

(D.2)

Because $q_i = \lfloor \frac{D_i}{TTRT} \rfloor \geq 2$ (when $D_i \geq 2 \cdot TTRT$), we have $q_i - 1 \geq 1 > 0$, and thus, from (D.2),

$$H_i = \frac{D_i}{q_i - 1} \cdot C_i$$

(D.3)

$D_i$ can be expressed as $D_i = q_i \cdot TTRT + r_{Di}$, where $0 \leq r_{Di} < TTRT$. Thus

$$H_i = \frac{\frac{q_i TTRT + r_{Di}}{q_i - 1} \cdot C_i}{q_i}$$

(D.4)

Let $H'_i = \frac{\frac{\frac{q_i TTRT + r_{Di}}{q_i - 1} \cdot C_i}{q_i}}{q_i - 1}$, it is obvious that $H'_i \leq H_i$. We have the following analysis:

1. If the synchronous bandwidth is allocated based on the expression $H'_i$ and the deadline constraint can be satisfied, then the bandwidth allocation according to $H_i$ must satisfy the deadline constraint too because during any given time interval there will be more synchronous bandwidth that can be used.
2. When the deadline constraint is satisfied, a smaller value of the synchronous bandwidth is always preferable due to a better chance to satisfy the protocol constraint.

The above analysis suggests using the smaller value $H_i'$ as the synchronous bandwidth. That is

$$H_i = \frac{q_i TTRT \cdot C_i}{q_i - 1} \quad \text{(D.5)}$$

The condition of this case, $D_i + H_i - (q_i + 1) \cdot TTRT \leq 0$, can now be transformed to an equivalent condition without $H_i$.

$$D_i + H_i - (q_i + 1) \cdot TTRT \leq 0$$

$$\Leftrightarrow D_i + \frac{q_i TTRT \cdot C_i}{q_i - 1} - (q_i + 1) \cdot TTRT \leq 0 \quad \text{(D.6)}$$

**Case 2:** $D_i + H_i - (q_i + 1) \cdot TTRT > 0$

The result of the max portion in (D.1) under this condition is $D_i + H_i - (q_i + 1) \cdot TTRT$ and from (D.1) we have

$$(q_i - 1) \cdot H_i + D_i + H_i - (q_i + 1) \cdot TTRT = \frac{D_i C_i}{P_i}$$

Thus

$$H_i = \frac{D_i C_i + (q_i + 1) \cdot TTRT - D_i}{q_i} \quad \text{(D.7)}$$

Let $H_i' = \frac{q_i TTRT \cdot C_i + (q_i + 1) \cdot TTRT - D_i}{q_i}$ and use a similar analysis as the one used in Case 1, we have

$$H_i = \frac{q_i TTRT \cdot C_i + (q_i + 1) \cdot TTRT - D_i}{q_i} \quad \text{(D.8)}$$

Similar to the transformation in case 1, with (D.9), we can derive an equivalent condition of $D_i + H_i - (q_i + 1) \cdot TTRT > 0$, as follows:

$$D_i + H_i - (q_i + 1) TTRT > 0$$

$$\Leftrightarrow D_i + \frac{q_i TTRT \cdot C_i + (q_i + 1) TTRT - D_i}{q_i} - (q_i + 1) TTRT > 0$$

$$\Leftrightarrow q_i D_i + \frac{q_i TTRT \cdot C_i + (q_i + 1) TTRT - D_i - q_i (q_i + 1) TTRT}{q_i} > 0$$

$$\Leftrightarrow (q_i - 1) D_i - (q_i - 1) (q_i + 1) TTRT + (q_i - 1) \frac{q_i TTRT \cdot C_i}{q_i - 1} > 0$$

$$\Leftrightarrow (q_i - 1) \cdot \frac{D_i - (q_i + 1) TTRT + \frac{q_i TTRT \cdot C_i}{q_i - 1}}{q_i} > 0$$

$$\Leftrightarrow (1 - \frac{1}{q_i}) \cdot \frac{D_i - (q_i + 1) TTRT + \frac{q_i TTRT \cdot C_i}{q_i - 1}}{q_i} > 0$$
Since \(1 - \frac{1}{q_i} \geq 1 - \frac{1}{2} > 0\) due to \(q_i \geq 2\)

\[
(1 - \frac{1}{q_i}) \cdot [D_i - (q_i + 1)TTRT + \frac{q_i^{TTRT} C_i}{q_i - 1}] > 0
\]

\[
\Leftrightarrow D_i - (q_i + 1)TTRT + \frac{q_i^{TTRT} C_i}{q_i - 1} > 0 \quad \text{(D.10)}
\]

Based on the above discussion, we have

\[
H_i = \begin{cases} 
\frac{q_i^{TTRT} C_i}{q_i - 1} & \text{if } D_i - (q_i + 1)TTRT + \frac{q_i^{TTRT} C_i}{q_i - 1} \leq 0 \\
\frac{q_i^{TTRT} C_i + (q_i + 1)TTRT - D_i}{q_i} & \text{if } D_i - (q_i + 1)TTRT + \frac{q_i^{TTRT} C_i}{q_i - 1} > 0 
\end{cases} \quad \text{(D.11)}
\]

Notice that \(\frac{q_i^{TTRT} C_i + (q_i + 1)TTRT - D_i}{q_i}\) can be transformed to \(\frac{q_i^{TTRT} C_i}{q_i - 1} - \frac{1}{q_i} \cdot [D_i - (q_i + 1)TTRT + \frac{q_i^{TTRT} C_i}{q_i - 1}]\), thus (D.11) can be simplified as the following uniformed equation:

\[
H_i = \frac{q_i^{TTRT} C_i}{q_i - 1} - \frac{1}{q_i} \max\{D_i - [(q_i + 1)TTRT - \frac{q_i^{TTRT} C_i}{q_i - 1}], 0\}
\]
Appendix E

Proof of Theorem 8

**Theorem 8.** Assuming that at time $t_0$, a synchronous message with period $P_i$ and deadline $D_i$ arrives at node $i$. Then, under protocol constraints, in time interval $[t_0, t_0 + (k-1) \cdot P_i + D_i]$, the minimum amount of available synchronous transmitting time $X^k_i(\vec{H})$, based on the global information of the network, is given by

$$X^k_i(\vec{H}) = (m_i - 1) \cdot H_i + \max[0, (k-1) \cdot P_i + D_i] - \{m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - \lfloor \frac{m_i}{n+1} \rfloor \cdot [TTRT - \sum_{j=1}^{n} H_j + \tau] - H_i \}$$ (E.1)

where $m_i$ is an integer ($m_i \geq 2$) which makes the inequality of $I(m_i-1) \leq (k-1) \cdot P_i + D_i < I(m_i)$ hold, and must be either

$$m_i = \left\lfloor \frac{[(k-1) \cdot P_i + D_i] \cdot (n+1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor$$

or

$$m_i = \left\lfloor \frac{[(k-1) \cdot P_i + D_i] \cdot (n+1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \right\rfloor - 1$$

**Proof.** To prove this theorem, we first show there must exist one and only one integer $m_i$ which makes $I(m_i-1) \leq (k-1) \cdot P_i + D_i < I(m_i)$ hold.
From (3.15), we have

\[ I(m_i) - I(m_i - 1) \]

\[ = m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - \left\lfloor \frac{m_i}{n+1} \right\rfloor \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau) \]

\[ - \left\lfloor \frac{m_i - 1}{n+1} \right\rfloor \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau) \]

\[ = TTRT + (TTRT - \sum_{j=1}^{n} H_j - \tau) \left( \left\lfloor \frac{m_i - 1}{n+1} \right\rfloor - \left\lfloor \frac{m_i}{n+1} \right\rfloor \right) \]

\[ > (TTRT - \sum_{j=1}^{n} H_j - \tau) + (TTRT - \sum_{j=1}^{n} H_j - \tau) \left( \left\lfloor \frac{m_i - 1}{n+1} \right\rfloor - \left\lfloor \frac{m_i}{n+1} \right\rfloor \right) \]

\[ = (TTRT - \sum_{j=1}^{n} H_j - \tau)(1 + \left\lfloor \frac{m_i - 1}{n+1} \right\rfloor - \left\lfloor \frac{m_i}{n+1} \right\rfloor) \]

\[ = (TTRT - \sum_{j=1}^{n} H_j - \tau)(\left\lfloor \frac{m_i + n}{n+1} \right\rfloor - \left\lfloor \frac{m_i}{n+1} \right\rfloor) \]

\[ \geq 0 \]

thus to any \( m_i \), there is \( I(m_i) > I(m_i - 1) \). This implies that \( I(m_i) \) is an increasing function of \( m_i \).

Comparing the value of \( I(1) = TTRT + \sum_{j=1}^{n} H_j + \tau < 2 \cdot TTRT \) and the minimum value of \((k - 1) \cdot P_i + D_i\), i.e., \((1 - 1) \cdot P_i + D_i = D_i\), we have

- If \( D_i < 2 \cdot TTRT \), according to Corollary 1, we have \( D_i \geq I(1) \). Otherwise, in the worst case, node \( i \) cannot use its allocated synchronous bandwidth even once.

- If \( D_i \geq 2 \cdot TTRT \), we have \( I(1) < 2 \cdot TTRT < D_i \).

Thus, the value of \( I(1) \) must be no larger than the minimum value of \((k - 1) \cdot P_i + D_i\).

The above analysis (referring to Fig. E.1) ensures that there must exist one and only one integer \( m_i \) which makes \( I(m_i - 1) \leq (k - 1) \cdot P_i + D_i < I(m_i) \) hold.
Figure E.1: The possibility of existence of $m_i$

Now we can find out such $m_i$. Because

$$I(m_i - 1) \leq (k - 1) \cdot P_i + D_i$$

$$\Rightarrow (m_i - 1) \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - \left\lfloor \frac{m_i - 1}{n + 1} \right\rfloor \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau) \leq (k - 1) \cdot P_i + D_i$$

(from (3.15))

$$\Rightarrow \frac{(m_i - 1) \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - (k - 1)P_i - D_i}{TTRT - \sum_{j=1}^{n} H_j - \tau} \leq \left\lfloor \frac{m_i - 1}{n + 1} \right\rfloor \leq \frac{m_i - 1}{n + 1}$$

$$\Rightarrow m_i \leq \frac{(n + 1)((k - 1) \cdot P_i + D_i) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}$$

(Since $m_i$ is integer)
Similarly, from
\[(k - 1) \cdot P_i + D_i < I(m_i)\]
\[\Rightarrow (k - 1) \cdot P_i + D_i < m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - \left\lfloor \frac{m_i}{n + 1} \right\rfloor \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)\]
\[\Rightarrow \frac{m_i}{n + 1} - \frac{n}{n + 1} \leq \left\lfloor \frac{m_i}{n + 1} \right\rfloor < \frac{m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - (k - 1) \cdot P_i - D_i}{TTRT - \sum_{j=1}^{n} H_j - \tau}\]
\[\Rightarrow \frac{m_i - n}{n + 1} < \frac{m_i \cdot TTRT + \sum_{j=1}^{n} H_j + \tau - (k - 1) \cdot P_i - D_i}{TTRT - \sum_{j=1}^{n} H_j - \tau}\]
\[\Rightarrow \frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] - \sum_{j=1}^{n} H_j - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \leq m_i \leq \frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\]  \hspace{1cm} (Since \(m_i\) is integer)

Thus, if there exists \(m_i\) which makes \((k - 1) \cdot P_i + D_i < I(m_i)\) hold, \(m_i\) must satisfy
\[\frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \leq m_i \leq \frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\]  \hspace{1cm} (E.2)

Further more, because
\[\frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} - \frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] - \sum_{j=1}^{n} H_j - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} \leq \frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} - \frac{(n + 1) \cdot [(k - 1) \cdot P_i + D_i] - \sum_{j=1}^{n} H_j - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} = \frac{n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau) + \sum_{j=1}^{n} H_j + \tau + n \cdot TTRT}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} = 1 + \frac{n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau} < 1 + 1 \quad (Since \ n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau) < n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau) = 2\]
and
\[
\begin{align*}
&\left[\frac{(n+1)((k-1) \cdot P_i + D_i) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\right] \\
&- \left[\frac{(n+1) \cdot [(k-1) \cdot P_i + D_i] - \sum_{j=1}^{n} H_j - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\right] \\
&\geq -1 \quad \text{Since } TTRT - \sum_{j=1}^{n} H_j - \tau \geq 0
\end{align*}
\]

we have
\[
0 \leq \left[\frac{(n+1)((k-1) \cdot P_i + D_i) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\right] - \left[\frac{(n+1) \cdot [(k-1) \cdot P_i + D_i] - \sum_{j=1}^{n} H_j - \tau - n \cdot TTRT}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\right] \leq 1 \quad \text{(E.3)}
\]

The expressions of (E.2), (E.3) and the fact that there is one and only one \( m_i \) imply that the possible value of \( m_i \) must be either
\[
m_i = \left[\frac{[(k-1) \cdot P_i + D_i] \cdot (n+1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\right]
\]
or
\[
m_i = \left[\frac{[(k-1) \cdot P_i + D_i] \cdot (n+1) + n \cdot (TTRT - \sum_{j=1}^{n} H_j - \tau)}{n \cdot TTRT + \sum_{j=1}^{n} H_j + \tau}\right] - 1
\]

The remaining task is to find out the minimum amount of available synchronous transmitting time during the time interval of \((k-1) \cdot P_i + D_i\).

By Theorem 4, we know that during the first \( I(m_i - 1) \) time interval of \((k-1) \cdot P_i + D_i\), i.e., in the time interval of \((t_0, t_0 + I(m_i - 1))\), node \( i \) can use its allocated synchronous bandwidth \( H_i \) at least \((m_i - 1)\) times. So we have,
\[
X_i^k(H_i) \geq (m_i - 1) \cdot H_i.
\]

In the worst case, node \( i \) can get the chance of using part of its allocated bandwidth \( H_i \) during the remaining time interval, i.e., \((t_0 + I(m_i - 1), t_0 + (k-1) \cdot P_i + D_i)\), if any, only when the following inequality holds:
\[
I_{m_i} - H_i < (k-1) \cdot P_i + D_i < I(m_i).
\]
The minimum amount of time available for node $i$ to do synchronous transmission during the remaining period can then be obtained by the calculation of $\max[(k-1)\cdot P_i + D_i - (I(m_i) - H_i), 0]$, in particular:

$$\max\{(k - 1) \cdot P_i + D_i - [m_i \cdot TT\tau T + \sum_{j=1}^{n} H_j + \tau - \lfloor \frac{m_i}{n+1} \rfloor \cdot (TT\tau T - \sum_{j=1}^{n} H_j - \tau) - H_i], 0\}. $$

Thus, the total minimum available time $X_k^i(\vec{H})$ for node $i$ to send its synchronous messages during $(k - 1) \cdot P_i + D_i$ in the worst case is the sum of the above two parts, i.e., Eq. E.1.

Theorem 8 follows from the above discussion. \qed
Appendix F

Proof of Theorem 9

Theorem 9. Under the protocol constraint, the deadline constraint can be satisfied for any stream $S_i$ with $D_i \geq 2 \cdot TTRT$ and $D_i > P_i$ when the synchronous bandwidth is allocated to $S_i$ using the allocation defined in (4.35) except when $q_i = \lceil \frac{P_i}{TTRT} \rceil = \lceil \frac{P}{TTRT} \rceil + 1$.

The following lemmas are required for the proof of Theorem 9. The proofs of these lemmas are listed later.

Lemma 5. For a synchronous message set, if the bandwidth allocated to a message stream $S_i$ with $D_i \geq 2 \cdot TTRT$ can meet the following condition, for all the integers $k \geq 1$, under the protocol constraint

$$x^k_i = \left\lceil \frac{(k-1)P_i + D_i}{TTRT} \right\rceil - 1 \cdot H_i + \max\{(k-1)P_i + D_i + H_i - \left\lceil \frac{(k-1)P_i + D_i}{TTRT} \right\rceil + 1 \cdot TTRT, 0\} \geq k \cdot C_i$$

then the deadline requirements on that stream can be guaranteed.

Lemma 6. If the synchronous bandwidth defined in (4.35) is allocated to any stream $S_i$ with $D_i \geq 2 \cdot TTRT$, $D_i > P_i$ and $D_i - \left\lceil (q_i + 1)TTRT - \frac{\max(q_i TTRT)}{q_i - 1} C_i \right\rceil \leq 0$, then all the messages in that stream can be sent before their deadlines.

Lemma 7. If the synchronous bandwidth defined in (4.35) is allocated to any stream $S_i$ with $D_i \geq 2 \cdot TTRT$, $D_i \geq P_i + 2 \cdot TTRT$ and $D_i - \left\lceil (q_i + 1)TTRT - \frac{\max(q_i TTRT)}{q_i - 1} C_i \right\rceil > 0$, then all the messages in that stream can be sent before their deadlines.
Lemma 8. If the synchronous bandwidth defined in (4.35) is allocated to any stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \), \( \lfloor \frac{P_i}{TTRT} \rfloor = q_i \) and \( D_i - [(q_i + 1)TTRT - \frac{\max(q_iTTRT, 1)C_i}{q_i - 1}] > 0 \), then all the messages in that stream can be sent before their deadlines.

Lemma 9. If the synchronous bandwidth defined in (4.35) is allocated to any stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \), \( \lfloor \frac{P_i}{TTRT} \rfloor + 2 = \lfloor \frac{D_i}{TTRT} \rfloor = q_i \) and \( D_i - [(q_i + 1)TTRT - \frac{\max(q_iTTRT, 1)C_i}{q_i - 1}] > 0 \), then all the messages in that stream can be sent before their deadlines.

Lemma 10. If the synchronous bandwidth defined in (4.35) is allocated to any stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \), \( \lfloor \frac{P_i}{TTRT} \rfloor + 1 = \lfloor \frac{D_i}{TTRT} \rfloor = q_i \) and \( D_i - [(q_i + 1)TTRT - \frac{\max(q_iTTRT, 1)C_i}{q_i - 1}] > 0 \), then NOT all the messages in that stream can be sent before their deadlines.

Proof of Lemma 5

Lemma 5. For a synchronous message set, if the bandwidth allocated to a message stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \) can meet the following condition, for all the integers \( k \geq 1 \), under the protocol constraint

\[
x_i^k = \left[ \frac{(k - 1)P_i + D_i}{TTRT} \right] - 1 \cdot H_i + \max\{(k - 1)P_i + D_i + H_i - \left[ \frac{(k - 1)P_i + D_i}{TTRT} \right] + 1 \cdot TTRT, 0\}
\]

\[
\geq k \cdot C_i
\]

then the deadline requirements on that stream can be guaranteed.

Proof. This lemma follows from the discussion in Section 4.2.2. Condition (F.2) is a sufficient condition when used in testing deadline constraints.

Proof of Lemma 6

Lemma 6. If the synchronous bandwidth defined in (4.35) is allocated to any stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \), \( D_i > P_i \) and \( D_i - [(q_i + 1)TTRT - \frac{\max(q_iTTRT, 1)C_i}{q_i - 1}] \leq 0 \), then all the messages in that stream can be sent before their deadlines.

