Software Engineering Institute

Analysis of Reservation-Based Dual-Link Networks

Lui Sha
Shirishq S. Sathaye
Jay K. Strosnider

June 1992

http://www.sei.cmu.edu

Carnegie Mellon

Analysis of Reservation-Based Dual-Link Networks for Real-Time Applications

Lui Sha

Rate Monotonic Analysis for Real-Time Systems Project

Shirish S. Sathaye

Digital Equipment Corporation

Jay K. Strosnider

Electrical & Computer Engineering
Carnegie Mellon University

This technical report was prepared for the

SEI Joint Program Office
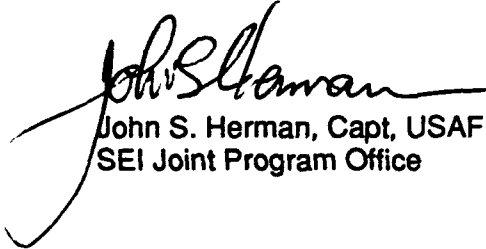ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official
DoD position. It is published in the interest of scientific and technical
information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

John S. Herman, Capt, USAF
SEI Joint Program Office

Contents

List of Figures

Analysis of Reservation-Based Dual-Link Networks for Real-Time Applications

Abstract: Next-generation networks are expected to support a wide variety of services. Some services such as video, voice, and plant control traffic have explicit timing requirements on a per-message basis rather than on the average. In this paper we develop a general model of reservation-based dual-link networks to support real-time communication. We examine the desirable properties of this network and the difficulties in achieving these properties. We then introduce the concept of *coherence* and develop a theory of coherent dual-link networks. We show that a coherent dual-link network can be analyzed as though it is a centralized system. We then discuss practical considerations in implementing a dual-link network, and implications of this work to address problems observed in the IEEE 802.6 metropolitan area network standard.

1 Introduction

Real-time communication, defined as communication with explicit timing requirements, is important to future networks which will concurrently support a wide variety of services. Examples include multimedia traffic, such as digital audio and digital video; and real-time computing traffic, such as plant process control and air-traffic control systems. In traditional applications of packet-switched networks, performance is measured by average throughput and average response time. However, guaranteed timing performance is needed for real-time communication. The desirable properties of a network that supports real-time communication include:

Predictable Operation: By *predictable* we mean that, given an arbitrary set of network connections, we can predict if timing constraints of all the connections can be met.

High Degree of Schedulability: Schedulability is the degree of network utilization at or below which individual message timing requirements can be insured. It can also be thought of as a measure of the capability of supporting timely connections.

Position-Independent Bandwidth Allocation: The amount of bandwidth allocated to a station must be position-independent and under protocol control.

Stability Under Transient Overload: When the network is overloaded and it is not possible to meet each connection's timing requirements, more critical connections must meet their timing requirements at the expense of less critical connections.

It may not be easy to achieve the above properties, as evidenced by the problems of IEEE 802.6 metropolitan area network standard, as discussed by several researchers [vAWZ90, CGL91, SS90]. Scheduling in a network is different from scheduling in a centralized environment. In a

---

centralized system, all resource requests are immediately known to the centralized scheduler. In a network, distributed scheduling decisions must be made with incomplete information. From the perspective of any particular station, some requests could be delayed and some may never be seen, depending on the relative position of the station in the network. The challenge is to achieve predictability under these circumstances.

In this paper, we develop an analytical model of reservation-based dual-link networks and use it to reason about the relationship between bandwidth requests on one link, and the patterns of slot usage by stations on the other link. The resulting model of slot usage serves as a foundation for studying the behavior of dual-link networks. We shall use this model to analyze the schedulability of periodic traffic and propose possible solutions to problems observed in IEEE 802.6.

The remainder of this paper is organized as follows: In **Section 2** we describe the architecture and operation of dual-link networks. We discuss the difficulties in scheduling traffic in dual-link networks and introduce the concept of *system coherence*. **Section 3** discusses a proposed protocol for media access control that results in both coherent operation using a coherent reservation protocol (CRP) and regulated access of the media through the flow control protocol. In **Section 4** we analyze the behavior of coherent systems and develop results about the relation between slot reservation patterns in coherent systems. **Section 5** discusses the scheduling of periodic traffic in a dual-link network that follows CRP and flow control. We introduce the notion of *transmission schedulability* for a dual-link networks and show that connections in a coherent dual-link network are transmission schedulable if they are schedulable in a centralized system. **Section 6** discusses practical considerations in implementing the conceptual model of a dual-link network, and the implications of this work to the addressing unpredictability observed in the IEEE 802.6 DQDB (distributed queue dual bus) protocol. **Section 7** makes concluding remarks and discusses future research directions.

2   Conceptual Framework

In this section, we first review the basic architecture of a reservation-based dual-link network as discussed in the IEEE 802.6 standard [Sta90]. However, we develop the bandwidth reservation abstraction using transmission queues in stations, instead of counters, since counters are simply an efficient implementation of queues. We then introduce the concept of system coherence as a basis for predictability of dual-link networks.