Proof. When \( D_i - [(q_i + 1)TTRT - \frac{\max(q_iTTRT, 1)C_i}{q_i - 1}] \leq 0 \), (4.35) can be written as:

\[
H_i = \frac{\max(q_iTTRT, 1)C_i}{q_i - 1}
\]

(F.3)
Based on Lemma 5, in order to prove the deadline constraint, we only need to establish (F.2). Since

\[ x_k^i = \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] - 1 \cdot H_i + \max\{((k-1)P_i + D_i + H_i - \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] + 1) \cdot TTRT, 0 \} \]

\[ \geq \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] - 1 \cdot H_i, \]

we only need to show

\[ \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] - 1 \cdot H_i \geq k \cdot C_i \quad (F.4) \]

The process of proof of this lemma is part of the proof in [55]. However, for a self-contained proof, we write up the proof here with slight changes and re-organization. Three are three cases to consider here:

**Case 1: When** \( P_i < D_i < P_i + TTRT \)

Since \( P_i < D_i < P_i + TTRT \), we have \( \frac{P_i}{TTRT} < \frac{D_i}{TTRT} < \frac{P_i}{TTRT} + 1 \), hence

\[ \left\lfloor \frac{P_i}{TTRT} \right\rfloor \leq \left\lfloor \frac{D_i}{TTRT} \right\rfloor \leq \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1 \]

thus, we have two sub-cases to consider under this case: when \( \left\lfloor \frac{P_i}{TTRT} \right\rfloor = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i \) and when \( \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1 = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i \)

**Sub-case 1:** \( \left\lfloor \frac{P_i}{TTRT} \right\rfloor = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i \)

In this case, we have

\[ P_i = q_i \cdot TTRT + r_{P_i} \quad (F.5) \]

\[ D_i = q_i \cdot TTRT + r_{D_i} \quad (F.6) \]

Since \( \frac{q_i \cdot TTRT - r_{P_i}}{P_i} \leq 1 \), from (F.3), we have

\[ H_i = \frac{C_i}{q_i - 1} \quad (F.7) \]
Thus

\[
\left\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \right\rfloor - 1 \cdot H_i = \left\lfloor \frac{(k - 1)(q_iTTRT + r_P) + q_iTTRT + r_D}{TTRT} \right\rfloor - 1 \cdot H_i
\]

\[
= \left\lfloor \frac{(k - 1)q_iTTRT + (k - 1)r_P + q_iTTRT + r_D}{TTRT} \right\rfloor - 1 \cdot H_i
\]

\[
= \left\lfloor \frac{k \cdot q_i \cdot TTRT + (k - 1)r_P + r_D}{TTRT} \right\rfloor - 1 \cdot H_i
\]

\[
\geq (kq_i - 1) \cdot H_i
\]

\[
\geq (kq_i - k) \cdot H_i \quad \text{(Since 1 \leq k)}
\]

\[
= k(q_i - 1) \cdot H_i
\]

\[
\geq k(q_i - 1) \cdot \frac{C_i}{q_i - 1} \quad \text{(Since (F.7))}
\]

\[
= k \cdot C_i
\]

Therefore (F.4) is established.

**Sub-case 2:** \( \left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1 = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i \)

Under this case, we have

\[
P_i = (q_i - 1) \cdot TTRT + r_P \quad \text{(F.8)}
\]

\[
D_i = q_i \cdot TTRT + r_D \quad \text{(F.9)}
\]

Since

\[
\frac{q_iTTRT}{P_i} = \frac{q_iTTRT}{(q_i - 1)TTRT + r_P}
\]

\[
= \frac{q_iTTRT}{q_iTTRT - (TTRT - r_P)}
\]

\[
\geq 1 \quad \text{(F.10)}
\]

from (F.3), we have

\[
H_i = \frac{q_iTTRT \cdot C_i}{P_i} \quad \text{(F.11)}
\]
Thus
\[ \left\lfloor \frac{(k-1)P_i + D_i}{TT\text{RT}} \right\rfloor - 1 \cdot H_i \]
\[ = \left\lfloor \frac{(k-1)(q_i-1)TT\text{RT} + r_{P_i}}{TT\text{RT}} + q_i TT\text{RT} + r_{D_i}} - 1 \right\rfloor \cdot H_i \]
\[ = \left\lfloor \frac{(k-1)(q_i-1)TT\text{RT} + q_i TT\text{RT} + (k-1)r_{P_i} + r_{D_i}}{TT\text{RT}} - 1 \right\rfloor \cdot H_i \]
\[ = \left\lfloor (k-1)(q_i-1) + q_i - 1 + \left( \frac{(k-1)r_{P_i} + r_{D_i}}{TT\text{RT}} \right) \right\rfloor \cdot H_i \]
\[ = [k \cdot (q_i - 1)] + \left( \frac{q_i TT\text{RT} C_i}{q_i - 1} \right) \cdot H_i \]
\[ \geq [k \cdot (q_i - 1)] \cdot H_i \]
\[ = [k \cdot (q_i - 1)] \cdot \frac{q_i TT\text{RT} C_i}{q_i - 1} \quad \text{(Since (F.11))} \]
\[ = k \cdot \frac{q_i TT\text{RT} C_i}{q_i - 1} \]
\[ \geq k \cdot C_i \quad \text{(Since (F.10))} \]

Therefore (F.4) is established.

Case 2: When \( P_i + TT\text{RT} \leq D_i < P_i + 2 \cdot TT\text{RT} \)

Since \( P_i + TT\text{RT} \leq D_i < P_i + 2 \cdot TT\text{RT} \), we have \( \frac{q_i TT\text{RT} C_i}{P_i} > \frac{D_i - TT\text{RT}}{P_i} > P_i \) Thus
\[
H_i = \frac{q_i TT\text{RT} C_i}{q_i - 1} \quad \text{(F.12)}
\]

When \( k = 1 \), we have \( \left\lfloor \frac{(k-1)P_i + D_i}{TT\text{RT}} \right\rfloor - 1 \cdot H_i = (q_i - 1) \cdot \frac{q_i TT\text{RT} C_i}{q_i - 1} = q_i TT\text{RT} C_i > C_i = k \cdot C_i \), therefore (F.4) is established when \( k = 1 \).

Now we only need to consider \( k \geq 2 \). There are two sub-cases which need to be considered.
Sub-case 1: $P_i < TTRT$

Let $D_i = q_i TTRT + r_{D_i}$, we have

$$\left\lfloor \left( \frac{(k-1)P_i}{TTRT} + D_i \right) - 1 \right\rfloor \cdot H_i$$

$$\geq \left\lfloor \left( \frac{(k-1)P_i}{TTRT} + q_i TTRT \right) - 1 \right\rfloor \cdot H_i$$

$$= \left\lfloor \left( \frac{(k-1)P_i}{TTRT} + q_i - 1 \right) \right\rfloor \cdot H_i$$

$$= \left\lfloor \left( \frac{(k-1)P_i}{TTRT} \right) + q_i - 1 \right\rfloor \cdot H_i$$

$$\geq \left\lfloor \left( \frac{(k-1)P_i}{TTRT} \right) + q_i - 2 \right\rfloor \cdot H_i$$

$$= \left\lfloor \left( \frac{(k-1)P_i}{TTRT} \right) + q_i - 2 \right\rfloor \cdot \frac{q_i TTRT C_i}{q_i - 1} \quad \text{(Since (F.12))}$$

$$= \left\lfloor \left( \frac{(k-1)P_i}{TTRT} \right) + q_i - 2 \right\rfloor \cdot \frac{q_i C_i}{(q_i - 1)P_i} \cdot TTRT$$

$$\geq \left\lfloor \left( \frac{(k-1)P_i}{TTRT} \right) + q_i - 2 \right\rfloor \cdot \frac{q_i C_i}{(q_i - 1)P_i} \cdot \frac{(k-1)P_i}{TTRT}$$

(Since $\left\lfloor \left( \frac{(k-1)P_i}{TTRT} \right) \right\rfloor \geq \frac{(k-1)P_i}{TTRT} \Rightarrow TTRT \geq \frac{(k-1)P_i}{TTRT}$ when $k \geq 2$)

$$= \frac{q_i}{q_i - 1} (k - 1)C_i + \left( \frac{q_i - 2}{q_i - 1} \right) \cdot \frac{k - 1}{TTRT} C_i$$

(F.13)

Recall $D_i \geq 2TTRT$, we only need to consider $q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor \geq 2$.

- If $q_i = 2$, from (F.13), we have

  $$\left\lfloor \left( \frac{(k-1)P_i}{TTRT} + D_i \right) - 1 \right\rfloor \cdot H_i \geq \frac{q_i}{q_i - 1} (k - 1)C_i + \left( \frac{q_i - 2}{q_i - 1} \right) \cdot \frac{k - 1}{TTRT} C_i$$

  $$= 2(k - 1)C_i$$

  $$\geq kC_i \quad \text{(Since $k \geq 2$)}$$

- If $q_i > 2$, from (F.13), we have

  $$\left\lfloor \left( \frac{(k-1)P_i}{TTRT} + D_i \right) - 1 \right\rfloor \cdot H_i$$

  $$\geq \frac{q_i}{q_i - 1} (k - 1)C_i + \left( \frac{q_i - 2}{q_i - 1} \right) \cdot \frac{k - 1}{TTRT} C_i$$

  $$> (k - 1)C_i + \left( \frac{q_i - 2}{q_i - 1} \right) \cdot \frac{k - 1}{TTRT} C_i \quad \text{(Since $\frac{q_i}{q_i - 1} > 1$)}$$

  $$> (k - 1)C_i + C_i$$

  (Since $\frac{q_i - 2}{q_i - 1} > 1$ and $\left\lfloor \frac{(k-1)P_i}{TTRT} \right\rfloor \leq [k - 1] = k - 1$)

  $$= kC_i \quad \text{(Since $k \geq 2$)}$$

153
Thus, (F.4) is established.

Sub-case 2: $P_i \geq TT\cdot RT$

In this case,

$$\left\lfloor \frac{(k-1)P_i + D_i}{TT\cdot RT} \right\rfloor \cdot H_i \geq \left\lfloor \frac{(k-1)P_i + P_i + TT\cdot RT}{TT\cdot RT} \right\rfloor \cdot H_i$$

$$= \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor \cdot H_i \quad \text{(Since } D_i \geq P_i + TT\cdot RT)$$

$$\geq \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor - 1 \cdot H_i$$

$$= \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor - 1 \cdot \frac{q_i TT\cdot RT}{(q_i - 1)P_i} \quad \text{(By (F.12))}$$

$$\geq \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor - 1 \cdot \frac{q_i C_i}{(q_i - 1)P_i} \cdot TT\cdot RT$$

(Since $\frac{kP_i}{TT\cdot RT} \leq \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor \Leftrightarrow TT\cdot RT \geq \frac{kP_i}{TT\cdot RT}$)

$$= \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor - 1 \cdot \frac{q_i C_i}{(q_i - 1)\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor} \cdot kC_i$$

in order to show (F.4), we only need to show

$$\frac{\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor - 1}{(q_i - 1)\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor} q_i \geq 1$$

\hspace{1cm} (F.14)

Since

$$\frac{\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor - 1}{(q_i - 1)\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor} q_i = \frac{\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor q_i - q_i}{q_i - \left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor} = \frac{q_i - q_i/\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor}{q_i - 1}$$

we only need to show

$$\frac{q_i - q_i/\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor}{q_i - 1} \geq 1$$

\hspace{1cm} (F.15)

This implies that we only need to establish

$$\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor \geq q_i$$

\hspace{1cm} (F.16)

Recall that we also have $k \geq 2$ and $P_i \geq TT\cdot RT$ here.

The following derivations show how (F.16) can be established.

- If $kP_i \geq D_i$, we have

$$\left\lfloor \frac{kP_i}{TT\cdot RT} \right\rfloor \geq \left\lfloor \frac{D_i}{TT\cdot RT} \right\rfloor \geq \left\lfloor \frac{D_i}{TT\cdot RT} \right\rfloor = q_i$$

So (F.16) holds for this case.
If \( kP_i < D_i \), we have

\[
kP_i < D_i < P_i + 2 \cdot TTRT \leq P_i + 2 \cdot P_i = 3 \cdot P_i
\]

thus \( k < 3 \). Since \( k \geq 2 \), we must have \( k = 2 \).

1. When \( P_i = TTRT \), there is

\[
\left\lfloor \frac{kP_i}{TTRT} \right\rfloor = \left\lfloor \frac{2P_i}{TTRT} \right\rfloor = 2
\]

On the other hand, because \( 2TTRT \leq D_i < P_i + 2TTRT = 3TTRT \), we have

\[
q_i = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = 2
\]

thus \( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor = q_i \).

2. When \( P_i > TTRT \), we have

\[
\left\lfloor \frac{kP_i}{TTRT} \right\rfloor = \left\lfloor \frac{2P_i}{TTRT} \right\rfloor \geq 3 \quad \text{(F.17)}
\]

By \( D_i < P_i + 2TTRT \) we also have

\[
TTRT > \frac{D_i - P_i}{2} \quad \text{(F.18)}
\]

Recall that we consider under \( D_i > kP_i = 2P_i \), from (F.18), we have

\[
TTRT > \frac{P_i}{2} \quad \text{(F.19)}
\]

From (F.17), in order to show (F.16), we only need to show \( q_i \leq 3 \). Now we give out that \( q_i \geq 4 \) is impossible. If \( q_i \geq 4 \), there is

\[
D_i = q_i TTRT + r_{D_i}
\]

\[
\geq 4 \cdot TTRT
\]

\[
= 2 \cdot TTRT + 2 \cdot TTRT
\]

\[
> P_i + 2 \cdot TTRT
\]

which is a contradiction because \( D_i < P_i + 2 \cdot TTRT \). Thus \( q_i \leq 3 \) and (F.16) is then established.

The above procedure shows the proof of (F.16) which, in turn, leads to the establishment of (F.4).
Case 3: When \( P_i + 2 \cdot TTTRT \leq D_i \)

In this case, we have \( q_i \cdot TTTRT > D_i - TTTRT > D_i - 2 \cdot TTTRT > P_i \), from (F.3), there is

\[
H_i = \frac{q_i \cdot TTTRT}{q_i - 1} \quad \text{(F.20)}
\]

Thus,

\[
\left\lfloor \left( \frac{k - 1}{P_i + D_i} \right) \right\rfloor H_i \geq \left\lfloor \left( \frac{k - 1}{P_i + 2 \cdot TTTRT} \right) \right\rfloor H_i
\]

\[
= \left\lfloor \left( \frac{k \cdot P_i + TTTRT}{TTTRT} \right) \right\rfloor H_i
\]

\[
= \left\lfloor \left( \frac{k \cdot P_i}{TTTRT} \right) \right\rfloor + 1 \cdot H_i
\]

\[
\geq \left\lfloor \frac{k \cdot P_i}{TTTRT} \right\rfloor \cdot H_i
\]

\[
\geq \frac{k \cdot P_i}{TTTRT} \cdot H_i
\]

\[
\geq \frac{k \cdot P_i}{TTTRT} \cdot \frac{q_i \cdot TTTRT \cdot C_i}{q_i - 1}
\]

\[
= \frac{q_i}{q_i - 1} \cdot kC_i
\]

\[
> k \cdot C_i
\]

Therefore (F.4) is established.

The lemma then follows from the above three cases. \(\Box\)

**Proof of Lemma 7**

**Lemma 7.** *If the synchronous bandwidth defined in (4.35) is allocated to any stream \( S_i \) with \( D_i \geq 2 \cdot TTTRT, D_i \geq P_i + 2 \cdot TTTRT \) and \( D_i - [(q_i + 1)TTTRT - \frac{\max(q_i \cdot TTTRT, P_i)}{q_i - 1}C_i] > 0 \), then all the messages in that stream can be sent before their deadlines.*

**Proof.** When \( D_i - [(q_i + 1)TTTRT - \frac{\max(q_i \cdot TTTRT, P_i)}{q_i - 1}C_i] > 0 \), (4.35) can be written as:

\[
H_i = \frac{\max(q_i \cdot TTTRT, P_i) \cdot C_i + (q_i + 1)TTTRT - D_i}{q_i} \quad \text{(F.21)}
\]

Since

\[
\frac{q_i \cdot TTTRT}{P_i} = \frac{D_i - r_{D_i}}{P_i}
\]

\[
> \frac{P_i + 2 \cdot TTTRT - r_{D_i}}{P_i} \quad \text{(Since \( D_i \geq P_i + 2 \cdot TTTRT \))}
\]

\[
= 1 + \frac{2 \cdot TTTRT - r_{D_i}}{P_i}
\]

\[
> 1 \quad \text{(Since \( TTTRT > r_{D_i} \))}
\]

156
Thus
\[
\max \left( \frac{q_i TTRT}{P_i}, 1 \right) = \frac{q_i TTRT}{P_i} \tag{F.22}
\]

From (F.21) and (F.22), we have
\[
H_i = \frac{q_i TTRT}{P_i} C_i + \frac{(q_i + 1) TTRT - D_i}{q_i} \tag{F.23}
\]

Based on Lemma 5, in order to prove the deadline constraint, we only need to establish (F.2).

Since
\[
x^k_i = \left\{ \left\lfloor \frac{(k-1)P_i + D_i}{TTRT} \right\rfloor - 1 \right\} \cdot H_i
\]
\[
+ \max \{ (k-1)P_i + D_i + H_i - \left\lfloor \frac{(k-1)P_i + D_i}{TTRT} \right\rfloor + 1 \} \cdot TTRT, 0 \}
\]
\[
\geq \left\lfloor \frac{(k-1)P_i + D_i}{TTRT} \right\rfloor - 1 \cdot H_i
\]
\[
\geq \left\lfloor \frac{(k-1)P_i + P_i + 2 \cdot TTRT}{TTRT} \right\rfloor - 1 \cdot H_i \quad \text{(Since } D_i \geq P_i + 2 \cdot TTRT \text{)}
\]
\[
= \left\lfloor \frac{k \cdot P_i}{TTRT} \right\rfloor + 1 \cdot H_i
\]
\[
> k \cdot \frac{P_i}{TTRT} \cdot H_i
\]
\[
= \frac{k \cdot P_i}{TTRT} \cdot \frac{q_i TTRT}{P_i} C_i + \frac{(q_i + 1) TTRT - D_i}{q_i}
\]
\[
> \frac{k \cdot P_i}{TTRT} \cdot \frac{q_i TTRT}{P_i} C_i \quad \text{(Since } (q_i + 1) TTRT - D_i > 0 \text{)}
\]
\[
= k \cdot C_i
\]

so, the deadline constraint can be satisfied. Therefore Lemma 7 is established. \(\square\)

**Proof of Lemma 8**

**Lemma 8.** If the synchronous bandwidth defined in (4.35) is allocated to any stream \(S_i\) with \(D_i \geq 2 \cdot TTRT\), \(\left\lfloor \frac{P_i}{TTRT} \right\rfloor = q_i\) and \(D_i - \left\lfloor (q_i + 1) TTRT - \frac{\max(q_i TTRT}{q_i - 1) C_i} \right\rfloor > 0\), then all the messages in that stream can be sent before their deadlines.

**Proof.** Because \(\left\lfloor \frac{P_i}{TTRT} \right\rfloor = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i\), we have
\[
P_i = q_i \cdot TTRT + r_P_i
\]
\[
D_i = q_i \cdot TTRT + r_D_i
\]

where \(0 \leq r_P_i < TTRT\) and \(0 \leq r_D_i < TTRT\).
Thus

\[
\frac{q_iTTRT}{P_i} = \frac{q_iTTRT}{q_iTTRT + r_P} \leq 1 \quad \text{(Since } 0 \leq r_P < TT_{RT})
\]

Recall the proof is under the condition \( D_i - [(q_i + 1)TT_{RT} - \frac{\max(q_iTTRT - 1)C_i}{q_i - 1}] > 0 \), from (4.35), we have

\[
H_i = \frac{C_i + (q_i + 1)TT_{RT} - D_i}{q_i}
\]  \hfill (F.24)

Let \( \Delta = \left\lfloor \frac{(k - 1) \cdot r_P + r_D}{TTR_{RT}} \right\rfloor \), we have

\[
\Delta = \left\lfloor \frac{(k - 1) \cdot r_P + r_D}{TTR_{RT}} \right\rfloor \leq \left\lfloor \frac{(k - 1) \cdot TT_{RT} + r_D}{TTR_{RT}} \right\rfloor \quad \text{(Since } r_P < TT_{RT})
\]

\[
= \left\lfloor (k - 1) + \frac{r_D}{TT_{RT}} \right\rfloor = (k - 1) + \left\lfloor \frac{r_D}{TT_{RT}} \right\rfloor
\]

\[
= k - 1 \quad \text{(Since } r_D < TT_{RT})
\]

Thus

\[
0 \leq \Delta \leq k - 1 \quad \text{where } \Delta \in \mathbb{N} \quad \text{(F.25)}
\]

This means under this case, we only need to consider two situations: \( 1 \leq \Delta \leq k - 1 \) and \( \Delta = 0 \).
Case 1: when $1 \leq \Delta \leq k - 1$

In this case

\[
x_k^i = \left[\frac{(k-1)P_i + D_i}{TTRT}\right] - 1 \cdot H_i \\
+ \max\{(k-1)P_i + D_i + H_i - \left[\frac{(k-1)P_i + D_i}{TTRT}\right] + 1\} \cdot TTRT, 0\}
\]

\[
\geq \left[\frac{(k-1)P_i + D_i}{TTRT}\right] - 1 \cdot H_i \\
= \left[\frac{(k-1)(q_iTTRT + r_P) + q_iTTRT + r_{D_i} - 1}{TTRT}\right] \cdot H_i \\
= \left[\frac{kq_iTTRT + (k-1)r_P + r_{D_i} - 1}{TTRT}\right] \cdot H_i \\
= (kq_i + \Delta - 1) \cdot H_i \\
\geq kq_i H_i \quad \text{(Since $\Delta \geq 1$)} \\
= kq_i C_i + \left(\frac{q_i + 1}{q_i}\right)TTRT - D_i \quad \text{(Since (F.24))} \\
= k \cdot C_i + k\left(\frac{q_i + 1}{q_i}\right)TTRT - D_i \\
> k \cdot C_i \quad \text{(Since $(q_i + 1)TTRT > D_i$)}
\]
Case 2: when $\Delta = 0$

Under this case, we have

$$x_k^i = \left[ \left( \frac{(k-1)P_i + D_i}{TTRT} \right) - 1 \right] \cdot H_i$$

$$+ \max\{((k-1)P_i + D_i + H_i - \left[ \left( \frac{(k-1)P_i + D_i}{TTRT} \right) + 1 \right] \cdot TTRT, 0\}$$

$$= \left[ \left( \frac{(k-1)(q_iTTRT + r_{P_i}) + q_iTTRT + r_{D_i} + H_i}{TTRT} \right) - 1 \right] \cdot H_i$$

$$+ \max\{((k-1)(q_iTTRT + r_{P_i}) + q_iTTRT + r_{D_i} + H_i - \left[ \left( \frac{(k-1)(q_iTTRT + r_{P_i}) + q_iTTRT + r_{D_i} + H_i}{TTRT} \right) + 1 \right] \cdot TTRT, 0\}$$

$$= (kq_i + \Delta - 1) \cdot H_i$$

$$+ \max\{(k-1)(q_iTTRT + r_{P_i}) + q_iTTRT + r_{D_i} + H_i - (kq_i + \Delta + 1) \cdot TTRT, 0\}$$

$$= (kq_i + \Delta - 1) \cdot H_i$$

$$+ \max\{(k-1)r_{P_i} + r_{D_i} + H_i - \Delta TTRT - TTRT, 0\}$$

$$= (kq_i - 1) \cdot H_i + \max\{(k-1)r_{P_i} + r_{D_i} + H_i - TTRT, 0\}$$

(Since $\Delta = 0$)

In order to show $x_k^i \geq k \cdot C_i$, we only need to show

$$(kq_i - 1) \cdot H_i + \max\{(k-1)r_{P_i} + r_{D_i} + H_i - TTRT, 0\} \geq k \cdot C_i \quad (F.26)$$

Sub-case 1: If $(k-1)r_{P_i} + r_{D_i} + H_i - TTRT < 0$

Here, we have

$$(k-1)r_{P_i} + r_{D_i} + H_i - TTRT < 0 \quad (F.27)$$

$$\Rightarrow (k-1)r_{P_i} + r_{D_i} + \frac{C_i + (q_i + 1)TTRT - D_i}{q_i} - TTRT < 0$$

(Since (F.24))

$$\Rightarrow (k-1)r_{P_i} + r_{D_i} + \frac{C_i + TTRT - D_i}{q_i} < 0$$

$$\Rightarrow (k-1)r_{P_i} + r_{D_i} + C_i + TTRT - D_i < 0$$

$$\Rightarrow -(k-1)r_{P_i} + r_{D_i} - TTRT + D_i > C_i \quad (F.28)$$
Since
\[(kq_i - 1) \cdot H_i + \max((k - 1)r_p, r_d_i + H_i - TTRT, 0) \geq k \cdot C_i\]
\[\Leftrightarrow (kq_i - 1) \cdot H_i \geq k \cdot C_i\]

(Since \((k - 1)r_p, r_d_i + H_i - TTRT < 0\))
\[\Leftrightarrow (kq_i - 1) \cdot \frac{C_i + (q_i + 1)TTRT - D_i}{q_i} \geq k \cdot C_i\]
\[(\text{Since } (F.24))\]
\[\Leftrightarrow (kq_i - 1) \cdot [C_i + (q_i + 1)TTRT - D_i] \geq kq_i C_i\]
\[\Leftrightarrow kq_i C_i + kq_i(q_i + 1)TTRT - kq_i D_i - C_i\]
\[\quad - (q_i + 1)TTRT + D_i \geq kq_i C_i\]
\[\Leftrightarrow kq_i(q_i + 1)TTRT - kq_i D_i + D_i - (q_i + 1)TTRT \geq C_i \quad \text{(F.29)}\]

Thus, in order to establish (F.26), we only need to show that (F.29) holds.

Since (F.28), in order to show (F.29), we only need to show

\[kq_i(q_i + 1)TTRT - kq_i D_i + D_i - (q_i + 1)TTRT \geq -(k - 1)r_p, q_i - r_d, q_i - TTRT + D_i\]
\[\Leftrightarrow kq_i(q_i + 1)TTRT - kq_i D_i - q_i TTRT + (k - 1)r_p, q_i + r_d, q_i \geq 0\]
\[\Leftrightarrow kq_i(q_i + 1)TTRT - kq_i(q_i TTRT + r_d) - q_i TTRT + (k - 1)r_p, q_i + r_d, q_i \geq 0\]
\[\Leftrightarrow kq_i^2TTRT + kq_i TTRT - kq_i^2 TTRT - kq_i r_d, q_i - q_i TTRT + (k - 1)r_p, q_i + r_d, q_i \geq 0\]
\[\Leftrightarrow (k - 1)q_i TTRT + (k - 1)r_p, q_i + (1 - k)r_d, q_i \geq 0\]
\[\Leftrightarrow (k - 1)q_i (TTRT + r_p, - r_d_i) \geq 0 \quad \text{(F.30)}\]

Since \(k \geq 1\) and \(r_d_i < TTRT \leq TTRT + r_p\), (F.30) can be established. Thus (F.26) is true, which means \(x_i^k \geq k \cdot C_i\).
Sub-case 2: If \((k - 1)r_{P_i} + r_{D_i} + H_i - TTRT \geq 0\)

In this Sub-case,

\[
(kq_i - 1) \cdot H_i + \max[(k - 1)r_{P_i} + r_{D_i} + H_i - TTRT, 0]
= (kq_i - 1) \cdot H_i + (k - 1)r_{P_i} + r_{D_i} + H_i - TTRT
\]

(Since \((k - 1)r_{P_i} + r_{D_i} + H_i - TTRT \geq 0\))

\[
= kq_i H_i - \{TTRT - [(k - 1)r_{P_i} + r_{D_i}]\}
= kq_i C_i + (q_i + 1)TTRT - D_i - \{TTRT - [(k - 1)r_{P_i} + r_{D_i}]\}
\]

(Since \((F.24)\))

\[
= kC_i + k[(q_i + 1)TTRT - D_i] - \{TTRT - [(k - 1)r_{P_i} + r_{D_i}]\}
= kC_i + k[(q_i + 1)TTRT - q_i TTRT - r_{D_i}] - \{TTRT - [(k - 1)r_{P_i} + r_{D_i}]\}
\]

\[
\geq kC_i + (TTRT - r_{D_i}) - \{TTRT - [(k - 1)r_{P_i} + r_{D_i}]\}
\]

(Since \(k \geq 1\) and \(TTRT > r_{D_i}\))

\[
= kC_i + (k - 1)r_{P_i}
\geq k \cdot C_i \quad \text{(Since} \ k \geq 1 \text{and} \ r_{P_i} \geq 0)\)

Thus \((F.26)\) holds and we have \(x^k \geq k \cdot C_i\).

Since \((F.2)\) can be established, based on Lemma 5, the deadline constraint can be satisfied. Therefore Lemma 8 can be established.

\[\square\]

**Proof of Lemma 9**

**Lemma 9.** If the synchronous bandwidth defined in (4.35) is allocated to any stream \(S_i\) with \(D_i \geq 2 \cdot TTRT\), \([rac{P_i}{TTRT}] + 2 = [rac{D_i}{TTRT}] = q_i\) and \(D_i - [q_i + 1)TTRT - \max(\frac{2TTRT}{q_i-1}, C_i) > 0\), then all the messages in that stream can be sent before their deadlines.

**Proof.** Because \([rac{P_i}{TTRT}] + 2 = [rac{D_i}{TTRT}] = q_i\), we have

\[
P_i = (q_i - 2) \cdot TTRT + r_{P_i},
D_i = q_i \cdot TTRT + r_{D_i},
\]

where \(0 \leq r_{P_i} < TTRT\) and \(0 \leq r_{D_i} < TTRT\).
Figure F.1: Relationship among \( kP_i, TTRT \) and \( (k-1)P_i + D_i \).

If \( r_{D_i} \geq r_{P_i} \), we have \( D_i = q_i TTRT + r_{D_i} \geq (q_i-2)TTRT + r_{P_i} + 2 \cdot TTRT = P_i + 2 \cdot TTRT \).

This situation has already been covered by Lemma 7, thus we only need to consider \( r_{D_i} < r_{P_i} \) here.

Since \( P_i = (q_i - 2) \cdot TTRT + r_{P_i} \), \( D_i = q_i TTRT + r_{D_i} \) and \( r_{D_i} < r_{P_i} \), we have

\[
P_i + TTRT < D_i < P_i + 2 \cdot TTRT
\]

and since \( \frac{q_i TTRT}{P_i} = \frac{(q_i-2)TTRT + 2 TTRT}{(q_i-2)TTRT + r_{P_i}} > 1 \), from (4.35), we have

\[
H_i = \frac{q_i TTRT}{P_i} C_i + \left( q_i + 1 \right) TTRT - D_i
\]  

(F.31)

Since \( P_i + TTRT < D_i < P_i + 2 \cdot TTRT \), we have \( kP_i + TTRT < (k-1)P_i + D_i < kP_i + 2 \cdot TTRT \). Fig. F.1 shows the relationship among \( kP_i, TTRT \) and \( (k-1)P_i + D_i \). The shadow area in Fig. F.1 shows the possible range of the \( (k-1)P_i + D_i \). Based on the actually used amount of the last \( H_i \) (before the deadline), we have three cases to consider here.

**Case 1: Non-availability of the Last \( H_i \)**

Fig. F.2 shows the situation that the last \( H_i \) sits beyond the deadline of the \( k^{th} \) message from \( S_i \), that is, in the time interval \( (k-1) \cdot P_i + D_i \), node \( i \) has no chance to use the last \( H_i \).

From Fig. F.2 we have

\[
(k-1) \cdot P_i + D_i < T - H_i
\]

where

\[
T = \left( \left\lfloor \frac{kP_i + TTRT}{TTRT} \right\rfloor + 1 \right) \cdot TTRT
\]

\[
= \left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2 \right) \cdot TTRT
\]

163
That is

\[
(k - 1) \cdot P_i + D_i < \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2\right)TTRT - H_i
\]

\[
\Leftrightarrow D_i < \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2\right)TTRT - H_i - (k - 1) \cdot P_i
\]

Thus we have

\[
D_i < \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2\right)TTRT - H_i - (k - 1) \cdot P_i
\]

\[
= \left(\left\lfloor \frac{kP_i + TTRT}{TTRT} \right\rfloor\right)TTRT + TTRT - H_i - (k - 1) \cdot P_i
\]

\[
= \left(\left\lfloor \frac{(k - 1)P_i + P_i + TTRT}{TTRT} \right\rfloor\right)TTRT + TTRT - H_i - (k - 1) \cdot P_i
\]

\[
< \left(\left\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \right\rfloor\right)TTRT + TTRT - H_i - (k - 1) \cdot P_i
\]

(Since \(P_i + TTRT < D_i\))

That is

\[
(k - 1)P_i + D_i + H_i - TTRT - \left(\left\lfloor \frac{(k - 1)P_i + D_i}{TTRT} \right\rfloor\right)TTRT < 0
\] (F.32)

Based on Lemma 5, in order to show the satisfaction of the requirement of the deadline constraint here, we only need to show (F.2).

Recall we have the conditions \(D_i - [(q_i + 1)TTRT - \frac{\max(q_{TTRT}, 1)C_i}{q_i - 1}] > 0\) and \(D_i \geq 2 \cdot TTRT\) here, we can have the equivalent condition (4.36). For clarity, it is restated here:

\[
D_i - (q_i + 1)TTRT + \frac{\max(a_{TTRT}, 1)C_i}{q_i - 1} > 0 \Leftrightarrow D_i - (q_i + 1)TTRT + H_i > 0
\] (F.33)

With (F.33) and (F.32), we have

\[
D_i - (q_i + 1)TTRT + H_i - \{(k - 1)P_i + D_i + H_i - TTRT - \left(\left\lfloor\frac{(k - 1)P_i + D_i}{TTRT}\right\rfloor\right)TTRT\} > 0
\]
Simplifying the above expression, we have

\[
\frac{(k-1)P_i + D_i}{TT\,RT} \cdot TT\,RT - (k-1)P_i - q_iTT\,RT > 0
\]

\[
\iff \left[ \frac{(k-1)P_i + D_i}{TT\,RT} \right] - 1]TT\,RT + TT\,RT - (k-1)P_i - q_iTT\,RT > 0
\]

\[
\iff \left[ \frac{(k-1)P_i + D_i}{TT\,RT} \right] - 1] > \frac{(k-1)P_i}{TT\,RT} + q_i - 1
\]

(F.34)

Thus, from (F.1) we have:

\[
x_i^k = \left[ \frac{(k-1)P_i + D_i}{TT\,RT} \right] - 1] \cdot H_i
\]

\[
+ \max\{((k-1)P_i + D_i + H_i - [\frac{(k-1)P_i + D_i}{TT\,RT} + 1] \cdot TT\,RT, 0\}
\]

\[
= \left[ \frac{(k-1)P_i + D_i}{TT\,RT} \right] - 1]H_i \quad \text{(Since (F.32))}
\]

\[
> \frac{(k-1)P_i}{TT\,RT} + q_i - 1]H_i \quad \text{(Since (F.34))}
\]

\[
= \left[ \frac{(k-1)P_i}{TT\,RT} + q_i - 1] \left[ \frac{g_{TT\,RT}C_i + (q_i + 1)TT\,RT - D_i}{P_i} \right] \quad \text{(By (F.31))}
\]

In order to show (F.2), we only need to show

\[
\left[ \frac{(k-1)P_i}{TT\,RT} + q_i - 1] \left[ \frac{g_{TT\,RT}C_i + (q_i + 1)TT\,RT - D_i}{P_i} \right] \geq kC_i
\]

because

\[
\left[ \frac{(k-1)P_i}{TT\,RT} + q_i - 1] \left[ \frac{g_{TT\,RT}C_i + (q_i + 1)TT\,RT - D_i}{P_i} \right] \geq kC_i
\]

\[
\iff (k-1)C_i + \frac{(k-1)P_i(q_i + 1)}{q_i} - \frac{(k-1)P_iD_i}{q_iTT\,RT}
\]

\[
+ (q_i - 1) \frac{TT\,RT \cdot C_i}{P_i} + (q_i - 1) \frac{(q_i + 1)TT\,RT}{q_i} - (q_i - 1) \frac{D_i}{q_i}
\]

\[
\iff (k-1)C_i(q_i) + (k-1)P_i(q_i + 1) - \frac{(k-1)P_iD_i}{TT\,RT}
\]

\[
+ (q_i - 1) \frac{TT\,RT \cdot C_i}{P_i}q_i + (q_i - 1)(q_i + 1)TT\,RT - (q_i - 1)D_i \geq kC_i q_i
\]

(F.35)

we only need to show that (F.35) holds, that is

\[
(k-1)P_i(q_i + 1) - \frac{(k-1)P_iD_i}{TT\,RT} + (q_i - 1)(q_i + 1)TT\,RT - (q_i - 1)D_i
\]

\[
\geq 1 - (q_i - 1) \frac{TT\,RT}{P_i} \iff q_iC_i
\]

(F.36)

Since \( P_i = (q_i - 2)TT\,RT + r_p = (q_i - 1)TT\,RT - (TT\,RT - r_p) < (q_i - 1)TT\,RT \), we have

\( 1 - (q_i - 1) \frac{TT\,RT}{P_i} < 0 \), thus we can establish (F.36) by showing

\[
(k-1)P_i(q_i + 1) - \frac{(k-1)P_iD_i}{TT\,RT} + (q_i - 1)(q_i + 1)TT\,RT - (q_i - 1)D_i \geq 0
\]

(F.37)
Case 2: Partial Usage of the Last $H_i$

Fig. F.3 shows the case that in the time interval $(k-1)P_i + D_i$, there is a chance to use a portion of the last $H_i$. In order to show the satisfaction of the deadline constraint, we only need to show that the $k$-th message can be sent before its deadline. Which means that in the time interval of $(k-1)P_i + D_i$, for $S_i$, there should be at least $k \cdot C_i$ available bandwidth.

From Fig. F.3 we see that the available amount of the synchronous bandwidth during the time interval of $(k-1)P_i + D_i$ is\footnote{Recall that under the condition $D_i \geq 2 \cdot TTRT$, in the worst case, the first $H_i$ cannot be used.}

$$x^k_i = ([kP_i \over TTRT] + 1)H_i + H_i$$

$$+ (H_i - \{([kP_i \over TTRT] + 2)TTRT - [(k-1)P_i + D_i]\})$$

$$= ([kP_i \over TTRT] + 1)H_i + (k-1)P_i + D_i - ([kP_i \over TTRT] + 2)TTRT$$

Thus, in order to prove the deadline constraint, we only need to show

$$([kP_i \over TTRT] + 1)H_i + (k-1)P_i + D_i - ([kP_i \over TTRT] + 2)TTRT \geq kC_i \quad (F.38)$$

There are two sub-cases to consider here:
Sub-case 1: $\frac{kP_i}{TTTRT} \not\in \mathbb{Z}$

Since $\frac{kP_i}{TTTRT} \not\in \mathbb{Z}$, we have $\lceil \frac{kP_i}{TTTRT} \rceil + 1 = \lceil \frac{kP_i}{TTTRT} \rceil$. Thus, (F.38) can be simplified as

$$
\left( \lceil \frac{kP_i}{TTTRT} \rceil \right) H_i + (k - 1)P_i + D_i - (\lceil \frac{kP_i}{TTTRT} \rceil + 1)TTTRT \geq kC_i
$$

From (F.39), we have,

$$
H_i \geq \frac{kC_i - (k - 1)P_i - D_i + \lceil \frac{kP_i}{TTTRT} \rceil TTTRT + TTTRT}{TTTRT} = TTTRT + \frac{kC_i - (k - 1)P_i - D_i + TTTRT}{\lceil \frac{kP_i}{TTTRT} \rceil}
$$

By (F.31), the above inequality can be expressed as

$$
\begin{align*}
TTTRT - \frac{k(P_i - C_i) + D_i - P_i - TTTRT}{\lceil \frac{kP_i}{TTTRT} \rceil} & \leq \frac{qTTTRT}{P_i} C_i + (q_i + 1)TTTRT - D_i \\
\Leftrightarrow TTTRT - \frac{k(P_i - C_i) + D_i - P_i - TTTRT}{\lceil \frac{kP_i}{TTTRT} \rceil} & \leq \frac{TTTRT}{P_i} C_i + TTTRT - D_i + TTTRT \\
\Leftrightarrow -\frac{k(P_i - C_i) + D_i - P_i - TTTRT}{\lceil \frac{kP_i}{TTTRT} \rceil} & \leq \frac{TTTRT}{P_i} C_i + TTTRT - D_i \\
\Leftrightarrow -\frac{kP_i + D_i - P_i - TTTRT}{\lceil \frac{kP_i}{TTTRT} \rceil} & \leq (\frac{TTTRT}{P_i} - \frac{k}{\lceil \frac{kP_i}{TTTRT} \rceil}) C_i
\end{align*}
$$