2.1   Architecture of Dual-Link Networks

A dual-link network consists of two slotted unidirectional links, say Forward Link (Flink) and Reverse Link (Rlink), as shown in Figure 1. Fixed-length slots are generated by slot generators of the corresponding links. Although the figure shows slot generators as separate functional units, the slot generation function can be embedded in stations at the end of the links. Each station is able to transmit and receive messages on both links. We assume that each message
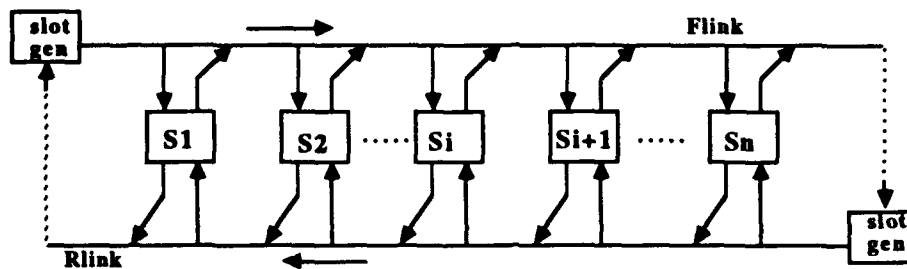
Figure 1: *Dual-Link Network Design*

is partitioned into one or more packets, and exactly one packet can be transmitted in a slot. We assume that a station wants to send a number of messages to another and call this a *connection* between the stations. In a high-speed metropolitan area network, the slot delay is small compared to the network delay. To simplify discussion, we will use the slot delay as the unit of measurement. We therefore assume that each slot is transmitted in unit time and stations are separated by an integral number of slot times. Stations reserve slots on Flink by making requests on Rlink. Since the delay for transmitting a single slot is small compared to network propagation delay, in this paper we ignore the slot delay by assuming that stations wish to make requests synchronously with the arrival of Rlink slots. In the following discussion, we will only discuss message transmissions on Flink and reservation on Rlink, because of symmetry in the network. Referring to Figure 1, stations on the right-hand side are called *downstream*. Stations on the left-hand side are called *upstream*.

Each Flink slot contains a BUSY bit to indicate whether or not the slot is used. BUSY=0 indicates an empty slot. A node may transmit in an empty slot by setting BUSY=1 and copying its packet into the slot. However, with BUSY bits alone, stations closer to the Flink slot generator can monopolize the link. To minimize this positional priority, stations use a REQ bit in Rlink slots to reserve Flink slots.

Before discussing the reservation mechanism, we describe Rlink slots and introduce an abstraction that will allow us to analyze if each station gets the Flink slots it requested. Since we discuss only Flink transmissions in this model, slots on the Rlink are used only to reserve Flink Bandwidth. Therefore Rlink slots carry request information consisting of the presence of a request and its priority. In our abstraction we represent an Rlink slot to contain an REQ bit and a priority field.[1] In addition, we imagine that the Rlink slot contains a field to hold the

---

[1]There may be alternate ways to communicate a request's priority level. For example, in IEEE 802.6 a separate REQ bit is used for each priority level, as discussed in Section 6.
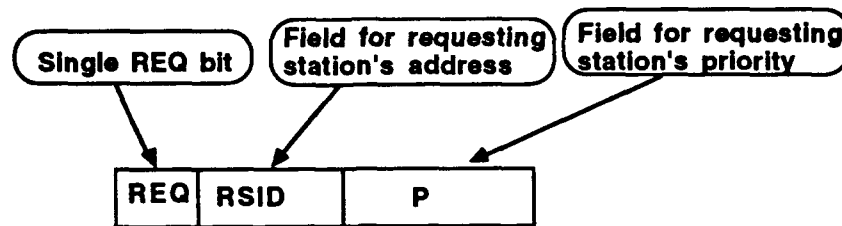
Figure 2: *Slots on Rlink with REQ Bit, RSID, and Priority Fields*

requesting station's address (RSID), as shown in Figure 2. The RSID field is not part of the implementation and is used only to facilitate analysis. If the REQ bit is set, the associated combination of the RSID and priority fields is defined as a request.

Flink slots can be considered to contain only a BUSY bit and data in this model. Furthermore, we introduce the concept of assignment of Flink slots to a station. That is, when an Rlink slot arrives at the head station, the next Flink slot is said to be *assigned* to the station that made the request. However the head station continues to release slots even if there are no Rlink requests. These slots are called *unassigned* slots. Assigned and unassigned slots are abstractions that we will use in the analysis.

A model of each station in the network is shown in Figure 3. Each station contains two sets of queues. For requests, a station contains a prioritized outgoing request queue that is used for holding pending requests in priority order. A station which wants to make a high-priority request can preempt a lower-priority request on the Rlink and replace it with its high-priority request. The preempted request is inserted in the station's outgoing request queue in priority order.

For transmission on the Flink, each station contains a prioritized *transmission queue*. Whenever requests pass the station on the Rlink, they are inserted into the transmission queue in priority order. For each unoccupied slot on the Flink, the station dequeues one request from the top of the transmission queue. In addition, there are additional buffers that are used for flow control purposes, which will be discussed in Section 3.