Since

$$
\frac{TTTRT}{P_i} - \frac{k}{\lceil \frac{kP_i}{TTTRT} \rceil} = \frac{\lceil \frac{kP_i}{TTTRT} \rceil TTTRT - kP_i}{P_i \lceil \frac{kP_i}{TTTRT} \rceil} > \frac{kP_i TTTRT - kP_i}{P_i \lceil \frac{kP_i}{TTTRT} \rceil} \quad \text{(Since } \frac{kP_i}{TTTRT} \not\in \mathbb{Z})
$$

$$
= 0
$$
(F.40) can be transformed as

\[\begin{align*}
C_i \geq & \quad \frac{kP_i + D_i - TTTRT}{TTTRP_i} - \frac{TTTR - D_i}{q_i} \\
= & \quad \frac{(D_i - TTTRT)\left(\frac{kP_i}{TTTR}\right) - (kP_i + D_i - TTTRT)q_i}{q_i\left(\frac{kP_i}{TTTR} - \frac{kP_i}{TTTR}\right) - kP_i} \\
= & \quad \frac{D_i - TTTRT}{q_i TTTR} P_i \\
& + \frac{(D_i - TTTRT)\left(\frac{kP_i}{TTTR}\right) - \frac{kP_i}{TTTR} q_i TTTR - q_i (D_i - TTTRT)}{q_i TTTR\left(\frac{kP_i}{TTTR} - \frac{kP_i}{TTTR}\right)} P_i \\
= & \quad \frac{D_i - TTTRT}{q_i TTTR} P_i \\
& + \frac{(D_i - TTTRT - q_i TTTR)\left(\frac{kP_i}{TTTR}\right) - q_i (D_i - TTTRT)}{q_i TTTR\left(\frac{kP_i}{TTTR} - \frac{kP_i}{TTTR}\right)} P_i \\
= & \quad \frac{D_i - TTTRT}{q_i TTTR} P_i \\
& + \frac{(q_i + 1) TTTR - D_i}{\left(\frac{kP_i}{TTTR}\right) - \frac{kP_i}{TTTR}} + q_i (D_i - TTTRT) - \frac{q_i TTTR}{q_i - 1} C_i P_i \\
\end{align*}\]

Recalling that the condition for this case is \(D_i - [(q_i + 1)TTTR - \frac{q_i TTTR}{q_i - 1} C_i] > 0\), we have

\[D_i - (q_i + 1) TTTR + \frac{q_i TTTR}{q_i - 1} C_i > 0\]

That is

\[C_i > \frac{[(q_i + 1) TTTR - D_i](q_i - 1)}{q_i TTTR} P_i\]
Thus, in order to establish (F.41), we only need to show
\[
\frac{[(q_i + 1)TTRT - D_i](q_i - 1)}{P_i} \geq P_i - \frac{kP_i}{TTRT} - \frac{kP_i}{TTRT} - \frac{kP_i}{TTRT} - \frac{kP_i}{TTRT}
\]
That is,
\[
[(q_i + 1)TTRT - D_i] - \frac{kP_i}{TTRT} - \frac{kP_i}{TTRT} - \frac{kP_i}{TTRT} - \frac{kP_i}{TTRT}
\]
That is,
\[
q_i^2TTRT - TTRT - q_iD_i + D_i
\]
That is,
\[
q_i^2TTRT - q_iD_i
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
\[
\frac{[(q_i + 1)TTRT - D_i]kP_i}{TTRT} + q_i(D_i - P_i - TTRT)
\]
That is,
That is
\[(q_i + 1)TTRT - D_i \frac{kP_i}{TTRT} + (q_i D_i - q_i^2 TTTRT) \frac{kP_i}{TTRT}
+ q_i (D_i - P_i - TTTRT)\]
\[\geq (q_i D_i - q_i^2 TTTRT) \left\lceil \frac{kP_i}{TTRT} \right\rceil \quad (F.42)\]

Since \((q_i + 1)TTRT > D_i\), we have \(\left\lceil \frac{(q_i + 1)TTRT - D_i}{kP_i} \right\rceil > 0\). So,

\[(q_i + 1)TTRT - D_i \frac{kP_i}{TTRT} + (q_i D_i - q_i^2 TTTRT) \frac{kP_i}{TTRT} + q_i (D_i - P_i - TTTRT)\]
\[> (q_i D_i - q_i^2 TTTRT) \frac{kP_i}{TTRT} + q_i (D_i - P_i - TTTRT)\]

Thus, in order to show (F.42), we only need to show

\[(q_i D_i - q_i^2 TTTRT) \frac{kP_i}{TTRT} + q_i (D_i - P_i - TTTRT)\]
\[\geq (q_i D_i - q_i^2 TTTRT) \left\lceil \frac{kP_i}{TTRT} \right\rceil \quad (F.43)\]

Recalling that \(P_i = (q_i - 2)TTRT + r P_i\) and \(D_i = q_i TTTRT + r D_i\), we have

\(r P_i < TTTRT\)
\[(q_i - 1)TTRT + r P_i < (q_i - 1)TTRT + TTTRT\]
\[(q_i - 2)TTRT + r P_i + TTTRT < q_i TTTRT\]
\(P_i + TTTRT < q_i TTTRT\)
\(q_i (P_i + TTTRT) < q_i^2 TTTRT\)
\(-q_i (P_i + TTTRT) > -q_i^2 TTTRT\)
\(q_i D_i - q_i (P_i + TTTRT) > q_i D_i - q_i^2 TTTRT\)
\(q_i (D_i - P_i - TTTRT) > q_i D_i - q_i^2 TTTRT\)

Thus

\[(q_i D_i - q_i^2 TTTRT) \frac{kP_i}{TTRT} + q_i (D_i - P_i - TTTRT)\]
\[> (q_i D_i - q_i^2 TTTRT) \frac{kP_i}{TTRT} + (q_i D_i - q_i^2 TTTRT)\]
\[= (q_i D_i - q_i^2 TTTRT) \left( \frac{kP_i}{TTRT} + 1 \right)\]
\[> (q_i D_i - q_i^2 TTTRT) \left\lceil \frac{kP_i}{TTRT} \right\rceil \quad (Since D_i \geq q_i TTTRT, \frac{kP_i}{TTRT} + 1 > \left\lfloor \frac{kP_i}{TTRT} \right\rfloor)\]

This shows that (F.43) can hold. Thus the deadline constraint is met for this sub-case.
Sub-case 2: \( \frac{kP_i}{TTRT} \in \mathbb{Z} \)

Based on the Lemma 5, in order to prove the deadline constraint, we only need to show that (F.2) holds for this sub-case.

From (F.1), we have the following derivations:

\[
x_i^k = \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] - 1 \cdot H_i + \max \{ (k-1)P_i + D_i + H_i - \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] + 1 \cdot TTTRT, 0 \} \\
\geq \left[ \frac{(k-1)P_i + D_i}{TTRT} \right] - 1 \cdot H_i \\
\geq \left[ \frac{(k-1)P_i + P_i + TTTRT}{TTRT} \right] - 1 \cdot H_i \quad \text{(Since } D_i > P_i + TTTRT \text{)} \\
= \left( \frac{kP_i + TTTRT}{TTRT} \right) - 1 \cdot H_i \quad \text{(Since } D_i > P_i + TTTRT \text{)} \\
= \left[ \frac{kP_i}{TTRT} \right] \cdot H_i \quad \text{(Since } \frac{kP_i}{TTRT} \in \mathbb{Z} \text{)} \\
= \frac{kP_i}{TTRT} \cdot H_i \quad \text{(Since } \frac{kP_i}{TTRT} \in \mathbb{Z} \text{)} \\
= \frac{kP_i}{TTRT} \cdot \frac{q_{TTRT} C_i + (q_i + 1) TTTRT - D_i}{q_i} \quad \text{(Since } (F.31) \text{)} \\
> \frac{kP_i}{TTRT} \cdot \frac{q_{TTRT} C_i}{q_i} \quad \text{(Since } (q_i + 1) TTTRT > D_i \text{)} \\
= k \cdot C_i
\]

Thus, the deadline constraint can be satisfied.

Case 3: Full Usage of the Last \( H_i \)

As in Case 2, in order to show the satisfaction of the requirement of the deadline constraint, we only need to show that in the time interval of \((k-1)P_i + D_i\), there should be at least \(kC_i\) available transmission time can be used.

From Fig. F.4, we see that the last \( H_i \) can be fully used. Thus, the available amount of
transmission time during the time interval of \((k - 1)P_i + D_i\) is

\[
x_i^k = \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor - 1 + 2 \right) \cdot H_i
\]

\[
= \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) \cdot H_i
\]

\[
> \frac{kP_i}{TTRT} \cdot H_i
\]

\[
> \frac{kP_i}{TTRT} \cdot \frac{qTTRT}{P_i} C_i + (q_i + 1)TTRT - D_i
\]

\[
> \frac{kP_i}{TTRT} \cdot \frac{qTTRT}{P_i} C_i
\]

\[
= k \cdot C_i
\]

Since (F.31)

Thus, the deadline constraint can be satisfied.

Now consider the different position of \(kP_i\) as depicted in Fig. F.5. The labelled last \(H_i\) in Fig. F.5 is not the real last \(H_i\) that the messages may use before the deadline. However, in order to compare to Fig. F.1, we still use this name to label this particular \(H_i\). Fig. F.1 and Fig. F.5 together demonstrate all the possible positions of \(kP_i\). The situation described in Fig. F.5 is not analysed from the beginning since it implies a better case than that in Fig. F.1. Compared with Fig. F.1, there maybe more chance to use the allocated synchronous bandwidth to transmit the messages. In particular,

1. In the time interval of \((k - 1)P_i + D_i\), a portion of the last \(H_i\) can always be consumed. The last \(H_i\) can also be fully used in some case, or there is also possibility to use a portion of the next arriving \(H_i\) to send messages. When either of these two situations happens (i.e.,

\[\text{Recall that under the condition } D_i \geq 2 \cdot TTRT, \text{ in the worst case, the first } H_i \text{ cannot be used.}\]
full use of the last $H_i$ or partial use of the next $H_i$ after last $H_i$), the available synchronous message transmitting time must be no less than

$$x_i^k = (\lfloor \frac{kP_i}{TTRT} \rfloor - 1 + 2) \cdot H_i \quad (F.45)$$

From Case 3 above, it is easy to show $x_i^k \geq k \cdot C_i$. Thus the deadline constraint can be satisfied.

2. Consider the case where only portion of the last $H_i$ can be consumed. Similar to Case 2, as depicted in Fig. F.3, the available amount of transmission time is

$$x_i^k = (\lfloor \frac{kP_i}{TTRT} \rfloor - 1)H_i + H_i$$

$$+ (H_i - \{(\lfloor \frac{kP_i}{TTRT} \rfloor + 2)TTRT - [(k - 1)P_i + D_i]\}) \quad (F.46)$$

Using a similar process described in Case 2 above, it is easy to check that the deadline constraint is met.

Based on the above discussion, Lemma 9 can be established.

Proof of Lemma 10

**Lemma 10.** If the synchronous bandwidth defined in (4.35) is allocated to any stream $S_i$ with $D_i \geq 2 \cdot TTRT$, $\lfloor \frac{P_i}{TTRT} \rfloor + 1 = \lfloor \frac{D_i}{TTRT} \rfloor = q_i$ and $D_i - [(q_i + 1)TTRT - \frac{\max(\frac{TTRT}{P_i},1)C_i}{q_i-1}] > 0$, then NOT all the messages in that stream can be sent before their deadlines.

**Proof.** To prove this lemma, it only needs to find out a message set which contains message streams whose conditions fulfil those indicated in this lemma, but these streams fail to meet the deadline constraint when the synchronous bandwidth allocated to this message set using the allocation scheme defined by (4.35).

One of such synchronous message sets is illustrated in Table F.1. The network contains four nodes. To simplify the discussion, let $\tau = 0$

We will show that when the synchronous bandwidths are allocated to these nodes using allocation scheme (4.35), the deadline constraints cannot be guaranteed.

For message stream $S_1$, there is $\lfloor \frac{P_1}{TTRT} \rfloor = \lfloor \frac{1.60 \times TTRT}{TTRT} \rfloor = 1$, $\lfloor \frac{D_1}{TTRT} \rfloor = \lfloor \frac{2.99 \times TTRT}{TTRT} \rfloor = 2$, that is

$$\lfloor \frac{P_1}{TTRT} \rfloor + 1 = \lfloor \frac{D_1}{TTRT} \rfloor = q_1 = 2$$

173
Also, we have

\[
D_1 - [(q_1 + 1)TTRT - \frac{\max(q_1 TTRT, 1)C_1}{q_1 - 1}] \\
= 2.99 \times TTRT - [(2 + 1)TTRT - \frac{\max(2TTRT, 1) \times 0.5 \times TTRT}{2 - 1}] \\
= 0.615 \times TTRT
\]

Thus, \( S_1 \) is the synchronous message stream which matches the conditions specified by this lemma. According to (4.35), the allocated bandwidth to this stream is

\[
H_1 = \frac{C_2 + (q_2 + 1)TTRT - D_2}{q_2} \quad \text{(By (F.24))}
\]

\[
= 0.48 \times TTRT + (4 + 1)TTRT - 4.95 \times TTRT \\
= 0.1325 \times TTRT
\]

With a similar analysis, the message streams \( S_2, S_3 \) and \( S_4 \) meet the requirements of Lemma 8, Lemma 9 and Lemma 6, correspondingly. Thus, the allocated synchronous bandwidths to the corresponding nodes are

\[
H_2 = \frac{C_3 + (q_3 + 1)TTRT - D_3}{q_3} \quad \text{(By (F.31))}
\]

\[
= \frac{10 \times TTRT \times 1.84 \times TTRT + (10 + 1)TTRT - 10.80 \times TTRT}{10}
\]

\[
= 0.25 \times TTRT
\]
\[ H_4 = \max \left( \frac{20 \times \text{TTRT}}{20 - 1}, 1 \right) C_4 \quad \text{(By (F.3))} \]
\[ = \max \left( \frac{20 \times \text{TTRT}}{20 - 1}, 1 \right) \times 1.14 \times \text{TTRT} \]
\[ = 0.30 \times \text{TTRT} \]

The above calculation shows that
\[ \sum_{i=1}^{n} H_i = 0.3175 \times \text{TTRT} + 0.1325 \times \text{TTRT} + 0.25 \times \text{TTRT} + 0.30 \times \text{TTRT} = \text{TTRT} - \tau \]
That is, the bandwidth allocated to this message set is a full allocation. Because under the full allocation situation, we have \( \sum_{i=1}^{n} H_i - \text{TTRT} - \tau = 0 \). Thus, no information from other nodes needs to be considered when verifying the deadline constraint of \( S_i \). The sufficient condition listed in (F.2) to test the deadline constraint, which is only based on node \( i \)'s local information is also a necessary condition. It is therefore reliable to use this condition to test the deadline constraint.

Lemma 8, Lemma 9 and Lemma 6 show that the deadline constraints of streams \( S_2 \), \( S_3 \) and \( S_4 \) can be satisfied, correspondingly. However, the problem turns up when considering the second message (i.e., \( k = 2 \)) from stream \( S_1 \).

From (F.1), we have
\[
X_{1}^{k}(\vec{H}_i)|_{k=2} = \left[ \left( \frac{(k-1)P_1 + D_1}{\text{TTRT}} \right) - 1 \right] \cdot H_1 \\
+ \max \{ (k-1)P_1 + D_1 + H_1 - \left[ \left( \frac{(k-1)P_1 + D_1}{\text{TTRT}} \right) + 1 \right] \cdot \text{TTRT}, 0 \} \\
= \left[ \left( \frac{(2-1) \times 1.60 \times \text{TTRT} + 2.99 \times \text{TTRT}}{\text{TTRT}} \right) - 1 \right] \times 0.3175 \times \text{TTRT} \\
+ \max \{ (2-1) \times 1.60 \times \text{TTRT} + 2.99 \times \text{TTRT} + 0.3175 \times \text{TTRT} \\
- \left[ \left( \frac{(2-1) \times 1.60 \times \text{TTRT} + 2.99 \times \text{TTRT}}{\text{TTRT}} \right) + 1 \right] \cdot \text{TTRT}, 0 \} \\
= 0.9525 \times \text{TTRT} + \max -0.0925 \times \text{TTRT}, 0 \\
= 0.9525 \times \text{TTRT}
\]

Sending these two messages needs at least \( 2 \cdot C_1 = 2 \times 0.5 \times \text{TTRT} = \text{TTRT} \) transmitting time, which is larger than the available transmitting time \( X_{1}^{k}(\vec{H}_i)|_{k=2} \). Thus, the deadline of the second message will be missed (i.e., condition (F.2) cannot hold).

\[ \square \]

**Proof of Theorem 9**

With the above 6 lemmas, we can prove Theorem 9 as follows.
Theorem 9. Under the protocol constraint, the deadline constraint can be satisfied for any stream \( S_i \) with \( D_i \geq 2 \cdot TTRT \) and \( D_i > P_i \) when the synchronous bandwidth is allocated to \( S_i \) using the allocation defined in (4.35) except when \( q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \).

Proof. To any message stream with \( D_i \geq 2 \cdot TTRT \) and \( D_i > P_i \):

1. When \( D_i - \lfloor (q_i + 1)TTRT - \frac{\max(\frac{D_i}{TTRT}, \frac{P_i}{TTRT})}{q_i - 1} \rfloor \leq 0 \), Lemma 6 shows that the allocation of synchronous bandwidth using (4.35) to any message stream \( S_i \) can guarantee the deadlines of the messages in that stream under the protocol constraint. This implies the deadline constraint holds even when \( q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \).

2. When \( D_i - \lfloor (q_i + 1)TTRT - \frac{\max(\frac{D_i}{TTRT}, \frac{P_i}{TTRT})}{q_i - 1} \rfloor > 0 \)

Because \( D_i > P_i \), to \( \forall \) \( D_i, P_i \), there is

\[
\left\lfloor \frac{D_i}{TTRT} \right\rfloor - \left\lfloor \frac{P_i}{TTRT} \right\rfloor \in \{k | k \geq 0\}
\]

i.e., to \( \forall \) \( D_i, P_i \), the difference between \( \lfloor \frac{D_i}{TTRT} \rfloor \) and \( \lfloor \frac{P_i}{TTRT} \rfloor \) can only be a non negative integer. To consider all the possibilities, it only needs to consider all the possible values of the above difference.

When \( \left\lfloor \frac{D_i}{TTRT} \right\rfloor - \left\lfloor \frac{P_i}{TTRT} \right\rfloor \geq m = 3 \), i.e.,

\[
D_i = q_iTTRT + r_{D_i},
\]

\[
P_i = (q_i - m)TTRT + r_{P_i},
\]

we have,

\[
D_i - P_i = q_iTTRT + r_{D_i} - [(q_i - m)TTRT + r_{P_i}]
\]

\[
= m \cdot TTRT + r_{D_i} - r_{P_i}
\]

\[
> m \cdot TTRT - TTRT \quad (\text{Since } r_{D_i} - r_{P_i} < TTRT)
\]

\[
= 3 \cdot TTRT - TTRT \quad (\text{Since } m = 3)
\]

\[
= 2 \cdot TTRT
\]

This result is covered by \( D_i - P_i \geq 2 \cdot TTRT \). Thus, Lemma 7 analyses all the possible synchronous message stream \( S_i \) with the difference between \( \lfloor \frac{D_i}{TTRT} \rfloor \) and \( \lfloor \frac{P_i}{TTRT} \rfloor \) no less than 3. Other possible values of this difference, i.e., 0, 1 and 2, are discussed in Lemma 8, Lemma 10 and Lemma 9, correspondingly. Based on these four lemmas, the allocation of synchronous bandwidth using (4.35) may \textbf{MISS} some messages’ deadlines only when these messages are in the message stream \( S_i \) with \( q_i = \lfloor \frac{D_i}{TTRT} \rfloor = \lfloor \frac{P_i}{TTRT} \rfloor + 1 \).

Theorem 9 follows from the above discussion. \( \square \)
Appendix G

Proof of Theorem 10

Theorem 10. When the synchronous bandwidth allocated to stream $S_i$ with $D_i \geq 2 \cdot TTRT$, 
\[
\left\lfloor \frac{P_i}{TTRT} \right\rfloor + 1 = \left\lfloor \frac{D_i}{TTRT} \right\rfloor = q_i \quad \text{and} \quad D_i - (q_i + 1)TTRT + H_i > 0\]
using Eq. (4.35), not all the messages from $S_i$, which satisfy either of the following two groups of conditions, can be transmitted before their deadlines, when only judged by node $i$’s local information.

The first group of conditions:

\[
\begin{cases} 
\left\lfloor \frac{k r_P}{TTRT} \right\rfloor TTRT < (k-1)r_P + r_D, \text{ and} \\
\left\lfloor \frac{k r_P}{TTRT} \right\rfloor TTRT > (k-1)r_P, \text{ and} \\
H_i > TTRT - r_D, 
\end{cases}
\]

(G.1)

The second group of conditions:

\[
\begin{cases} 
(k-1)r_P + r_D < \left( \left\lfloor \frac{k r_P}{TTRT} \right\rfloor + 1 \right) TTRT \text{ and} \\
\left\lfloor \frac{k r_P}{TTRT} \right\rfloor TTRT > (k-1)r_P, \text{ and} \\
H_i \geq \left( \left\lfloor \frac{k r_P}{TTRT} \right\rfloor TTRT - (k-1)r_P \right) + TTRT - r_D, 
\end{cases}
\]

(G.2)

where $k \geq 2$, $r_D = D_i - q_i \cdot TTRT$ and $r_P = P_i - (q_i - 1) \cdot TTRT$.