The dual-link architecture abstraction provides us with a convenient vehicle to reason about properties of a dual-link network. Finally, it is important to note that distributed scheduling decisions with incomplete information are unavoidable in a dual-link network. Some requests made by stations are seen by other stations after a propagation delay, while some requests may never be seen. For example in Figure 1, station S_{i+1} does not see requests from station S_i, and S_i sees requests from stations S_{i+k} after some delay. The challenge is to achieve predictability under these circumstances.
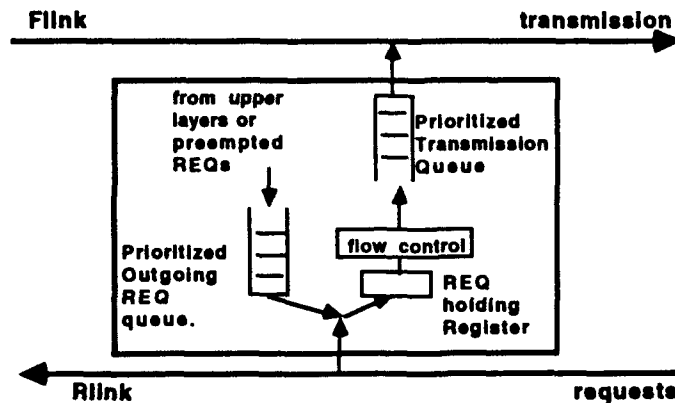
Figure 3: *Station Model Using Priority Queue*

2.2 The Concept of System Coherence

To address the distributed scheduling problem, we describe the fundamental concept of *system coherence*. Intuitively, coherence is a logical and orderly relationship between elements of a system. In the context of dual-link networks, the relationships that make a system coherent are: losslessness, consistency, and bounded priority inversion. In the following discussion, we define each of these concepts.

Definition 1 Lossless System:

A prioritized reservation system is said to be lossless if and only if each request from downstream stations is registered correctly. That is, a station copies each passing request from Rlink without error or loss.

Definition 2 Consistent System:

A prioritized reservation system is said to be consistent if and only if the queues of requests in different station queues are consistent with each other. That is, if request R_1 and request R_2 both exist in queue Q_a and queue Q_b, and if R_1 is ahead of R_2 in Q_a, then R_1 must also be ahead of R_2 in Q_b.

Note that in Figure 1, if requests from station S_2 are not correctly registered by S_1, station S_1 may not let unoccupied slots go by, and S_2 may be unable to transmit and meet its timing requirements. Therefore it is easy to see that lossless queues are necessary for predictable operation. Some additional concepts that bind slot types to requesting stations are necessary to illustrate the importance of consistent queues. We therefore defer the discussion to Section 4.2, where we give an example to show that inconsistent queues lead to unpredictability.

In an idealized priority scheduling system, a high-priority request will never need to wait for lower-priority traffic. In a real system, a higher-priority request may have to wait for lower-priority messages, The duration of such waiting is known as *priority inversion* [SRL90]. Priority inversion interferes with the operation of priority-based scheduling [SRL90]. For a system to be predictable, the worst-case priority inversion must be bounded by some function so that its impact can be taken into account in the analysis.

Definition 3 Bounded Priority Inversion:

A prioritized system is said to suffer from priority inversion if higher-priority activity can be delayed by lower-priority activity [SRL90]. The duration of priority inversion is said to be bounded with respect to the network size if the delay is not larger than $2kD$, where k is an arbitrary weight and D is the end-to-end network propagation delay.

An example of unbounded priority inversion is given in Section 3.1. We will show that in a coherent dual-link network $k=1$ and priority inversion is bounded by $2D$.

In summary, we define a coherent system as follows:

Definition 4 System Coherence:

A system is said to be coherent if it has the following properties:

- *It is a lossless system.*

- *It is a consistent system.*

- *Priority inversion is bounded.*

In the next section, we discuss the conditions that a system must satisfy to achieve the above properties.

3   Media Access Control

In this section, we will consider two protocols to control access to the dual-link network. First we describe a protocol for making reservations for slots on the Flink. Then we describe a flow control protocol that regulates the use of Flink slots.

3.1   Coherent Reservation Protocol

As discussed in Section 2.2 a coherent system must be lossless, consistent, and must have bounded priority inversion. In this section, we discuss the conditions to be satisfied or rules

to be followed to achieve these properties. The system will be lossless if the queues are lossless. The system will be consistent if the *self-entry rule*, and the *tie-breaking rule* (described below) are followed. The system will have bounded priority inversion if the station has priority queues, requests on the Rlink can be made autonomously, and the lower-priority requests can be preempted.

Before considering the above rules and conditions, we introduce the following notation: a request by station S_i at priority p is denoted as R_{ip}. When discussing requests of equal priority, the second subscript is dropped. $R_{ip} \prec R_{jq}$ denotes that R_{ij} is "ahead" of R_{jq}.

A condition for a lossless system is that all requests on Rlink must be entered into station queues without loss.

> **Condition 1:** Lossless queues
> The station must be fast enough to copy every request on the Rlink in the observed order without loss or error.

We now consider the rules for system consistency. The *self-entry rule* defines the relative ordering in which a station must make a self-entry in its transmission queue and a request on the Rlink.

> **Condition 2:** Self-entry rule
> A station that wishes to transmit must make a request on the Rlink before making a self-entry into its transmission queue.

The following example illustrates the importance of this rule:

Example 1 Consider three stations C and B and A which are at the same priority and A is downstream with respect to B, which is itself downstream with respect to C. Suppose B makes a self-entry R_b in its transmission queue and then attempts to make a request on the Rlink. Let B be prevented by making a request on Rlink by higher-priority requests until request R_a by station A passes by. On the request stream $R_a \prec R_b$ while in B's transmission queue $R_b \prec R_a$. After the requests are registered in station C, the transmission queue of C will have $R_a \prec R_b$ which is inconsistent with the queue of station B as shown in Figure 7.

Another rule for system consistency is the *tie-breaking rule*, which is designed to preserve the ordering of equal-priority requests on the Rlink. When a station preempts a request at a certain priority and inserts it into its outgoing request queue, it must give the preempted request higher priority than other equal-priority requests it observes on the Rlink. An efficient method of accomplishing this is to favor local requests over equal-priority requests on the Rlink.

Condition 3: Tie-breaking rule

- Preempted requests with equal priority are stored in FIFO order.
- When a request local to a station and Rlink requests have same priority, the local request replaces the Rlink request and the Rlink request is inserted in the station's outgoing request queue.

The importance of this condition is illustrated in the following example:

Example 2 Consider two requests of equal priority R_i and R_j on the Rlink, such that initially $R_i \prec R_j$. Let a station S_k preempt R_i and replace it with a high-priority request R_H. Now S_k wants to make request R_i. Let it observe request R_j. Since R_i and R_j are at the same priority, it cannot preempt R_j, (if local requests are not given higher priority than equal-priority requests on the Rlink), and has to let R_j pass. Eventually station S_k successfully makes request R_i. Note that now $R_j \prec R_i$ on Rlink, reversing the initial order. This reversing of the initial order makes station queues inconsistent.

We now consider three conditions to achieve bounded priority inversion. It is self-evident that priority queues minimize priority inversion. The other two conditions are autonomous request traffic and request preemption property. We discuss each of these conditions as follows:

Condition 4: Priority-ordered queues
All the requests in each station's transmission queue and outgoing request queue must be in priority order. Equal-priority requests are in FIFO order.

Ability to make autonomous requests is important; if stations are prevented from making requests on the Rlink by traffic on the Flink, priority inversions may occur. Suppose a station cannot make a new request if any of its previous requests are outstanding. This results in a lack of autonomy between making requests on Rlink and the presence of occupied slots on Flink. That is, requests from a station are "throttled" by traffic on the Flink.

Condition 5: Autonomous requests
The request generation on Rlink is said to be autonomous if each station can make its request at the Rlink independent of the traffic at the Flink.

The dependence of the request rate on Flink traffic may cause unbounded priority inversion, as shown in the following example:

Example 3 Consider a network with two stations S_1 and S_2, which are d_{12} slot times apart, as shown in Figure 4. Let station S_1 have n slots to transmit every period of $100n$, where n is large compared with d_{12}. Let station S_2 generate real-time traffic that must be transmitted in 1
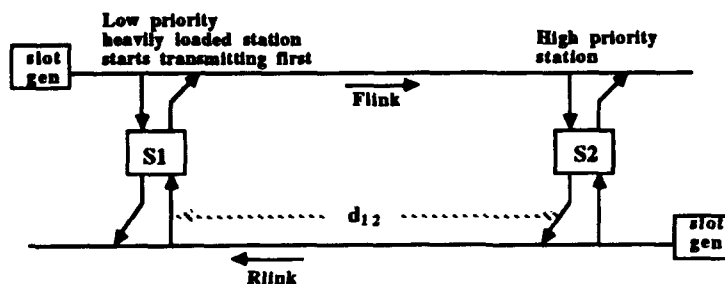
Figure 4: *Effect of Non-Autonomous Requests*

slot out of every 10 slots. Let S_2 be assigned a higher priority than S_1. Let the protocol require that a station cannot make a new request if it has an outstanding request.

Let S_1 start transmitting first. Since it is the only active station on the network, it transmits in the first n slots on the Flink. When S_2 desires to transmit, it will be able to make one request on the Rlink and must wait until its request is satisfied before it can make another request. The request from S_2 will reach S_1 after d_{12} slot times. Then S_1 will let an unoccupied slot go by on Flink that will be used by S_2 after an additional delay of d_{12} slot times. Therefore S_2 will be able to transmit once every $2d_{12}$ slot times. However the station wishes to transmit once every 10 slots. Therefore for $d_{12} > 5$, station S_2 will miss deadlines even though it has higher priority than S_1. This occurs because S_2 is prevented from making requests at a high priority by occupied Flink slots even though they are at a lower priority. Note that the priority inversion lasts as long as S_1 wishes to transmit. Therefore, since transmission time of S_1 may be longer than $2kD$ for any chosen k and D, the inversion is unbounded. Note that this priority inversion occurs even if the network utilization is as small as 11%.

This "throttling" effect is implemented in IEEE 802.6, and behavior similar to this example has been observed [vAWZ90]. Another effect of the "throttling" property is that it can also cause priority inversion among sources within a station. Consider a station with two sources at different priorities. If the station has an outstanding request, it is prevented from making another request at any priority. Therefore a high-priority request may be blocked by an outstanding lower-priority request.