Proof. As it is stated in Section 4.2.2, the expected synchronous bandwidth $H_i$ allocated to message stream $S_i$ should definitely meet the deadline constraint for all messages from $S_i$, if the protocol constraint of any message set which $S_i$ belongs to can be satisfied. Thus, we still
consider the question about meeting the deadline constraint based on node \( i \)'s local information, i.e., parameters of \( S_i \) (i.e., \( C_i, P_i, D_i \)), \( TTRT \) and \( \tau \).

Lemma 10 in the Appendix F only gives conditions of message stream \( S_i \) whose deadline constraint may fail to be met when (4.35) is used to allocate synchronous bandwidth. Here, we are going to narrow down the range of messages from \( S_i \) whose deadlines cannot be guaranteed when only based upon the locally available information.

Because \( \lfloor \frac{P_i}{TTRT} \rfloor + 1 = \lfloor \frac{D_i}{TTRT} \rfloor = q_i \), we have

\[
P_i = (q_i - 1) \cdot TTRT + r_{P_i} \geq TTRT \quad \text{(Since } q_i \geq 2)\]
\[
D_i = q_i \cdot TTRT + r_{D_i}\]

where \( 0 \leq r_{P_i} < TTRT \) and \( 0 \leq r_{D_i} < TTRT \).

Since \( \frac{q_i}{P_i} \frac{TTRT}{P_i} = \frac{(q_i - 1)TTRT + \frac{TTRT}{P_i}}{(q_i - 1)TTRT + \frac{TTRT}{P_i}} > 1 \), from (4.35), we have

\[
H_i = \frac{q_i}{P_i} \frac{TTRT}{P_i} C_i + (q_i + 1)TTRT - D_i \quad \text{(G.3)}
\]

\[
= \frac{q_i}{P_i} \frac{TTRT}{P_i} C_i + (q_i + 1)TTRT - (q_i TTRT + r_{D_i})
\]

\[
= \frac{TTRT}{P_i} C_i + \frac{TTRT - r_{D_i}}{q_i} \quad \text{(G.4)}
\]

Recall that the conditions of \( D_i - [(q_i + 1)TTRT - \frac{\max(\frac{q_i TTRT}{q_i - 1}, C_i)}{q_i - 1}] > 0 \) and \( D_i \geq 2 \cdot TTRT \) here imply an equivalent condition expressed in (4.36). For the convenience of discussion, it is restated here:

\[
D_i - (q_i + 1)TTRT + \frac{\max(\frac{q_i TTRT}{P_i}, C_i)}{q_i - 1} > 0 \Leftrightarrow D_i - (q_i + 1)TTRT + H_i > 0 \quad \text{(G.5)}
\]

For the special case of \( k = 1 \), we have

\[
x_{r_i}^{k} |_{k=1} = \left( \lfloor \frac{D_i}{TTRT} \rfloor - 1 \right) \cdot H_i + \max \{ D_i + H_i - (\lfloor \frac{D_i}{TTRT} \rfloor + 1)TTRT, 0 \}
\]

\[
= (q_i - 1)H_i + \max \{ D_i + H_i - (q_i + 1)TTRT, 0 \}
\]

\[
= (q_i - 1)H_i + D_i + H_i - (q_i + 1)TTRT \quad \text{(Since (G.5))}
\]

\[
= q_i \cdot H_i + D_i - (q_i + 1)TTRT
\]

\[
= q_i \cdot \frac{q_i}{P_i} \frac{TTRT}{P_i} C_i + (q_i + 1)TTRT - D_i + D_i - (q_i + 1)TTRT
\]

\[
= \frac{q_i TTRT}{P_i} C_i + (q_i + 1)TTRT - D_i + D_i - (q_i + 1)TTRT
\]

\[
= \frac{q_i TTRT}{P_i} C_i
\]

\[
> C_i \quad \text{(Since } \frac{q_i TTRT}{P_i} > 1) \quad \text{(G.6)}
\]

178
Figure G.1: The possible range of \((k-1)P_i + D_i\) relative to \(k \cdot P_i\).

Thus, according to Lemma 5, the deadline constraint can be satisfied for the first (i.e., \(k = 1\)) message.

Therefore, the following discussion only considers the situation when \(k \geq 2\).

Because

\[-TTRT < r_{D_i} - r_{P_i} < TTTR\]

we have

\[k \cdot P_i < k \cdot P_i + TTTR + (r_{D_i} - r_{P_i}) < k \cdot P_i + 2 \cdot TTTR\]

Since

\[(k - 1) \cdot P_i + D_i = k \cdot P_i + D_i - P_i\]
\[= k \cdot P_i + q_i \cdot TTTR + r_{D_i} - [(q_i - 1) \cdot TTTR + r_{P_i}]\]
\[= k \cdot P_i + TTTR + (r_{D_i} - r_{P_i})\]

we have

\[k \cdot P_i < (k - 1)P_i + D_i < k \cdot P_i + 2 \cdot TTTR\]

With \(P_i \geq TTTR\) and \(k \geq 2\), we have \(k \cdot P_i \geq 2 \cdot TTTR\). Thus, we have

\[2 \cdot TTTR \leq k \cdot P_i < (k - 1)P_i + D_i < k \cdot P_i + 2 \cdot TTTR\]  \(\text{(G.7)}\)

Fig. G.1 shows the relationship among \(k \cdot P_i\), \(TTTR\) and \((k - 1)P_i + D_i\) in the worst case. The shadow area in Fig. G.1 shows the possible range of the \((k - 1)P_i + D_i\). There are five cases to consider based on the sub range in which the \((k - 1)P_i + D_i\) resides. Fig. G.2 to Fig. G.6 depict these five cases, correspondingly. To make the following discussion simple, we label the two allocated synchronous bandwidth \(H_i\) in the shadow area as the second last \(H_i\) and the last \(H_i\) from left to right.
Figure G.2: Non-availability of the second last $H_i$

Figure G.3: Partial usage of the second last $H_i$

Figure G.4: Full usage of the last $H_i$

Figure G.5: Full usage of the second last $H_i$ but non-availability of the last $H_i$
Case 1: Non-availability of the Second Last \( H_i \)

From Fig. G.2 we have
\[
(k - 1) \cdot P_i + D_i \leq (\lfloor \frac{kP_i}{TTRT} \rfloor + 1) \cdot TTTRT - H_i
\]
\[
\Leftrightarrow (k - 1)P_i + D_i + H_i - (\lfloor \frac{kP_i}{TTRT} \rfloor + 1)TTRT \leq 0 \tag{G.8}
\]

This shows the situation that the second last \( H_i \) sits beyond the deadline of the \( k \)-th message from \( S_i \), that is, in the time interval \( (k - 1) \cdot P_i + D_i \), and there is no chance of the used amount of the second last \( H_i \).

From (G.5), we also have
\[
D_i - (q_i + 1) \cdot TTTRT + H_i > 0 \tag{G.9}
\]

From (G.9) and (G.8), we have
\[
D_i - (q_i + 1)TTTRT + H_i > (k - 1)P_i + D_i + H_i - (\lfloor \frac{kP_i}{TTRT} \rfloor + 1)TTRT
\]
\[
\Leftrightarrow - (q_i + 1)TTTRT > (k - 1)P_i - (\lfloor \frac{kP_i}{TTRT} \rfloor + 1)TTRT
\]
\[
\Leftrightarrow | \frac{kP_i}{TTRT} | TTTRT > (k - 1)P_i + q_i TTTRT
\]
\[
\Leftrightarrow | \frac{kP_i}{TTRT} | TTTRT > (k - 1)P_i + q_i TTTRT + | \frac{D_i}{TTTRT} |
\]
\[
\Leftrightarrow | \frac{kP_i}{TTRT} | > (k - 1)P_i + q_i TTTRT + | \frac{D_i}{TTTRT} | \tag{G.10}
\]

However, because
\[
(k - 1)P_i = kP_i - P_i
\]
\[
= kP_i - [(q_i - 1)TTTRT + rP_i]
\]
\[
= kP_i - (q_i - 1)TTTRT - rP_i
\]
\[
> kP_i - (q_i - 1)TTTRT - TTTRT \quad \text{(Since } rP_i < TTTRT \text{)}
\]
\[
= kP_i - q_i TTTRT
\]
we also have
\[
\frac{(k-1)P_i}{TTRT} > \frac{kP_i - q_iTTRT}{TTRT} = \frac{kP_i}{TTRT} - q_i
\]
This implies
\[
\frac{(k-1)P_i}{TTRT} + \left\lfloor \frac{D_i}{TTRT} \right\rfloor > \frac{kP_i}{TTRT} - q_i + \left\lfloor \frac{D_i}{TTRT} \right\rfloor
= \frac{kP_i}{TTRT} - q_i + q_i
= \frac{kP_i}{TTRT}
\geq \left\lfloor \frac{kP_i}{TTRT} \right\rfloor
\]
which contradicts with (G.10).

The above analysis shows that this case never happens in the real situation, i.e., the deadline of the k-th message from \( S_i \) is always later than the arrival of the second last \( H_i \).

**Case 2: Partial Usage of the Second Last \( H_i \)**

In this situation, Fig. G.3 gives the following condition:
\[
\left\lceil \frac{kP_i}{TTRT} \right\rceil + 1)TTRT - H_i < (k-1)P_i + D_i \leq \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1)TTRT \tag{G.11}
\]
and also
\[
x^k_i = \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor - 1\right)H_i + \{(k-1)P_i + D_i - \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1\right)TTRT - H_i\}
= \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor - 1\right)H_i + (k-1)P_i + D_i - \left\lfloor \frac{kP_i}{TTRT} \right\rfloor TTRT - TTRT
= \left\lfloor \frac{kP_i}{TTRT} \right\rfloor \cdot H_i + (k-1)P_i + D_i - \left\lfloor \frac{kP_i}{TTRT} \right\rfloor TTRT - TTRT
\]
According to Lemma 5, in order to prove the deadline constraint, it only needs to show that
\[
x^k_i \geq k \cdot C_i
\]
\[
\iff \left\lfloor \frac{kP_i}{TTRT} \right\rfloor \cdot H_i + (k-1)P_i + D_i - \left\lfloor \frac{kP_i}{TTRT} \right\rfloor TTRT - TTRT \geq k \cdot C_i
\]
Since \( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor \geq 2 \), and also \( k \geq 2 \) and \( P_i \geq TTRT \), we have,
\[
H_i \geq \frac{kC_i + TTRT - (k-1)P_i - D_i + \left\lfloor \frac{kP_i}{TTRT} \right\rfloor TTRT}{\left\lfloor \frac{kP_i}{TTRT} \right\rfloor} \tag{G.12}
\]
From (G.3), in order to show (G.12), we only need to show

\[
\frac{q TTTRT}{P_i} C_i + (q_i + 1) TTTRT - D_i \geq k C_i + TTTRT - (k - 1) P_i - D_i + \frac{k P_i}{TTTRT} TTTRT
\]

Because

\[
TTTRT > r
\]

we have

\[
TTTRT - r_{D_i} > r_P - r_{D_i}
\]

(G.14)

\[
TTTRT - r_{D_i} > 0
\]

(G.15)
Also, we have

\[ q_i \geq 2 \quad \text{(Since } D_i \geq 2 \cdot TTRT) \]
\[ \iff 2 \cdot q_i - q_i \geq 2 \]
\[ \iff 2 \cdot q_i - 2 \geq q_i \]
\[ \iff 2 \cdot (q_i - 1) \geq q_i \]

Thus,

\[
\left\lfloor \frac{kP_i}{TTRT} \right\rfloor \geq \left\lfloor \frac{2 \cdot P_i}{TTRT} \right\rfloor \quad \text{(Since } k \geq 2) \]
\[
= \left\lfloor \frac{2 \cdot [(q_i - 1)TTRT + r_p]}{TTRT} \right\rfloor \]
\[
= \left\lfloor \frac{2 \cdot (q_i - 1)TTRT + 2 \cdot r_p}{TTRT} \right\rfloor \]
\[
\geq \left\lfloor \frac{2 \cdot (q_i - 1)TTRT}{TTRT} \right\rfloor \]
\[
= 2 \cdot (q_i - 1) \]
\[
\geq q_i > 0 \quad \text{(G.16)}
\]

From (G.14), (G.15) and (G.16), (G.13) can be established, which leads to \( X_k^i \geq k \cdot C_i \).

Therefore, the deadline constraints can be satisfied.

**Case 3: Full Usage of the Last \( H_i \)**

The condition for this case, as shown in Fig. G.4, is

\[
\left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2 \right)TTRT < (k - 1)P_i + D_i < kP_i + 2 \cdot TTRT
\]

In order to show the satisfaction of the requirement of the deadline constraint, it is only need to show in the time interval of \((k - 1)P_i + D_i\), available transmission time should be at least \( k \cdot C_i \).

From Fig. G.4, the last \( H_i \) can be fully used. Thus, the available transmission time during
the time interval of \((k - 1)P_i + D_i\) is

\[
x_i^k = \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2 - 1\right) \cdot H_i
\]

\[
= \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1\right) \cdot H_i
\]

\[
> \frac{kP_i}{TTRT} \cdot H_i
\]

\[
= \frac{kP_i}{TTRT} \cdot \frac{q_iTTRT}{p_i} C_i + \left(\frac{q_iTTRT}{p_i} + D_i\right)
\]

\[
> \frac{kP_i}{TTRT} \cdot \frac{q_iTTRT}{p_i} C_i
\]

\[
= k \cdot C_i
\]

(Since \((G.3))\)

\[
> \frac{kP_i}{TTRT} \cdot \frac{q_iTTRT}{p_i} C_i
\]

\[
= \frac{kP_i}{TTRT} \cdot \frac{q_iTTRT}{p_i} C_i
\]

\[
= k \cdot C_i
\]

Thus, the deadline constraint can be satisfied.

**Case 4: Full Usage of the Second Last \(H_i\) but Non-availability of the Last \(H_i\)**

Fig. G.5 showing the condition for this case is

\[
\left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1\right)TTRT < (k - 1)P_i + D_i \leq \left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2\right)TTRT - H_i
\]

\[
\Leftrightarrow \left(k\left\lfloor \frac{(q_i - 1)TTRT + r_p_i}{TTRT} \right\rfloor + 1\right)TTRT
\]

\[
< (k - 1)((q_i - 1)TTRT + r_p_i) + (q_iTTRT + r_D_i)
\]

\[
\leq \left(k\left\lfloor \frac{(q_i - 1)TTRT + r_p_i}{TTRT} \right\rfloor + 2\right)TTRT - H_i
\]

\[
\Leftrightarrow (k(q_i - 1) + \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 1)TTRT < (k - 1)((q_i - 1)TTRT + r_p_i) + (q_iTTRT + r_D_i)
\]

\[
\leq (k(q_i - 1) + \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 2)TTRT - H_i
\]

\[
\Leftrightarrow (k(q_i - 1) + \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 1)TTRT
\]

\[
< (k - 1)(q_i - 1)TTRT + (k - 1)r_p_i + (q_i - 1)TTRT + TTTRT + r_D_i
\]

\[
\leq (k(q_i - 1) + \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 2)TTRT - H_i
\]

\[
\Leftrightarrow (k(q_i - 1) + \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 1)TTRT < k(q_i - 1)TTTRT + TTTRT + (k - 1)r_p_i + r_D_i
\]

\[
\leq (k(q_i - 1) + \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 2)TTRT - H_i
\]

\[
\Leftrightarrow \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor TTTRT < (k - 1)r_p_i + r_D_i \leq \left\lfloor \frac{kq_iTTRT}{TTRT} \right\rfloor + 1)TTRT - H_i
\]

\[
(G.18)
\]
Recall that we also have (4.36), i.e.,

\[ D_i - (q_i + 1)TTRT + H_i > 0 \]

\[ \iff q_iTTRT + r_{Di} - (q_i + 1)TTRT + H_i > 0 \]

\[ \iff r_{Di} - TTTR + H_i > 0 \]

(G.19)

From the two conditions (G.18) and (G.19), we have

\[
\begin{cases}
\lfloor \frac{kr_{Pi}}{TTRT} \rfloor TTRT < (k-1)r_{Pi} + r_{Di} \\
(k-1)r_{Pi} + r_{Di} \leq (\lfloor \frac{kr_{Pi}}{TTRT} \rfloor + 1)TTRT - H_i \\
r_{Di} - TTTR + H_i > 0 \\
\lfloor \frac{kr_{Pi}}{TTRT} \rfloor TTRT < (k-1)r_{Pi} + r_{Di} \\
H_i \leq \lfloor \frac{kr_{Pi}}{TTRT} \rfloor TTRT - (k-1)r_{Pi} + TTTR - r_{Di} < TTTR
\end{cases}
\]

(G.20)

(G.20) also implies that when \( \lfloor \frac{kr_{Pi}}{TTRT} \rfloor TTRT \leq (k-1)r_{Pi} \), \( H_i \) will not exist. This means that the case here cannot happen. Thus, in this case, we only need to consider

\[
\lfloor \frac{kr_{Pi}}{TTRT} \rfloor TTRT > (k-1)r_{Pi} \] 

(G.21)

In order to prove the deadline constraint, according to Lemma 5, it only needs to show \( x_k^i \geq k \cdot C_i \). From Fig. G.6, we have

\[
x_k^i = (\lfloor \frac{kP_i}{TTRT} \rfloor - 1 + 1) \cdot H_i
\]

\[ = \lfloor \frac{kP_i}{TTRT} \rfloor \cdot H_i \] 

(G.22)
Thus, we only need to show

\[ \left\lfloor \frac{kP_i}{TTTRT} \right\rfloor \cdot \left( \frac{TTTRT}{P_i} C_i + \frac{TTTRT - r_{D_i}}{q_i} \right) \geq k \cdot C_i \quad \text{(Since G.4)} \]

\[ \left\lfloor \frac{k((q_i - 1)TTTRT + r_{D_i})}{TTTRT} \right\rfloor \cdot \left( \frac{TTTRT}{P_i} C_i + \frac{TTTRT - r_{D_i}}{q_i} \right) \geq k \cdot C_i \]

\[ \left\lfloor \frac{k(q_i - 1)TTTRT - r_{D_i}}{TTTRT} \right\rfloor \cdot \left( \frac{TTTRT}{P_i} C_i + \frac{TTTRT - r_{D_i}}{q_i} \right) \geq k \cdot C_i \]

\[ \left\lfloor \frac{k(q_i - 1)TTTRT - r_{D_i}}{TTTRT} \right\rfloor \cdot \left( \frac{TTTRT}{P_i} C_i + \frac{TTTRT - r_{D_i}}{q_i} \right) \geq \left( \frac{kP_i}{TTTRT} \right) \cdot \left( \frac{TTTRT}{P_i} C_i \right) \]

(G.23)

From the condition (G.20), we have

\[ H_i < \left\lfloor \frac{kP_i}{TTTRT} \right\rfloor TTTRT - (k - 1)r_{P_i} + TTTRT - r_{D_i} \]

\[ \left\lfloor \frac{k(q_i - 1)TTTRT - r_{D_i}}{TTTRT} \right\rfloor \cdot \left( \frac{TTTRT}{P_i} C_i + \frac{TTTRT - r_{D_i}}{q_i} \right) \geq \left( \frac{kP_i}{TTTRT} \right) \cdot \left( \frac{TTTRT}{P_i} C_i \right) \]

\[ \left\lfloor \frac{k(q_i - 1)TTTRT - r_{D_i}}{TTTRT} \right\rfloor \cdot \left( \frac{TTTRT}{P_i} C_i + \frac{TTTRT - r_{D_i}}{q_i} \right) \geq \left( \frac{kP_i}{TTTRT} \right) \cdot \left( \frac{TTTRT}{P_i} C_i \right) \]

(G.24)

The above analysis shows that under this case, in order to guarantee the deadline constraint, we need to show (G.24) holds under \( \left\lfloor \frac{kP_i}{TTTRT} \right\rfloor TTTRT > (k - 1)r_{P_i} \). We leave the discussion on this after the next case.
Case 5: Partial Usage of the Last $H_i$

From Fig. G.6, in this case, we have

$$
\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2)TTRT - H_i < (k - 1)P_i + D_i \leq \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2)TTRT
\Rightarrow \left\lfloor \frac{k[(q_i - 1)TTRT + r_P]}{TTRT} \right\rfloor + 2)TTRT - H_i
\leq (k(q_i - 1)TTRT + r_P) + (q_iTTRT + r_D_i)
\leq \left\lfloor \frac{k[(q_i - 1)TTRT + r_P]}{TTRT} \right\rfloor + 2)TTRT
\Rightarrow (k(q_i - 1) + \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2)TTRT - H_i
\leq (k(q_i - 1)TTRT + (k - 1)r_P + (q_i - 1)TTRT + TTRT + r_D_i)
\leq (k(q_i - 1) + \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2)TTRT
\Rightarrow (k(q_i - 1) + \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2)TTRT - H_i < k(q_i - 1)TTRT + TTRT + (k - 1)r_P + r_D_i
\leq (k(q_i - 1) + \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 2)TTRT
\Rightarrow \left\lfloor \frac{kP_i}{TTRT} \right\rfloor TTRT + TTRT - H_i < (k - 1)r_P + r_D_i \leq \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1)TTRT \quad \text{(G.25)}
$$

Similarly to in Case 4, we also have the condition of (G.19). From (G.25) and (G.19), we
have

\[
\begin{align*}
(k - 1) r_{P_i} + r_{D_i} & \leq (\lfloor \frac{kr_{P_i}}{TTRT} \rfloor + 1) TTRT \\
\lfloor \frac{kr_{P_i}}{TTRT} \rfloor TTRT + TTRT - H_i & < (k - 1) r_{P_i} + r_{D_i} \\
r_{D_i} - TTRT + H_i & > 0
\end{align*}
\]

\(\text{(G.26)}\)

\[
\begin{align*}
(k - 1) r_{P_i} + r_{D_i} & < (\lfloor \frac{kr_{P_i}}{TTRT} \rfloor + 1) TTRT \\
H_i & \geq \lfloor \frac{kr_{P_i}}{TTRT} \rfloor TTRT - (k - 1) r_{P_i} + TTRT - r_{D_i} \\
H_i & > TTRT - r_{D_i}
\end{align*}
\]

\(\text{(G.27)}\)

\[
\begin{align*}
(k - 1) r_{P_i} + r_{D_i} & < (\lfloor \frac{kr_{P_i}}{TTRT} \rfloor + 1) TTRT \\
H_i & \geq \max\{\lfloor \frac{kr_{P_i}}{TTRT} \rfloor TTRT - (k - 1) r_{P_i} + TTRT - r_{D_i}, TTRT - r_{D_i}\}
\end{align*}
\]

\(\text{(G.28)}\)

\[
\begin{align*}
(k - 1) r_{P_i} + r_{D_i} & < (\lfloor \frac{kr_{P_i}}{TTRT} \rfloor + 1) TTRT \\
H_i & \geq TTRT - r_{D_i}
\end{align*}
\]

(When \(\lfloor \frac{kr_{P_i}}{TTRT} \rfloor TTRT \leq (k - 1) r_{P_i}\))

or

\[
\begin{align*}
(k - 1) r_{P_i} + r_{D_i} & < (\lfloor \frac{kr_{P_i}}{TTRT} \rfloor + 1) TTRT \\
H_i & \geq \lfloor \frac{kr_{P_i}}{TTRT} \rfloor TTRT - (k - 1) r_{P_i} + TTRT - r_{D_i}
\end{align*}
\]

(When \(\lfloor \frac{kr_{P_i}}{TTRT} \rfloor TTRT > (k - 1) r_{P_i}\))

\(\text{(G.29)}\)
In order to show the deadline constraints, we only need to show $x^k_i \geq k \cdot C_i$. From Fig. G.6,

\[
x^k_i = \left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) H_i + (k - 1) P_i + D_i - \left\lfloor \left( \frac{kP_i}{TTRT} \right) + 2 \right\rfloor TTRT - H_i
\]

\[
= \left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) H_i + (k - 1) P_i + D_i - 2 \cdot TTTR T - \frac{kP_i}{TTRT} TTRT
\]

\[
= \left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) \left( \frac{TTRT}{P_i} C_i + \frac{TTRT - r_{D_i}}{q_i} \right)
\]

\[
+ (k - 1) P_i + D_i - 2 \cdot TTRT - \frac{kP_i}{TTRT} TTRT \quad \text{(Since G.4)}
\]

\[
= \left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) \left( \frac{TTRT}{P_i} C_i + \frac{TTRT - r_{D_i}}{q_i} \right)
\]

\[
+ (k - 1) \left[ (q_i - 1) TTRT + r_{P_i} \right] + q_i TTRT + r_{D_i} - 2 \cdot TTRT
\]

\[
- \frac{k(q_i - 1) TTRT + r_{P_i}}{TTRT} TTRT
\]

\[
= \left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) \left( \frac{TTRT}{P_i} C_i + \frac{TTRT - r_{D_i}}{q_i} \right)
\]

\[
+ (k - 1) r_{P_i} + r_{D_i} - TTRT - \frac{kr_p}{TTRT} TTRT
\]

Thus, we only need to show

\[
\left( \left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1 \right) \left( \frac{TTRT}{P_i} C_i + \frac{TTRT - r_{D_i}}{q_i} \right) + (k - 1) r_{P_i} + r_{D_i} - TTRT - \frac{kr_p}{TTRT} TTRT
\]

\[
\geq k \cdot C_i
\]
That is,

\[
\left\lfloor k \frac{P_i}{TTRT} \right\rfloor + 1 + \frac{TTRT - r_{D_i}}{q_i} + (k - 1)r_{P_i} + r_{D_i} - TTRT - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor TTRT
\]

\[
\geq k \cdot C_i - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor + 1 \frac{TTRT}{P_i} C_i
\]

\[
\iff \left\lfloor k \frac{P_i}{TTRT} \right\rfloor + 1 + \frac{TTRT - r_{D_i}}{q_i} + (k - 1)r_{P_i} + r_{D_i} - TTRT - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor TTRT
\]

\[
\geq \left( k - 1 - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor \right) \frac{TTRT}{P_i} C_i
\]

\[
\iff \left( k \frac{(q_i - 1)TTRT + r_{P_i}}{TTRT} \right) + 1 + \frac{TTRT - r_{D_i}}{q_i}
\]

\[
\geq (k - 1)r_{P_i} + r_{D_i} - TTRT - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor TTRT
\]

\[
\iff \left( k \frac{(q_i - 1)TTRT + r_{P_i}}{TTRT} - 1 - \left\lfloor k \frac{(q_i - 1)TTRT + r_{P_i}}{TTRT} \right\rfloor \right) TTRT C_i
\]

\[
\iff (k(q_i - 1) + \frac{k_{P_i} TTRT}{TTRT} + 1) \frac{TTRT - r_{D_i}}{q_i} + (k - 1)r_{P_i} + r_{D_i} - TTRT - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor TTRT
\]

\[
\geq (k(q_i - 1) + \frac{k_{P_i} TTRT}{TTRT} - 1 - k(q_i - 1) - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor) \frac{TTRT}{P_i} C_i
\]

\[
\iff [k(q_i - 1) + 1 + \frac{k_{P_i} TTRT}{TTRT} TTRT - r_{D_i}] + (k - 1)r_{P_i} + r_{D_i} - TTRT - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor TTRT
\]

\[
\geq (k_{P_i} TTRT - 1 - \left\lfloor k \frac{P_i}{TTRT} \right\rfloor) \frac{TTRT}{P_i} C_i
\]

Based on (G.29), there are two sub-cases to consider.

**Sub-case 1:** \( \left\lfloor k \frac{P_i}{TTRT} \right\rfloor TTRT \leq (k - 1)r_{P_i} \)

In this sub-case, we have

\[
\iff \frac{TTRT}{P_i} C_i + \frac{TTRT - r_{D_i}}{q_i} \geq TTRT - r_{D_i} \quad (\text{Since G.4})
\]

\[
\iff \frac{TTRT}{P_i} C_i \geq TTRT - r_{D_i} - \frac{TTRT - r_{D_i}}{q_i}
\]

\[
\iff \frac{TTRT}{P_i} C_i \geq (q_i - 1) \frac{TTRT - r_{D_i}}{q_i}
\]
Because \( \frac{kr_P}{TTRT} - 1 - \left| \frac{kr_P}{TTRT} \right| < 0 \), in order to show (G.30), it is only need to show

\[
\left( k(q_i - 1) + 1 \right) \frac{TTTRT - r_{D_i}}{q_i} + (k - 1)r_P + r_{D_i} - TTTRT - \left| \frac{kr_P}{TTRT} \right| TTTRT
\]

\[
\geq \left( \frac{kr_P}{TTRT} - 1 - \left| \frac{kr_P}{TTRT} \right| \right) (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i}
\]

\[
\Leftrightarrow k(q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} + TTTRT - r_{D_i} + \left| \frac{kr_P}{TTRT} \right| TTTRT - r_{D_i}
\]

\[
+ (k - 1)r_P + r_{D_i} - TTTRT - \left| \frac{kr_P}{TTRT} \right| TTTRT
\]

\[
\geq \frac{kr_P}{TTRT} (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} - (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} - \left| \frac{kr_P}{TTRT} \right| (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i}
\]

\[
\Leftrightarrow k(q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} + TTTRT - r_{D_i} + \left| \frac{kr_P}{TTRT} \right| TTTRT - r_{D_i}
\]

\[
+ (k - 1)r_P - \left| \frac{kr_P}{TTRT} \right| TTTRT
\]

\[
\geq \frac{kr_P}{TTRT} (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} - \left| \frac{kr_P}{TTRT} \right| (TTTRT - r_{D_i})
\]

\[
\Leftrightarrow k(q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} - \frac{kr_P}{TTRT} (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} + \left| \frac{kr_P}{TTRT} \right| (TTTRT - r_{D_i})
\]

\[
+ (k - 1)r_P - \left| \frac{kr_P}{TTRT} \right| TTTRT
\]

\[
\geq 0
\]

\[
\Leftrightarrow k(1 - \frac{r_P}{TTTRT}) (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} + \left| \frac{kr_P}{TTRT} \right| (TTTRT - r_{D_i})
\]

\[
+[k(1 - \frac{r_P}{TTTRT}) (q_i - 1) \frac{TTTRT - r_{D_i}}{q_i} > 0
\]

and

\[
\left| \frac{kr_P}{TTRT} \right| (TTTRT - r_{D_i}) > 0
\]

Recall that the condition of this sub-case is \( \frac{kr_P}{TTRT} TTTRT \leq (k - 1)r_P \), and we also have \( (k - 1)r_P - \left| \frac{r_P}{TTTRT} \right| TTTRT \geq 0 \), thus the expression (G.31) can be established. Therefore, the deadline constraint can be satisfied.
Sub-case 2: \( \frac{k r_p}{TTRT} |TTRT| > (k - 1)r_P \)

According to (G.29), under this case, we have

\[
H_i \geq \left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i
\]

\[
\Leftrightarrow \frac{TTRT}{P_i} C_i + \frac{TTRT - r_D_i}{q_i} \geq \left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i
\]

(Since (G.4))

\[
\Leftrightarrow \frac{TTRT}{P_i} C_i \geq \left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i - \frac{TTRT - r_D_i}{q_i}
\]

Similarly to in Sub-case 1, because \( \frac{k r_p}{TTRT} - 1 - \left[ \frac{k r_p}{TTRT} \right] < 0 \), in order to establish (G.30), we only need to show

\[
[k(q_i - 1) + 1 + \left[ \frac{k r_P}{TTRT} \right]] \frac{TTRT - r_D_i}{q_i} + (k - 1)r_P + r_D_i - TTRT - \left[ \frac{k r_P}{TTRT} \right] TTRT
\]

\[
\geq (\frac{k r_P}{TTRT} - 1 - \left[ \frac{k r_P}{TTRT} \right]) (\left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i - \frac{TTRT - r_D_i}{q_i})
\]

\[
\Leftrightarrow k(q_i - 1) \frac{TTRT - r_D_i}{q_i} + \left[ \frac{k r_P}{TTRT} \right] \frac{TTRT - r_D_i}{q_i} + \frac{TTRT - r_D_i}{q_i}
\]

\[
- (k - 1)r_P + TTRT - r_D_i - \left[ \frac{k r_P}{TTRT} \right] TTRT
\]

\[
\geq (\frac{k r_P}{TTRT} - 1 - \left[ \frac{k r_P}{TTRT} \right]) (\left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i)
\]

\[
- (\left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i - \left[ \frac{k r_P}{TTRT} \right] TTRT - r_D_i)
\]

\[
\Leftrightarrow k(q_i - 1) \frac{TTRT - r_D_i}{q_i} + \left[ \frac{k r_P}{TTRT} \right] \frac{TTRT - r_D_i}{q_i} + \frac{TTRT - r_D_i}{q_i}
\]

\[
\geq (\frac{k r_P}{TTRT} - 1 - \left[ \frac{k r_P}{TTRT} \right]) (\left[ \frac{k r_P}{TTRT} \right] TTRT - (k - 1)r_P + TTRT - r_D_i)
\]

193
That is
\[
\left(\left\lfloor \frac{kr_P}{TTRT} \right\rfloor - \frac{kr_P}{TTRT}\right)\left(\left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT - (k - 1)r_{P_i} + TTRT - r_{D_i}\right)
\]
\[+ [k(q_i - 1) + \frac{kr_P}{TTRT} TTRT - r_{D_i}] \geq 0 \quad (G.32)
\]

The above analysis shows that under this sub-case, in order to guarantee the deadline constraint, we need to show (G.32) holds under \(\left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT > (k - 1)r_{P_i}\). This is exactly the same requirement as that described at the end of Case 4.

However, (G.32) (i.e., (G.24)) cannot always hold under \(\left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT > (k - 1)r_{P_i}\). The example given when proving Lemma 10 can be still used here to show this conclusion.

Similarly to in Lemma 9, Fig. G.7 shows a different position of \(k \cdot P_i\). Figures G.1 and G.7 together depict all the possible positions of \(k \cdot P_i\). The situation described in Fig. G.7 is not analysed from the beginning since it implies a better case than that in Fig. G.1. Compared with Fig. G.1, there may be more chances to use the allocated synchronous bandwidth to transmit the messages. The detailed explanation is similar to that in Lemma 9 and thus not repeated here.

Theorem 10 follows from the above discussion.
Appendix H

Deducing $\Delta$ in Eq. (4.42)

Theorem 10 actually refers to two cases from the further study of Lemma 10 given in Appendix F. They are:

- Case 4: fully use the second last $H_i$ but cannot use the last $H_i$
- Sub-case 2 of Case 5: use a portion of the last $H_i$.

Only these two cases need to consider when calculating $\Delta$. Only node $i$’s local information needs to be taken into account, following the discussion in Section 4.2.2.

The following lemma is needed before continuing our discussion on calculating $\Delta$.

**Lemma 11.** The minimum value of $\left\lfloor \frac{kP_i}{TTRT} \right\rfloor$, where $P_i = (q_i - 1)TTRT + r_{P_i}$ and $k \geq 1$, is achieved when $k = 1$, i.e., $\frac{P_i}{TTRT} = \frac{q_i - 1}{TTRT}$.

**Proof.** When $k = 1$, we have

$$\frac{kP_i}{TTRT} = \frac{P_i}{TTRT}$$

Assume that when $k > 1$, we have

$$\frac{kP_i}{TTRT} < \frac{P_i}{TTRT}$$

$$\Leftrightarrow \left\lfloor \frac{kP_i}{TTRT} \right\rfloor < \left\lfloor \frac{P_i}{TTRT} \right\rfloor$$

$$\Leftrightarrow \left\lfloor \frac{k[(q_i - 1)TTRT + r_{P_i}]}{TTRT} \right\rfloor < \left\lfloor \frac{(q_i - 1)TTRT + r_{P_i}}{TTRT} \right\rfloor$$

$$\Leftrightarrow k(q_i - 1) + \left\lfloor \frac{kr_{P_i}}{TTRT} \right\rfloor < k(q_i - 1) + \left\lfloor \frac{r_{P_i}}{TTRT} \right\rfloor$$

$$\Leftrightarrow \left\lfloor \frac{kr_{P_i}}{TTRT} \right\rfloor < 0$$

(H.1)
Because (H.1) cannot hold for any \( k > 1 \), the above assumption is not true, which implies that when \( k > 1 \), we have

\[
\left\lfloor \frac{kP_i}{TTTRT} \right\rfloor \geq \left\lfloor \frac{P_i}{TTTRT} \right\rfloor
\]

Thus, the minimum value can be achieved when \( k = 1 \), which is \( \left\lfloor \frac{P_i}{TTTRT} \right\rfloor \).

Now we are going to calculate the value of \( \Delta \), which can be expressed as

\[
\Delta = f(C_i) + g
\]

where \( f(C_i) \) is a function of \( C_i \) and \( g \) is not a function of \( C_i \). Since function \( f(C_i) \) could be in any form, in order to simplify the discussion, we consider only two possible forms here\(^1\):

\( f(C_i) = 0 \) and \( f(C_i) = aC_i + b \).

**Form 1:** \( f(C_i) = 0 \)

If \( f(C_i) = 0 \), under Case 4, in order to meet the deadline constraint, from (G.22), we have

\[
x_k^i = \left( \left\lfloor \frac{kP_i}{TTTRT} \right\rfloor \right) \cdot H_i \geq k \cdot C_i
\]

\[
\Leftrightarrow \left( \frac{kP_i}{TTTRT} \right) \cdot \left( \frac{q_iTTRT}{q_i - 1} C_i - \Delta \right) \geq k \cdot C_i \quad \text{(By (4.42))}
\]

\[
\Leftrightarrow \Delta \leq \frac{q_iTTRT}{q_i - 1} - \frac{kC_i}{TTTRT} - \frac{kP_i}{TTTRT}
\]

\[
\Leftrightarrow \Delta \leq \frac{q_iTTRT}{q_i - 1} - \frac{k}{TTTRT} C_i \quad \text{(H.2)}
\]

Also,

\[
D_i - (q_i + 1)TTTRT + H_i > 0
\]

\[
\Leftrightarrow D_i - (q_i + 1)TTTRT + \left( \frac{q_iTTRT}{q_i - 1} C_i \right) - \Delta > 0
\]

\[
\Leftrightarrow \left( \frac{q_iTTRT}{q_i - 1} C_i \right) > (q_i + 1)TTTRT - D_i + \Delta
\]

\[
\Leftrightarrow C_i > \frac{(q_i + 1)TTTRT - D_i + \Delta}{\frac{q_iTTRT}{q_i - 1}} \quad \text{(H.3)}
\]

\(^1\)It could be difficult to consider every possible form of \( f(C_i) \). Because polynomials can be used in a wide range of problems and they have a simple form, we only consider \( f(C_i) \) as a polynomial function. Furthermore, in all the previous results, the degree of \( C_i \) is always no more than one. Thus, we only consider those \( f(C_i) \)s whose degrees are no more than one.
Because
\[
\frac{q_iTTRT}{P_i} \left[ \frac{kP_i}{TTRT} \right] - k(q_i - 1) = \frac{q_iTTRT}{P_i} \left[ \frac{kP_i}{TTRT} \right] - k(q_i - 1)P_i
\]
\[
= \frac{q_iTTRT}{P_i} \left[ \frac{k(q_i - 1)TTRT + rP_i}{TTRT} \right] - k(q_i - 1)\left[(q_i - 1)TTRT + rP_i\right]
\]
\[
= \left[ \frac{kP_i}{TTRT} \right] q_iTTRT + k(q_i - 1)\left[q_iTTRT - (q_i - 1)TTRT - rP_i\right]
\]
\[
= \frac{kP_i}{TTRT} \left[q_iTTRT + k(q_i - 1)\left[q_iTTRT - (q_i - 1)TTRT - rP_i\right]\right]
\]
\[
> 0 \quad \text{(Since } q_i \geq 2 \text{ and } TTRT > rP_i \text{)}
\]

we have
\[
\left( \frac{q_iTTRT}{P_i} \frac{k}{q_i - 1} - \frac{kP_i}{TTRT} \right)C_i = \left[ \frac{kP_i}{TTRT} \right] q_iTTRT + k(q_i - 1)\left[q_iTTRT - (q_i - 1)TTRT - rP_i\right]
\]
\[
> 0 \quad \text{(Since } q_i \geq 2 \text{)} \quad \text{(H.4)}
\]