Now consider the example of Figure 4 with the "throttling" restriction relaxed. Let S_1 start first in overload condition as before and transmit in all slots on the Flink. When S_2 starts, it will make one request every 10 slots irrespective of Flink traffic. After an initial delay of d_{12} slots, station S_1 will not use one slot every 10 slots. The first unoccupied slot will reach S_2 after an additional d_{12} slot times. Therefore S_2 will be prevented from transmitting for an initial $2d_{12}$ slot times but thereafter will be able to transmit once in every 10 slots and meet its timing requirements. Priority inversion between sources within a station will also be avoided. Since the station can
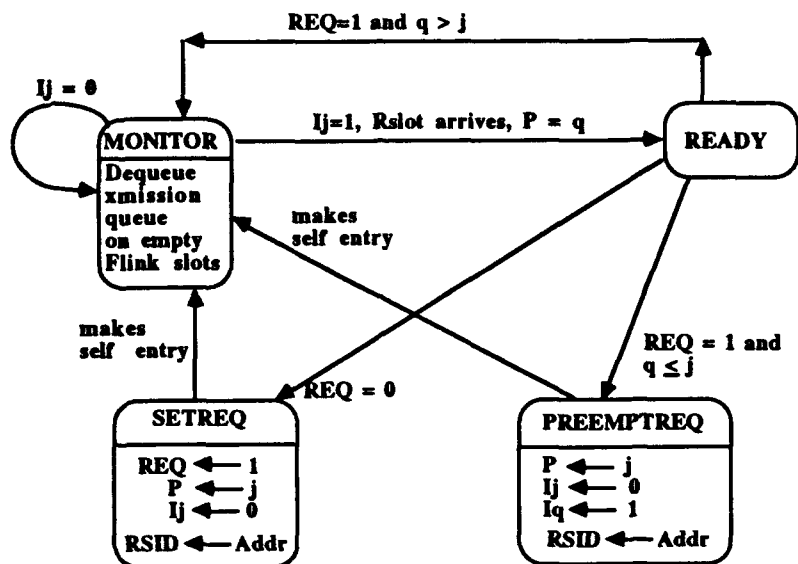
Figure 5: *Station Operation Under CRP*

make a new request even when a previous request is outstanding, a high-priority source in a station is not prevented from making a request when a low-priority request is outstanding.

Another condition for minimizing priority inversion is the request preemption rule.

Condition 6: Request preemption rule
A station which wants to make a high-priority request can preempt a lower-priority request on the Rlink and replace it with its high-priority request. The preempted request is inserted in the station's outgoing request queue in priority order.

The following example illustrates that a lack of request preemption can result in unbounded priority inversion.

Example 4 Consider a station S with a high-priority connection, and assume that S wants to make a request. Let all downstream stations have lower-priority connections. Without request preemption, the downstream stations can make requests in all Rlink slots and thus indefinitely prevent S from making requests. This results in unbounded priority inversion.

We now propose a coherent reservation protocol (CRP) that implements the conditions and rules described in this section. Consider the state diagram in Figure 5.

Definition 5 Coherent Reservation Protocol

In the MONITOR state, the station copies each request it sees on the Rlink into the appropriate position in the transmission queue. The position depends on the value of the P field of the Rslot that contains the request. When an unoccupied slot passes on the Flink, the station dequeues the entry at the top of its transmission queue. If the dequeued entry is not a self-entry, the station lets the slot go by. If the dequeued entry is a self-entry, the station will also set BUSY=1 and copy its packet into the slot. The request is then said to be satisfied.

If a station intends to transmit at priority j (I_j=1), it goes into the READY state whenever it observes an Rlink slot. In this state there are three possibilities to be considered, depending on the contents of the REQ and P fields of the observed slot.

- *If REQ = 1 and P = q, where $q > j$, then the station goes back to the MONITOR state.*

- *If REQ = 1 and P = q, where $q \leq j$, the station goes to the PREEMPTREQ state. In this state the station replaces q in the P field with j, clears I_j=0. It also replaces the contents of the RSID field with its own address. The preempted request is held in the outgoing request queue in priority order. The station then makes a self-entry into its transmission queue, and goes back to the MONITOR state.*

- *If REQ = 0, the station goes to the SETREQ state. It sets REQ = 1, P = j, writes its address into the RSID field, clears I_j=0, makes a self-entry into its transmission queue, and goes back to the MONITOR state.*

3.2 Flow Control Protocol

In this section we discuss a flow control mechanism. We give an example to show that flow control can be used to bound the delay of lower-priority connections without affecting higher-priority connections. Next we describe the flow control protocol.

The delay between a connection's request and the arrival of its assigned slot is equal to $2d_i$, where d_i is the distance between the source station and the Flink slot generator. Since d_i can equal multiple periods, the connection's requests from multiple periods can be accumulated in the transmission queue of the source station and all upstream stations. This creates the need for flow control, as illustrated in Example 5.

Example 5 Consider a network with two stations, as shown in Figure 6. Let connection τ_0 at station S_1 be transmitting a message in n slots every $100n$ slots, where n is large compared with propagation delay (in slots) between S_1 and S_2. Let this message have the lowest priority. Let station S_2 have a connection τ_1 to some unspecified downstream station. Let τ_1 want to transmit two packet in every 8 slots. Connection τ_1 makes 2 requests in every 8 slots and receives two assigned slots that it uses. Let a new connection τ_2 be established at t_0 which wishes to transmit 1 slot in every 4 slots. The condition at time t_0 is shown in Figure 6.
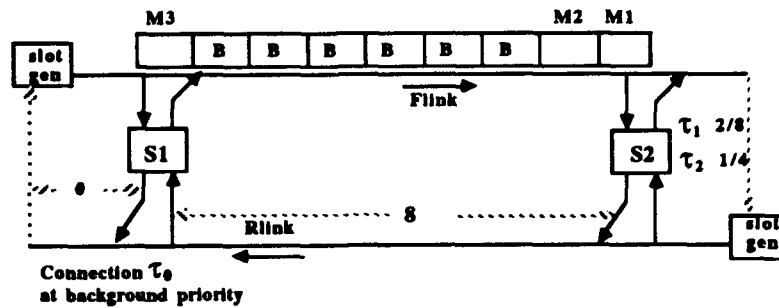
Figure 6: *Flow Control Example*

Connection	Num. Pkts	Period	Priority
τ_0	n	$100n$	Lowest (B)
τ_1	2	8	Medium (M)
τ_2	1	4	High (H)

Table 1: Connections on Network

The connections and their relative priorities are shown in Table 1.