From (H.2), (H.3) and (H.4), we have
\[
\Delta \leq \left( \frac{q_iTTRT}{P_i} \frac{k}{q_i - 1} - \frac{kP_i}{TTRT} \right) (q_i + 1)TTRT - D_i + \Delta
\]
\[
\Leftrightarrow \Delta \leq \left( 1 - \frac{k(q_i - 1)}{\left[ \frac{kP_i}{TTRT} \right] q_iTTRT} \right) [(q_i + 1)TTRT - D_i + \Delta]
\]
\[
\Leftrightarrow \Delta \leq \frac{k(q_i - 1)}{\left[ \frac{kP_i}{TTRT} \right] q_iTTRT} \left[(q_i + 1)TTRT - D_i\right]
\]
\[
\Leftrightarrow \Delta \leq \frac{k(q_i - 1)}{\left[ \frac{kP_i}{TTRT} \right] q_iTTRT} \left[(q_i + 1)TTRT - D_i\right]
\]
\[
\Leftrightarrow \Delta \leq \left( \frac{\left[ \frac{kP_i}{TTRT} \right]}{k(q_i - 1)} \right) - 1 \left[(q_i + 1)TTRT - D_i\right]
\]
\[
\Leftrightarrow \Delta \leq \left( \frac{\left[ \frac{kP_i}{TTRT} \right]}{k(q_i - 1)} \right) \cdot \frac{q_i}{q_i - 1} \left[(q_i + 1)TTRT - D_i\right] \quad \text{(H.5)}
\]

The possible maximum value of \( \Delta \) must be no larger than the minimum value of
\[
\left( \frac{\left[ \frac{kP_i}{TTRT} \right]}{k(q_i - 1)} \right) \cdot \frac{q_i}{q_i - 1} \left[(q_i + 1)TTRT - D_i\right]
\]

where \( k \geq 1 \).
To calculate this minimum value, we only need to find out the minimum value of \( \frac{k P_i}{TTRT} \). According to Lemma 11, the minimum value is
\[
\frac{P_i}{TTRT} = q_i - 1
\]
Thus
\[
\Delta = \frac{q_i - 1}{q_i - 1} \cdot q_i \cdot (q_i + 1)/TTRT - D_i
\]
\[
= \left( \frac{q_i}{TTRT} P_i - 1 \right) \cdot (q_i + 1)/TTRT - D_i
\]
While under Sub-case 2 of Case 5, in order to satisfy deadline constraints, we have
\[
x_i = (k P_i/TTRT - 1 + 1)H_i + (k - 1)P_i + D_i - \left(\left(\frac{k P_i}{TTRT}\right) + 2\right)/TTRT - H_i
\]
\[
= \left(\left(\frac{k P_i}{TTRT}\right) + 1\right)H_i + (k - 1)P_i + D_i - 2 \cdot TTRT - \left(\frac{k P_i}{TTRT}\right)TTRT \geq kC_i
\]
That is,
\[
\left(\left(\frac{k P_i}{TTRT}\right) + 1\right)\left(\frac{q_i/TTRT C_i}{q_i - 1} - \Delta\right) + (k - 1)P_i + D_i - 2 \cdot TTRT - \left(\frac{k P_i}{TTRT}\right)TTRT \geq kC_i
\]
Thus,
\[
\Delta \leq \frac{q_i/TTRT C_i}{q_i - 1} - kC_i - \left(\left(\frac{k P_i}{TTRT}\right) + 1\right)H_i + (k - 1)P_i + D_i - 2 \cdot TTRT - \left(\frac{k P_i}{TTRT}\right)TTRT
\]
\[
= \left(\frac{q_i/TTRT}{q_i - 1} - \frac{k}{\left(\frac{k P_i}{TTRT}\right) + 1}\right)C_i + \left(\left(\frac{k P_i}{TTRT}\right) + 1\right)H_i + (k - 1)P_i + D_i - 2 \cdot TTRT - \left(\frac{k P_i}{TTRT}\right)TTRT
\]
(H.6)
Because \( \frac{k}{\left(\frac{k P_i}{TTRT}\right) + 1} < \frac{k}{\frac{k P_i}{TTRT}} \), from (H.4), we have
\[
\frac{q_i/TTRT}{q_i - 1} > \frac{k}{\left(\frac{k P_i}{TTRT}\right) + 1}
\]
(H.7)
Also, under this case, we have
\[
H_i \geq \left(\frac{k P_i}{TTRT}\right)TTRT - (k - 1)rP_i + TTRT - rD_i
\]
\[
\Leftrightarrow \frac{q_i/TTRT C_i}{q_i - 1} - \Delta \geq \left(\frac{k P_i}{TTRT}\right)TTRT - (k - 1)rP_i + TTRT - rD_i
\]
\[
\Leftrightarrow C_i \geq \left(\frac{k P_i}{TTRT}\right)TTRT - (k - 1)rP_i + TTRT - rD_i + \Delta
\]
(H.8)
From (H.6), (H.7) and (H.8), we have

\[
\Delta \leq \left( \frac{q_{i} TTRT}{P} - \frac{k}{q_{i} - 1} \right) \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + TTRT - r D_i + \Delta \right) + (k - 1)P_i + D_i - \frac{k P}{TTRT} TTRT \]

\[
\Rightarrow \Delta \leq \left( 1 - \frac{k}{(\frac{k P}{TTRT}) + 1} \right) \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + TTRT - r D_i + \Delta \right) + (k - 1)((q_i - 1)TTRT + r P) + q_i TTRT + r D_i - 2 \cdot TTRT \]

\[
\Rightarrow \Delta \leq \left( 1 - \frac{k}{(\frac{k P}{TTRT}) + 1} \right) \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + TTRT - r D_i \right) + \frac{k}{(\frac{k P}{TTRT}) + 1} q_{i - 1} \]

\[
\Rightarrow \Delta \leq \left( \left( \frac{k P}{TTRT} \right) q_{i - 1} - 1 \right) \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + TTRT - r D_i \right) + \left( \left( \frac{k P}{TTRT} \right) q_{i - 1} - 1 \right) \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + TTRT + r D_i \right) \]

\[
\Rightarrow \Delta \leq \left( \left( \frac{k P}{TTRT} \right) q_{i - 1} - 1 \right) \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + (q_i + 1)TTRT - D_i \right) \quad (H.9)
\]

Recalling the condition of \( \left( \frac{k P}{TTRT} \right) TTRT > (k - 1)r P \), and also because of \( \frac{k P}{q_{i - 1}} > 1 \), we have

\[
\left( \frac{k P}{q_{i - 1}} \right) q_{i - 1} - 1 \left( \left( \frac{k r P}{TTRT} \right) TTRT - (k - 1)r P + (q_i + 1)TTRT - D_i \right) \geq \left( \frac{k P}{q_{i - 1}} \right) q_{i - 1} - 1 \left( (q_i + 1)TTRT - D_i \right)
\]
Thus, the maximum value of $\Delta$ which is calculated from (H.5) must satisfy the inequality (H.9). It is neither necessary nor right to deduce the minimum value of $\Delta$ from (H.9) because a value such calculated may not satisfy (H.5).

From the above discussion, the value of $\Delta$ can be worked out when $\Delta$ dose not contain any information of $C_i$, i.e., $f(C_i) = 0$, which is

$$\Delta = \left(\frac{q_iTTRT}{P_i} - 1\right)[(q_i + 1)TTRT - D_i] \quad (H.10)$$

**From 2: $f(C_i) = aC_i + b$**

Similarly to Section H, if $f(C_i) = aC_i + b$, under Case 4, in order to meet the deadline constraint, from (H.2), we have

$$\Delta \leq \left(1 - \frac{k}{\left\lfloor \frac{kP_i}{TTRT} \right\rfloor}\right)\frac{q_iTTRT}{P_i} - 1 \quad (H.11)$$

From Lemma 11, we have

$$\frac{\left\lfloor \frac{kP_i}{TTRT} \right\rfloor}{\frac{kP_i}{TTRT}} = \frac{q_i - 1}{P_iTTRT} \quad \Leftrightarrow \quad \frac{\left\lfloor \frac{kP_i}{TTRT} \right\rfloor}{\frac{kP_i}{TTRT}} \leq \frac{P_i}{TTRT} \frac{q_i - 1}{P_iTTRT}$$

Thus,

$$\left(1 - \frac{\left\lfloor \frac{kP_i}{TTRT} \right\rfloor}{\frac{kP_i}{TTRT}}\right)\frac{q_iTTRT}{P_i} - 1 \leq \left(1 - \frac{P_i}{TTRT} \frac{q_i - 1}{q_iTTRT}\right)\frac{q_iTTRT}{P_i} - 1 \quad C_i$$

$$\quad = \left(1 - \frac{P_i}{q_iTTRT}\right)\frac{q_iTTRT}{q_iTTRT} - 1 \quad C_i$$

$$\quad = \left(\frac{q_iTTRT}{P_i(q_i - 1)}\right)\frac{q_iTTRT}{P_i(q_i - 1)} \quad C_i$$

This leads to the maximum value of $\Delta$, as expressed below:

$$\Delta = \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)} \quad (H.12)$$

Instead of calculating the value of $\Delta$ under Sub-case 2 of Case 5, we are going to check if the value of $\Delta$ defined by (H.12) can guarantee the deadline constraint in this case. Refer to the description in the last section, in order to show that the deadline constraint can be satisfied, we only need to show

$$\left(\left\lfloor \frac{kP_i}{TTRT} \right\rfloor + 1\right)\left(\frac{q_iTTRT}{P_i(q_i - 1)}C_i - \Delta\right) + (k - 1)P_i + D_i - 2 \cdot TTRT - \left\lfloor \frac{kP_i}{TTRT} \right\rfloor TTRT \geq kC_i$$

200
According to (H.12),
\[
\frac{q_i^{TTRT} C_i}{q_i - 1} - \Delta = \frac{q_i^{TTRT} C_i}{q_i - 1} - \left( \frac{q_i^{TTRT} - P_i}{P_i(q_i - 1)} \right) C_i
\]
\[
= \frac{C_i}{q_i - 1}
\]

Thus, we only need to show
\[
\left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor + 1 \right) \frac{C_i}{q_i - 1} + (k - 1) P_i + D_i - 2 \cdot TTRT - \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT
\]
\[
\geq kC_i
\]
\[
\iff (k - 1) P_i + D_i - 2 \cdot TTRT - \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT
\]
\[
\geq [k(q_i - 1) - \left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor + 1 \right) C_i]
\]
\[
\iff (k - 1)\left( (q_i - 1) TTRT + r_P \right) + q_i TTRT + r_D - 2 \cdot TTRT
\]
\[
- \left\lfloor k[(q_i - 1) TTRT + r_P] \right\rfloor TTRT
\]
\[
\geq [k(q_i - 1) - \left( \left\lfloor (q_i - 1) TTRT + r_P \right\rfloor + 1 \right) C_i]
\]
\[
\iff (k - 1) r_P + r_D - TTRT - \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT
\]
\[
\geq - \left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor + 1 \right) \frac{C_i}{q_i - 1}
\]

Because of \(- \left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor + 1 \right) < 0\) and recalling the condition here
\[
H_i \geq \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT - (k - 1) r_P + TTRT - r_D,
\]
\[
\iff \frac{C_i}{q_i - 1} \geq \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT - (k - 1) r_P + TTRT - r_D,
\]
in order to show the above inequality, we only need to show
\[
(k - 1) r_P + r_D - TTRT - \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT
\]
\[
\geq - \left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor \right) \left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT - (k - 1) r_P + TTRT - r_D \right)
\]
That is
\[
\left\lfloor \frac{kr_P}{TTRT} \right\rfloor \left( \left\lfloor \frac{kr_P}{TTRT} \right\rfloor TTRT - (k - 1) r_P + TTRT - r_D \right) \geq 0
\]

It is obvious that the above inequality can hold. Thus, the value of \(\Delta\) defined in (H.12) can be used in this case to meet the deadline constraint.

Because we want a maximum value of \(\Delta\), combining (H.10) and (H.12), we have
\[
\Delta = \max\left\{ \left( \frac{q_i TTRT}{P_i} - 1 \right) \left[ (q_i + 1) TTRT - D_i \right], \frac{(q_i TTRT - P_i) C_i}{P_i(q_i - 1)} \right\}
\]
\[
\text{(H.13)}
\]
The final result of $\Delta$

We will further analyse (H.13) in the following discussion to simply the value of $\Delta$. Recalling that $H_i$ should be

$$H_i = \frac{\frac{q_iTTRT}{P_i}C_i}{q_i - 1} - \Delta \quad \text{(H.14)}$$

according to the “max” part in (H.13), two cases need to be considered here.

**Case 1:** if $\frac{q_iTTRT}{P_i} - 1 \geq \frac{(q_i + 1)TTRT - D_i}{P_i(q_i - 1)}$

Under this case, we have

$$H_i = \frac{\frac{q_iTTRT}{P_i}C_i}{q_i - 1} - \Delta$$

$$= \frac{\frac{q_iTTRT}{P_i}C_i}{q_i - 1} - \frac{(q_iTTRT - 1)((q_i + 1)TTRT - D_i)}{P_i(q_i - 1)}$$

Since our discussion about $\Delta$ is based on the condition of $D_i - (q_i + 1)TTRT + H_i > 0$, we have

$$D_i - (q_i + 1)TTRT + H_i > 0$$

$$\Leftrightarrow D_i - (q_i + 1)TTRT + \frac{\frac{q_iTTRT}{P_i}C_i}{q_i - 1} - \frac{(q_iTTRT - 1)((q_i + 1)TTRT - D_i)}{P_i(q_i - 1)} > 0$$

$$\Leftrightarrow D_i - (q_i + 1)TTRT + \frac{\frac{q_iTTRT}{P_i}C_i}{q_i - 1} - \frac{q_iTTRT}{P_i}((q_i + 1)TTRT - D_i) + (\frac{q_iTTRT}{P_i} - 1)((q_i + 1)TTRT - D_i) > 0$$

$$\Leftrightarrow \frac{\frac{q_iTTRT}{P_i}C_i}{q_i - 1} - \frac{q_iTTRT}{P_i}((q_i + 1)TTRT - D_i) > 0$$

$$\Leftrightarrow \frac{q_iTTRT}{P_i} - \frac{C_i}{q_i - 1} - ((q_i + 1)TTRT - D_i) > 0$$

$$\Leftrightarrow \frac{C_i}{q_i - 1} - ((q_i + 1)TTRT - D_i) > 0$$

$$\Leftrightarrow \frac{C_i}{q_i - 1} - (q_i + 1)TTRT + D_i > 0$$

(\text{H.15})

Because the condition for this case can be equivalently transformed to

$$\frac{q_iTTRT}{P_i} - 1 \geq \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)}$$

$$\Leftrightarrow \frac{q_iTTRT - P_i}{P_i}((q_i + 1)TTRT - D_i) \geq \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)}$$

$$\Leftrightarrow (q_i + 1)TTRT - D_i \geq \frac{C_i}{q_i - 1}$$

(Since $P_i = (q_i - 1)TTRT + r_i < q_iTTRT$)

$$\Leftrightarrow (q_i + 1)TTRT - D_i - \frac{C_i}{q_i - 1} \geq 0$$

$$\Leftrightarrow \frac{C_i}{q_i - 1} - (q_i + 1)TTRT + D_i \leq 0,$$
which validates (H.15).

This implies that the case discussed here is impossible.

**Case 2:** if \( \frac{q_{TTRT}}{P_i} - 1 \)\( [(q_i + 1)TTRT - D_i] < \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)} \)

Under this case, we have

\[
H_i = \frac{q_{TTRT}C_i}{P_i - 1} - \Delta \\
= \frac{q_iTTRT - P_i)C_i}{P_i(q_i - 1)} \\
= \frac{q_iTTRT - (q_iTTRT - P_i)C_i}{P_i(q_i - 1)} \\
= \frac{C_i}{q_i - 1} \quad \text{(H.16)}
\]

Thus, from the condition of this case, we have

\[
\left( \frac{q_{TTRT}}{P_i} - 1 \right)\left( (q_i + 1)TTRT - D_i \right) < \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)} \\
\Leftrightarrow \frac{C_i}{q_i - 1} - (q_i + 1)TTRT + D_i > 0 \quad \text{(Similar as above)} \\
\Leftrightarrow D_i - (q_i + 1)TTRT + H_i > 0 \quad \text{(Since (H.16))}
\]

The condition of \( D_i - (q_i + 1)TTRT + H_i > 0 \), under which \( \Delta \) is calculated, still holds for this case.

Based on the above discussion, the value of \( \Delta \) is

\[
\Delta = \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)}
\]

That is, the “max” part of \( \Delta \) in (H.13) must be always \( \frac{(q_iTTRT - P_i)C_i}{P_i(q_i - 1)} \) under the condition of \( D_i - (q_i + 1)TTRT + H_i > 0 \).
Appendix I

Proof of Theorem 12

**Theorem 12.** Let the synchronous bandwidth be allocated as in Eq. (4.47). Let $b_i$ be the maximum number of bytes in each synchronous message from stream $S_i$. If the deadline constraint is satisfied, then the buffer needed at node $i$ is no more than

$$b_i \cdot \max\{\lceil \frac{2 \cdot TTRT}{P_i} + 1 \rceil, 3\}$$ bytes.

To prove this theorem, we need the following lemma.

**Lemma 12.** Let the synchronous bandwidth be allocated as in Eq. (4.47). Let $w_{i,j}$ be an upper bound on the waiting time of the $j$-th message in stream $S_i$ with $D_i > P_i$. Then under the deadline constraint, $w_{i,j}$ is bounded by

$$w_{i,j} = s_{i,j} - t_{i,j} \leq \min(D_i, P_i + 2TTRT) \quad \text{(I.1)}$$

where $s_{i,j}, t_{i,j}$ are the transmission completion time and the arrival time of the message, respectively.

**Proof.** There are three cases to consider.

**Case 1: When** $TTRT < D_i < 2 \cdot TTRT$

Because the deadline constraint is satisfied, $w_{i,j}$ must be no more than $D_i$, we have:

$$w_{i,j} \leq D_i.$$

Considering the min part of (I.1), because

$$D_i < 2 \cdot TTRT < P_i + 2 \cdot TTRT,$$
we have
\[ \min(D_i, P_i + 2 \cdot TT \cdot RT) = D_i. \]

Thus (I.1) can hold for this case.

**Case 2: When** \( 2 \cdot TT \cdot RT \leq D_i \leq P_i + 2 \cdot TT \cdot RT \)

Because the deadline constraint is satisfied, with a similar analysis to Case 1, (I.1) must hold.

**Case 3: When** \( D_i > P_i + 2 \cdot TT \cdot RT \)

For this case, consider an arbitrary busy interval \([t_0, t_1]\). For (I.1) to hold, each message in stream \( S_i \) must be sent within \( P_i + 2 \cdot TT \cdot RT \) units of time since its arrival. Therefore, the \( k \)-th message in the busy interval must be sent by \( t_0 + (k - 1)P_i + P_i + 2 \cdot TT \cdot RT \). This means that between \( t_0 \) and \( t_0 + (k - 1)P_i + P_i + 2 \cdot TT \cdot RT \), at least \( k \cdot C_i \) messages must be sent. By Corollary 2, during the time interval of \([t_0, t_0 + (k - 1)P_i + P_i + 2 \cdot TT \cdot RT]\) (i.e., the length of this time interval is \( kP_i + 2TT \cdot RT \)), the minimum available time for node \( i \) to transmit synchronous messages is
\[
x_i(kP_i + 2TT \cdot RT) = \left(\left\lfloor \frac{kP_i + 2 TT \cdot RT}{TT \cdot RT} \right\rfloor - 1 \right) \cdot H_i \\
+ \max[0, kP_i + 2 TT \cdot RT + H_i - \left(\left\lfloor \frac{kP_i + 2 TT \cdot RT}{TT \cdot RT} \right\rfloor + 1\right) TT \cdot RT] \\
\geq \left(\left\lfloor \frac{kP_i + 2 TT \cdot RT}{TT \cdot RT} \right\rfloor - 1 \right) \cdot H_i. \tag{I.2}
\]

For the \( k \)-th message in the busy interval to be sent by \( t_0 + (k - 1)P_i + P_i + 2 \cdot TT \cdot RT \), the following expression must hold
\[
x_i(kP_i + 2TT \cdot RT) \geq k \cdot C_i.
\]

To satisfy this condition, from (I.2), we only need to show
\[
\left(\left\lfloor \frac{kP_i + 2 TT \cdot RT}{TT \cdot RT} \right\rfloor - 1 \right) \cdot H_i \geq k \cdot C_i. \tag{I.3}
\]

Because \( D_i > P_i + 2 \cdot TT \cdot RT \), we have
\[
D_i \geq 2 \cdot TT \cdot RT
\]

and
\[
q_i = \left\lceil \frac{D_i}{TT \cdot RT} \right\rceil \geq \left\lceil \frac{P_i + 2 \cdot TT \cdot RT}{TT \cdot RT} \right\rceil = \left\lceil \frac{P_i}{TT \cdot RT} \right\rceil + 2 > \left\lceil \frac{P_i}{TT \cdot RT} \right\rceil + 1
\]

and
\[
\frac{q_i \cdot TT \cdot RT}{P_i} > \frac{D_i - rD_i}{D_i - 2 \cdot TT \cdot RT} > 1. \tag{I.4}
\]
From (4.47), we have
\[ H_i = \frac{a_{TT RT} C_i}{q_i} - \frac{1}{q_i} \max\{D_i - (q_i + 1) TT RT + \frac{a_{TT RT} C_i}{q_i - 1}, 0\} \]

If \( D_i - (q_i + 1) TT RT + \frac{a_{TT RT} C_i}{q_i - 1} \leq 0 \)
\[
\left(\left\lfloor \frac{k_p + 2 TT RT}{TT RT} \right\rfloor - 1\right) \cdot H_i
\]
\[
= \left(\left\lfloor \frac{k_p + 2 TT RT}{TT RT} \right\rfloor - 1\right) \cdot \frac{a_{TT RT} C_i}{q_i - 1}
\]
\[
= \left(\left\lfloor \frac{k_p}{TT RT} \right\rfloor + 1\right) \cdot \frac{a_{TT RT} C_i}{q_i - 1}
\]
\[
> (\left\lfloor \frac{k_p}{TT RT} \right\rfloor) \cdot \frac{a_{TT RT} C_i}{q_i - 1}
\]
\[
= k \cdot \frac{q_i}{q_i - 1} C_i
\]
\[
> k C_i
\]

If \( D_i - (q_i + 1) TT RT + \frac{a_{TT RT} C_i}{q_i - 1} > 0 \)
\[
\left(\left\lfloor \frac{k_p + 2 TT RT}{TT RT} \right\rfloor - 1\right) \cdot H_i
\]
\[
= \left(\left\lfloor \frac{k_p + 2 TT RT}{TT RT} \right\rfloor - 1\right) \cdot \left[\frac{a_{TT RT} C_i}{q_i} - \frac{1}{q_i} \left(D_i - (q_i + 1) TT RT + \frac{a_{TT RT} C_i}{q_i - 1}\right)\right]
\]
\[
= \left(\left\lfloor \frac{k_p}{TT RT} \right\rfloor + 1\right) \cdot \left[\frac{TT RT}{P_i} C_i + \frac{(q_i + 1) TT RT - D_i}{q_i}\right]
\]
\[
> (\left\lfloor \frac{k_p}{TT RT} \right\rfloor + 1) \cdot \frac{TT RT}{P_i} C_i
\]
\[
> k \cdot \frac{q_i}{P_i} TT RT C_i
\]
\[
= k C_i
\]

Thus, (I.3) can be established. This means that (I.1) can hold under this case.

Lemma 12 follows from the above discussion. \qed

Now we are going to prove Theorem 12, which gives the bound on the maximum queue size.

**Theorem 12.** Let the synchronous bandwidth be allocated as in Eq. (4.47). Let \( b_i \) be the maximum number of bytes in each synchronous message from stream \( S_i \). If the deadline constraint is satisfied, then the buffer needed at node \( i \) is no more than
\[
b_i \cdot \max\{\left\lfloor \frac{2 \cdot TT RT}{P_i} + 1\right\rfloor, 3\}
\]
bytes.
Proof. It is easy to check that when $D_i \leq P_i$, at any time, there is at most one synchronous message waiting in the queue at node $i$. Thus, the buffer needed is no more than $b_i \cdot 1 = b_i$.

When $D_i > P_i$, according to the $\min$ part of (I.1), there are two cases to consider.

**Case 1: When $D_i \geq P_i + 2 \cdot TTRT$**

By (I.1), the waiting time of the $j$-th message in stream $S_i$ satisfies

$$w_{i,j} \leq \min(D_i, P_i + 2TTRT) = P_i + 2TTRT.$$ 

This means that the $j$-th message in $S_i$, which arrives at time $t_{i,j}$, will be sent by $t_{i,j} + P_i + 2TTRT$. This is true for all possible value of $j > 0$. Thus, the maximum number of messages that can be queued at node $i$ at any time is the maximum number of messages that can arrive in an interval of length $P_i + 2TTRT$, i.e., $\left\lceil \frac{P_i + 2TTRT}{P_i} \right\rceil = \left\lceil \frac{2TTRT}{P_i} + 1 \right\rceil$. So, the buffer needed is no more than $b_i \cdot \left\lceil \frac{2TTRT}{P_i} + 1 \right\rceil$ bytes.

**Case 2: When $D_i < P_i + 2 \cdot TTRT$**

By (I.1), the waiting time of the $j$-th message in stream $S_i$ satisfies

$$w_{i,j} \leq \min(D_i, P_i + 2TTRT) = D_i$$

With a similar analysis, the buffer needed is upper bounded by

$$b_i \cdot \left\lceil \frac{D_i}{P_i} \right\rceil \leq b_i \cdot \left\lceil \frac{P_i + 2TTRT}{P_i} \right\rceil = b_i \cdot \left\lceil \frac{2 \cdot TTRT}{P_i} + 1 \right\rceil.$$

Thus, the buffer needed when $D_i > P_i$ is no more than $b_i \cdot \left\lceil \frac{2TTRT}{P_i} + 1 \right\rceil$ bytes. Moreover, if $TTRT \leq P_i$, we have $b_i \cdot \left\lceil \frac{2TTRT}{P_i} + 1 \right\rceil = b_i \cdot \left\lceil \frac{2 \cdot TTRT}{P_i} + 1 \right\rceil = b_i \cdot 3$. While $TTRT > P_i$, $b_i \cdot \left\lceil \frac{2TTRT}{P_i} + 1 \right\rceil$ must be no less than $b_i \cdot 3$.

It is concluded, from the above discussion, that the buffer needed at node $i$ is no more than $b_i \cdot \max\{\left\lceil \frac{2TTRT}{P_i} + 1 \right\rceil, 3\}$ bytes. \(\square\)
Appendix J

Proof of Lemma 4

Lemma 4. When $D_{\text{min}} \geq 2 \cdot TTRT$ and Eq. (4.47) is to be used for allocating synchronous bandwidths, the maximum value of the worst case achievable utilisation

$$U^*_e = \frac{q_{\text{min}} - 1}{q_{\text{min}} + 1 - \frac{1 - \alpha}{n}} (1 - \alpha) = \frac{\left\lfloor \frac{D_{\text{min}}}{TTRT} \right\rfloor - 1}{\left\lfloor \frac{D_{\text{min}}}{TTRT} \right\rfloor + 1 - \frac{1 - \tau}{n}} (1 - \frac{\tau}{TTRT})$$

occurs when

$$D_{\text{min}} = \frac{-n + 2 + \frac{\tau}{D_{\text{min}}} + \sqrt{9n^2 + 8n^2 \frac{D_{\text{min}}}{\tau} + 6n \frac{\tau}{D_{\text{min}}} + \frac{\tau^2}{D_{\text{min}}} - 4n \frac{D_{\text{min}}}{\tau}}}{2n + 2 \frac{\tau}{D_{\text{min}}}} = a$$

or

$$D_{\text{min}} = \frac{-3n + 2 - \frac{\tau}{D_{\text{min}}} + \sqrt{9n^2 + 8n^2 \frac{D_{\text{min}}}{\tau} + 6n \frac{\tau}{D_{\text{min}}} + \frac{\tau^2}{D_{\text{min}}} - 4n \frac{D_{\text{min}}}{\tau}}}{2n + 2 \frac{\tau}{D_{\text{min}}}} = b$$

where $0 \leq a - b \leq 1$ and $a \geq 2$.

Proof. First, we show that $U^*_e$ is maximised when $\frac{D_{\text{min}}}{TTRT}$ is an integer by contradiction. Suppose that $U^*_e$ is maximised by a particular value of $TTRT$, say $TTRT'$, such that $\frac{D_{\text{min}}}{TTRT'}$ is not an integer. Thus

$$U^*_e(TTRT') \geq U^*_e(TTRT) \quad (J.1)$$

Let $q'_{\text{min}} = \left\lfloor \frac{D_{\text{min}}}{TTRT'} \right\rfloor$ and $r'_{\text{min}} = D_{\text{min}} - q'_{\text{min}} TTRT' > 0$, we can define $TTRT''$ to be

$$TTRT'' = TTRT' + \frac{r'_{\text{min}}}{q'_{\text{min}}} = TTRT' + \frac{D_{\text{min}} - q'_{\text{min}} TTRT'}{q'_{\text{min}}} = \frac{D_{\text{min}}}{q'_{\text{min}}}$$

208
Because \( \frac{D_{\min}}{TTRT'} = q'_{\min} \) and \( q'_{\min} \) is an integer, \( \frac{D_{\min}}{TTRT'} \) must be an integer. Therefore, we have

\[
U^*_e(TTTR^T') = \frac{[\frac{D_{\min}}{TTRT'}] - 1}{[\frac{D_{\min}}{TTRT'}] + 1 - \frac{\tau}{TTRT'}}(1 - \frac{\tau}{TTRT'})
\]

\[
= \frac{q'_{\min} - 1}{q'_{\min} + 1 - \frac{\tau}{TTRT'}}(1 - \frac{\tau}{TTRT'})
\]

This contradicts (J.1). Thus, the maximum value of the worst case achievable utilisation occurs when \( \frac{D_{\min}}{TTRT} \) is an integer.

Let this integer be \( m \), i.e., \( m = \frac{D_{\min}}{TTRT} \geq 2 \), we have

\[
U^*_e = f(m) = \frac{m - 1}{m + 1 - \frac{\tau}{q'_{\min}}}(1 - \frac{m\tau}{D_{\min}})
\]  

(J.2)

First, we consider a function \( f_1(x) \)

\[
f_1(x) = \frac{x - 1}{x + 1 - \frac{\tau}{q'_{\min}}}(1 - \frac{x\tau}{D_{\min}})
\]

when \( x > 0 \).

Because the first derivative of \( f_1(x) \) with respect to \( x \) is

\[
f'_1(x) = \frac{[-(n\tau D_{\min} + \tau^2)x^2 + (2\tau D_{\min} - 2n\tau D_{\min})x + 2nD^2_{\min} - D_{\min}^2 + n\tau D_{\min}n]}{x(nD_{\min} + nD_{\min} - D_{\min} + x\tau)^2}
\]

It is easy to check that the function of \( f'_1(x) = 0 \) has one solution when \( x > 0 \). Thus, \( f_1(x) \) has one extreme value point. Furthermore, because the second derivative of \( f_1(x) \) with respect to \( x \) is

\[
f''_1(x) = -\frac{2n^2D_{\min}(\tau + D_{\min})(\tau + 2nD_{\min} - D_{\min})}{(x(nD_{\min} + nD_{\min} - D_{\min} + x\tau)^3}
\]

we have \( f''_1(x) < 0 \) when \( x > 0 \). Thus, with \( f'_1(x) \) and \( f''_1(x) \), \( f_1(x) \) is a concave.

Compared function \( f(m) \) with function \( f_1(x) \), \( f(m) \) is identical to \( f_1(x) \) except that \( f(m) \) is defined on an integer domain \( (m = 1, 2, \cdots) \) and \( f_1(x) \) is defined on a real domain \( (x > 0) \). Because \( f_1(x) \) is concave, \( f(m) \) will be monotonically increasing up to its maximum, and monotonically decreasing thereafter.

Assume \( f_m \) is maximised when \( m = m^* \). Then, \( m^* \) must be the largest integer that satisfies the following:

\[
f(m^* - 1) \leq f(m^*)
\]  

(J.3)
Also, \( m^* \) must be the smallest integer that satisfies the following:

\[
f(m^*) \geq f(m^* + 1)
\]  

(J.4)

Substituting (J.2) into (J.3) and (J.4), we have

\[
\frac{m^* - 2}{m^* - \frac{1}{D_{\min}}}(1 - \frac{(m^* - 1)\tau}{D_{\min}}) \leq \frac{m^* - 1}{m^* + 1 - \frac{1}{D_{\min}}}(1 - \frac{m^*\tau}{D_{\min}})
\]

(J.5)

and

\[
\frac{m^* - 1}{m^* + 1 - \frac{1}{D_{\min}}}(1 - \frac{m^*\tau}{D_{\min}}) \geq \frac{m^*}{m^* + 2 - \frac{1}{D_{\min}}}(1 - \frac{(m^* + 1)\tau}{D_{\min}})
\]

(J.6)

Simplifying (J.5), we have

\[
\left( m^* - \frac{n + 2 + \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}} \right) \times \left( m^* - \frac{n + 2 + \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}} \right) \leq 0
\]

Note that for \( m^* > 0 \), it is each to check that

\[
m^* - \frac{n + 2 + \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}} > 0
\]

Thus we have

\[
m^* \leq \frac{n + 2 + \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}}
\]

(J.7)

Because \( m^* \) is the largest integer that satisfies (J.7), we have

\[
m^* \leq \left| \frac{n + 2 + \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}} \right|
\]

(J.8)

Similarly, because \( m^* \) is the smallest integer that satisfies (J.6), the following lower bound on \( m^* \) can be derived:

\[
m^* \geq \left| \frac{-3n + 2 - \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}} \right|
\]

(J.9)

Let integer \( a \) be

\[
a = \left| \frac{-n + 2 + \frac{\tau}{D_{\min}}}{2n + 2\frac{\tau}{D_{\min}}} \sqrt{9n^2 + 8n^2\frac{D_{\min}}{\tau} + 6n\frac{\tau}{D_{\min}} + \frac{\tau^2}{D_{\min}^2} - 4n\frac{D_{\min}}{\tau}} \right|
\]

(J.10)
and integer $b$ be

$$
b = \left[ -3n + 2 - \frac{\tau}{\tau_{\min}} + \sqrt{9n^2 + 8n^2 \frac{D_{\min}}{\tau} + 6n \frac{\tau}{\tau_{\min}} + \frac{\tau^2}{\tau_{\min}^2} - 4n^2 \frac{D_{\min}}{\tau}} \right] \quad (J.11)
$$

Then, from (J.8) and (J.9), when $b \leq m \leq a$, $m$ maximise $f(m)$.

It is easy to verify that

$$0 \leq a - b \leq 1$$

and

$$a \geq 2$$

From the above discussion, we have

1. If $a - b = 0$, then there is a unique point $m^*$ that maximises $f(m)$, i.e., $f(m^*) = f(a) = f(b) \geq f(m)$ for all $m \geq 2$.

2. If $a - b = 1$, then both $m_1^* = a$ and $m_2^* = b$ maximise $f$. Thus $f(a) = f(b) \geq f(m)$ for all $m \geq 2$.

Lemma 4 follows from the above discussion. \qed
Appendix K

Matlab code for simulation

This appendix lists the main portion of matlab code used for simulation in Chapter 6. The allocation functions and schedulability test functions used for different local SBA schemes other than the NLA scheme are not listed here. These functions can be easily produced based on the definitions and schedulability test conditions of local SBA schemes by referencing the functions used for the NLA scheme listed in Fig. K.2.

```matlab
function vectU=UtilisationGen(n, U)
    vectU=zeros(1, n);
    sumU = U;
    for i=1:n-1,
        nextSumU=sumU.*rand((1/(n-i)));
        vectU(i)=sumU-nextSumU;
        sumU = nextSumU;
    end
    vectU(n)= sumU;
end

function vectP=PeriodGen(n, min, max)
    vectP = rand(1, n)*(max-min)+min;
end

function vectD=DeadlineGen(n, min, max)
    vectD = rand(1, n)*(max-min)+min;
end

function vectC=ComputationTimeGen(n, vectP, vectD, U)
    vectU=UtilisationGen(n, U);
    minScale = min(vectP, vectD);
    vectC = vectU.*minScale;
end
```

Figure K.1: The functions used for generating a synchronous message set
function vectH = SBA_N_L_A_(n, TTRT, vectC, vectP, vectD)
    vectH = zeros(1, n);
    for i = 1:n,
        if (vectD(i) < 2*TTRT)
            if ((TTRT < vectD(i) && vectD(i) <= vectP(i)) || (TTRT <= vectP(i) && vectP
                (i) < vectD(i)))
                vectH(i) = vectC(i);
            else
                vectH(i) = (TTRT/vectP(i)+1)*vectC(i);
        else
            qi = floor(vectD(i)/TTRT);
            if (qi == floor(vectP(i)/TTRT)+1)
                vectH(i) = vectC(i)/(qi-1);
            else
                temp = vectD(i)-(qi+1)*TTRT+(max(qi*TTRT/vectP(i),1))*vectC(i)/(qi-1);
                if (temp > 0)
                    vectH(i) = max(qi*TTRT/vectP(i),1)*vectC(i)/(qi-1)-1/qi*temp;
                else
                    vectH(i) = max(qi*TTRT/vectP(i),1)*vectC(i)/(qi-1);
            end
        end
    end
end

function bFlag = Schedulability_N_L_A_(n, TTRT, tau, vectC, vectP, vectD)
% generate the synchronous bandwidth allocation
vectH = SBA_N_L_A_(n, TTRT, vectC, vectP, vectD);

% testing the schedulable
bFlag = false;

temp = sort(vectD);
compare = TTRT - tau;
if (temp(1) < 2* TTRT)
    compare = min(TTRT - tau, temp(1) - TTRT - tau);
end

if (sum(vectH) <= compare)
    bFlag = true;
end
end

Figure K.2: The functions used for the NLA scheme, including allocation function and schedu-
ability test function
function bFlag = Schedulability(n, TTRT, tau, vectC, vectP, vectD, SBAScheme)
bFlag = false;
switch(SBAScheme)
    case 'N.L.A.'
        bFlag = Schedulability_N_L_A_A(n, TTRT, tau, vectC, vectP, vectD);
    case 'I.N.L.A.'
        bFlag = Schedulability_I_N_L_A_A(n, TTRT, tau, vectC, vectP, vectD);
    case 'NLA'
        bFlag = Schedulability_NLA(n, TTRT, tau, vectC, vectP, vectD);
    case 'L.A.'
        bFlag = Schedulability_L_A_(n, TTRT, tau, vectC, vectP, vectD);
    case 'LA'
        bFlag = Schedulability_LA(n, TTRT, tau, vectC, vectP, vectD);
    case 'Zheng'
        bFlag = Schedulability_Zheng(n, TTRT, tau, vectC, vectP, vectD);
    case 'PA'
        bFlag = Schedulability_PA(n, TTRT, tau, vectC, vectP, vectD);
    case 'FLA'
        bFlag = Schedulability_FLA(n, TTRT, tau, vectC, vectP, vectD);
end
end

Figure K.3: The function used for schedulability test based on different local SBA schemes

function MissRatio = Simulation(n, TTRT, tau, P_min, P_max, D_min, D_max, U_min, U_max, U_step, runTimes, SBAScheme, DEqualP)
    MissRatio = zeros(length(SBAScheme), n);
    runIndex = 0;
    for U = U_min:U_step:U_max
        runIndex = runIndex + 1;
        passed = zeros(1, length(SBAScheme));
        for i = 1:runTimes
            vectD = DeadlineGen(n, D_min, D_max);
            if (DEqualP == false)
                vectP = PeriodGen(n, P_min, P_max);
            else
                vectP = vectD;
            end
            vectC = ComputationTimeGen(n, vectP, vectD, U);
            for j=1:length(SBAScheme)
                if Schedulability(n, TTRT, tau, vectC, vectP, vectD, SBAScheme{j})
                    passed(j) = passed(j) + 1;
                end
            end
        end
        MissRatio(:,runIndex) = (runTimes-passed)/runTimes;
    end
end

Figure K.4: The main function used for simulation to produce the percentage of infeasible sets

214