Note that after time t_0, connection τ_2 transmits in slots M1, which have been assigned to τ_1. Therefore τ_1 cannot meet its timing requirements. Furthermore, τ_2 cannot transmit in 1 slot out of every 4. Therefore neither connection τ_1 nor τ_2 can meet its timing requirements. Note that if connection τ_1 had its source at S_1 instead of S_2, the new connection τ_2 would not cause it to miss its deadlines.

As shown in the above example, the establishment of a new connection may prevent previously established connections from meeting their timing requirements, depending on their positions relative to the new connection. Note that the total load is only 51 %. Depending on the parameters of the connection and the network, connections can miss deadlines even though work loads are arbitrarily small.

Suppose that a connection's periodic source makes C_i requests every T_i slot times. The objective of flow control is to ensure that no more than C_i requests from the connection are satisfied every T_i. Together with system coherence, flow control provides a foundation for analyzing the schedulability of coherent dual-link networks.

Section 2 discussed the station model for dual-link networks and briefly mentioned the need for flow control buffers. We now discuss the flow control architecture in detail. In the station,

requests at each priority will be inserted into a FIFO buffer associated with that priority level. In addition, each buffer is associated with a set of packet-counters (PCs), a set of propagation delay timers, and a set of period-timers (PTs). One counter-timer triple is assigned to a connection at that priority level.

Definition 6 Flow Control Protocol

1. *Initialization: When the upper network layer protocol initiates a connection τ at station S_i with period T and number of requests per period C, the period-timer is preset to T, the packet-counter is preset to C, and the propagation delay timer is preset to $2d_i$.*

2. *Operation:*

   (a) *When the first request of τ arrives at station S_i, the propagation delay timer starts counting down. The timer expires after a time equal to twice the propagation delay between the station and the Flink slot generator. When the timer expires, the request is inserted into the transmission queue. The period-timer starts counting down and the packet-counter is decremented.*

   (b) *Before the period-timer expires, whenever a request for the connection arrives (or is present in the flow control buffer), it is inserted into the transmission queue, and the packet-counter is decremented. If the counter reaches zero before the timer expires, no additional requests can be inserted into the transmission queue until the timer is reset. This ensures that no more than C requests are inserted into the transmission queue per period of T.*

   (c) *When the timer expires, both the packet-counter and the timer are reset and the process continues.*

3. *When the upper network layer protocol disconnects the connection, the above process continues when the FIFO buffers are emptied.[2] The timer-counter pair is then reset and made available for new connections.*

The above protocol can be optimized to reduce the total number of timers and counters. However the optimized implementation is outside the scope of this paper.

4 Analysis of Coherent Reservation Protocol

We now show that the conditions described above result in coherent systems. We assume that a network follows CRP, and prove that station queues are consistent, priority inversion is bounded, and the system is coherent.

---

[2]In this paper we do not consider abrupt connection termination.

4.1 System Consistency

We first show that station queues in a dual-link network that follows CRP are consistent with each other. Lemma 1 shows that the order of equal-priority requests on Rlink is maintained. Lemma 2 shows that the order of equal-priority requests in station transmission queues is the same as their order on the Rlink. Lemma 3 combines the previous lemmas to show that equal-priority requests in station queues are consistent with each other. Lemma 4 observes that since station queues are in priority order, different priority requests in station queues are consistent. Then by Lemmas 3 and 4, Theorem 5 shows that the system is consistent.

Lemma 1 *In a multi-priority dual-link network that follows CRP, each station preserves the order of equal-priority requests on the Rlink.*

Proof:

Consider two equal-priority requests, R_i and R_j, that pass station S. Without loss of generality, let $R_i \prec R_j$ on the Rlink. We must show that preemption does not reverse the order between R_i and R_j if the tie-breaking rule is used. There are only the following four cases to be considered:

Case 1: Neither R_i nor R_j are preempted by S.
In this case the lemma is true since no preemption occurs and $R_i \prec R_j$ by assumption.

Case 2: Only R_i is preempted by S.
The order between R_i and R_j can be reversed only if R_j passes S before the station can make the preempted request R_i on the Rlink. But by the tie-breaking rule, the station favors preempted request R_i over R_j; therefore, R_j cannot pass station S if the station is waiting to make preempted request R_i, and so order reversal is not possible.

Case 3: Only R_j is preempted by S.
In this case the lemma is true since R_i is not preempted and remains ahead of R_j.

Case 4: Both R_i and R_j are preempted by S.
In this case both R_i and R_j will exist in the outgoing request queue of Station S. Then by the request preemption rule, the preempted requests are held in the outgoing request queue in FIFO order. Hence $R_i \prec R_j$ in the outgoing request queue. Therefore by operation of the CRP protocol, R_i and R_j will reappear on Rlink in the order that $R_i \prec R_j$. □

Lemma 2 *In a dual-link network that follows CRP, for any pair of equal-priority requests R_i and R_j, if $R_i \prec R_j$ on the Rlink, then whenever both R_i and R_j exist in the same queue, $R_i \prec R_j$ in each station's transmission queue and outgoing request queue.*

Proof:

Due to CRP, station queues are lossless; that is, R_i and R_j will be copied in the transmission queue without loss. Condition 4 ensures FIFO order. Hence $R_i \prec R_j$ in the transmission queue.

Similarly if a station preempts both R_i and R_j, then $R_i \prec R_j$ in the outgoing request queue. Therefore the Lemma follows.   □

Lemma 3 *In a multi-priority dual-link network that follows CRP, equal-priority requests in station queues are consistent. That is, for two equal-priority requests R_i and R_j, if $R_i \prec R_j$ in any station queue, then $R_i \prec R_j$ in every station queue where both R_i and R_j exist.*

Proof:

From Lemma 2, since equal-priority requests in all station queues are consistent with the order of requests on the Rlink, and since by Lemma 1, the order of requests of equal-priority requests is maintained, it must be the case that if $R_i \prec R_j$ in any station queue, then $R_i \prec R_j$ in every station queue where both R_i and R_j exist. □

Lemma 4 *In a multi-priority dual-link network that follows CRP, different priority requests in station queues are consistent. That is, for two different priority requests R_{ip} and R_{jq}, if $R_{ip} \prec R_{jq}$ in any station queue, then $R_{ip} \prec R_{jq}$ in every station queue where both R_{ip} and R_{jq} exist.*

Proof:

Let $R_{ip} \prec R_{jq}$ in station S_k. Since the dual-link network follows CRP, station queues are in local priority order. Therefore R_{ip} is at higher priority than R_{jq}. Since every station queue is in priority order, $R_{ip} \prec R_{jq}$ in every queue where both exist.   □

Theorem 5 *In a dual-link network that follows CRP, station queues are consistent with each other.*

Proof:

By Lemma 3, equal-priority requests are consistent in station queues. By Lemma 4, different-priority requests are consistent in station queues. Therefore the theorem follows.   □

4.2 Bounded Priority Inversion and System Coherence

We have shown that a dual-link network using CRP has consistent queues. To show that the system is coherent, we need to demonstrate that priority inversion is bounded. We begin by establishing a relationship between requests on the Rlink and the pattern of Flink slot usage. Then in Theorem 7 we show that a request cannot be satisfied by a slot assigned to a higher-priority request. In Theorem 8 we show that priority inversion is bounded by the round trip network delay. Finally, the combination of consistent queues and bounded priority inversion results in system coherence.

We introduced the concept of Flink slots *assigned* to a station in Section 2. When a Rlink slot arrives at the head station, the next Flink slot is said to be *assigned* to the station that made the request. However the head station continues to release slots even if there are no Rlink requests. These slots are called *unassigned* slots. The importance of this assignment abstraction is that if each station were to use only its assigned slot, it would be possible to determine the worst-case slot usage patterns by stations. We show later that coherent systems do exhibit the above behavior. First, we will show that an incoherent system exhibits unpredictable behavior depending on the location of unassigned slots.

Example 6 Consider the network in Figure 7, with three stations, say A, B and C, that wish to transmit at the same priority. The slot generator has assigned slots to station requests in the order shown in Figure 7, where A_a is assigned to Station A, A_b assigned to B, and A_c assigned to C. Therefore station requests on the Rlink must have been in the order R_a (request by station A), followed by R_b, followed by R_c. Note that the queue in station B is inconsistent with the ordering of the requests. This inconsistency can be caused as demonstrated in Example 1. After passage of some time, the first slot will have moved past station C. Therefore C will have dequeued the entry at the top of its queue. The slots will be used by the stations as follows: slot A_a used by station B; slot A_b used by station A; and slot A_c used by station C.

Notice from Figure 7 that station B will use the slot assigned to A even though the request from B is outstanding. Station B's request is satisfied by a slot earlier than its assigned slot, while station A's request is satisfied by a slot after its assigned slot. Station C's request is satisfied by its assigned slot.

However, if the unassigned slot shown in Figure 7 had been present ahead of the assigned slots, the following pattern of slot usage would have occurred: unassigned slot used by station B, slot A_b deassigned; slot A_a used by station A; slot A_b used by station C and slot A_c deassigned.

In this case, notice that although station C uses A_b, B's request had already been satisfied by an earlier slot and hence A_b was deassigned. Requests by stations B and C are satisfied by slots earlier than their assigned slots, while station A's request is satisfied by its assigned slot.

Therefore the behavior of the system depends on the presence and location of unassigned slots. In particular it is not possible to predict whether station A's request will be satisfied by its assigned slot, an earlier slot, or a later slot. This occurs because the transmission queue in
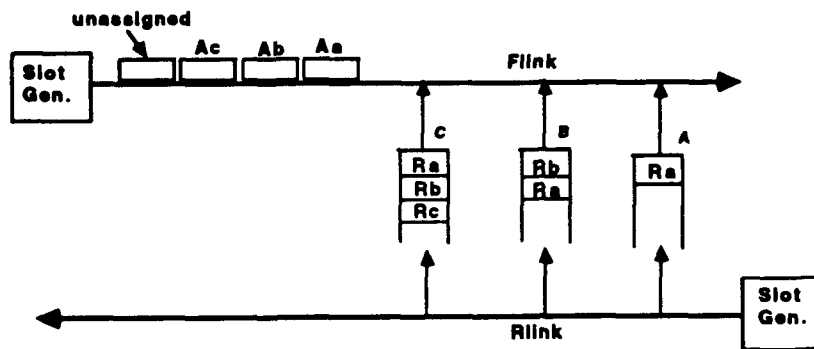
Figure 7: *Unpredictable Behavior of Inconsistent Systems*

station B is inconsistent with the other station queues. Therefore, if the queues in the stations are inconsistent, the behavior of the system is *unpredictable*. We will now show that in a coherent multi-priority system , a request cannot be satisfied by a slot assigned to a request of higher priority. To reason about equal-priority requests in a multi-priority system, we introduce the notion of *effective priority*.

Definition 7 Effective Priority:

Given two equal-priority requests R_i and R_j, if $R_i \prec R_j$ on the Rlink, then we consider R_i to have a higher effective priority than that of R_j.

Lemma 6 *Slot usage patterns caused by effective priorities are equivalent to those caused by priorities.*

Proof:

Consider two requests R_i and R_j. Suppose R_i has a higher assigned priority than that of R_j. The by Condition 4 (priority queues) R_i will be ahead of R_j in all queues where both appear.

Now consider R_i and R_j to have the same assigned priority and $R_i \prec R_j$ on the Rlink. Then by Lemma 2, $R_i \prec R_j$ in all queues where both appear.

The slot usage patterns by requests is determined by their relative positions in station queues. Therefore the lemma follows.

Theorem 7 *In a multi-priority coherent system, a request cannot be satisfied by a slot assigned to another request with a higher priority or higher effective priority.*

Proof:

Consider two requests R_i and R_m such that R_i has higher priority than R_m. By Lemma 6 this also covers the case of R_i having higher effective priority than that of R_j.

Consider stations S_i and S_m that generated R_i and R_m respectively. It is sufficient to consider only the operation of S_i and S_m because, as defined in the operational rule of CRP, a request can only be satisfied by the station that generated it.

Let the slots assigned to R_i and R_m be A_i and A_m respectively. Suppose that R_i is waiting to be satisfied and A_i remains to be an assigned slot, and that R_m is satisfied by A_i. We show that this is not possible under the CRP protocol.

Consider the case in which S_i and S_m are in fact the same station. Since R_m is satisfied by A_i while R_i is waiting in the queue, R_m must be ahead of R_i in the queue. But the priority of R_i is higher than that of R_m. This contradicts Condition 4 which states that station queues are priority ordered. We now consider the case where S_i and S_j are two different stations. There are two cases.

Case 1: Station S_m is upstream with respect to S_i.

Since A_i has been generated, request R_i must have traveled all the way upstream and reached the Flink slot generator. Therefore station S_m must have entered R_i in its queue. Therefore both R_i and R_m are in the queue of S_m. The assumption that R_m is satisfied by A_i while R_i is waiting implies lower-priority R_m is ahead of higher-priority R_i in the transmission queue of S_m. This contradicts the assumption that queues are priority ordered.

Case 2: S_i is upstream with respect to S_m.

In this case, the assigned slot A_i passes station S_i first. In the following discussion we ignore any station with an empty transmission queue since it does not affect the analysis.

The assigned slot A_i can be used by station S_i or any station S between station S_i and S_m, unless there is a non-self entry request at the top of all their transmission queues. In this case, A_i will be let go and each of these non-self entry requests are dequeued. As a result, A_i will be available for station S_m to use. However, we show that station S_m cannot use A_i to satisfy R_m. Let R_H be the non-self entry request at the top of the queue of station S_i.

We first establish the intermediate result that moving downstream from station S_i to the upstream station next to S_m, S_{m-1}, the priorities of their top non-self entry requests are non-decreasing. We shall refer to this result as the *non-decreasing priority argument*. Suppose that this argument is false and consider any pair of stations from S_i to S_{m-1}, say station S_k and its next downstream station S_{k+1}. Let the non-self entry requests at the top of their transmission queues be R_k and R_{k+1}

respectively, with R_k having a higher priority than R_{k+1}. Since R_k is a non-self entry request, it must have been generated by either station S_{k+1} or a station further downstream. In either case, request R_k must appear at station S_{k+1}'s transmission queue. Since R_k is presumed to have higher priority, it should be ahead of R_{k+1}. This contradicts the assumption that R_{k+1} is at the top of station S_{k+1}'s transmission queue. This completes the proof of the non-decreasing priority argument.

We now prove that station S_m cannot use A_i to satisfy request R_m. Let the non-self entry request at the top of S_{m-1} be R_{m-1}. Since R_{m-1} is a non-self entry request at S_{m-1}, it must be generated by either station S_m or a station further down stream. In either case, request R_{m-1} must appear at the transmission queue of station S_m.

Because of the non-decreasing priority argument, the priority of R_{m-1} is at least as high is the non-self entry request at the top of S_i's transmission queue, request R_H. Since the priority of request R_H is higher than that of R_i and the priority of R_i is higher than that of R_m, the priority of R_{m-1} is higher than that of R_m.

As a result, R_{m-1} must be ahead of R_m at the transmission queue of S_m. Hence, when A_i passes station S_m, S_m can either

- Use A_i to satisfy R_{m-1} if R_{m-1} is a self entry of station S_m, or
- Let go of A_i and dequeue R_{m-1} if R_{m-1} is not a self entry.

Either of these two cases contradicts the assumption that R_m is satisfied by A_i. □

Theorem 8 *For any periodic connection in a dual-link network that follows CRP, the maximum duration of priority inversion is bounded by $2d_i$ where d_i is the distance in slot times between the source station and the Flink slot generator.*

Proof:

Suppose there is a low-priority connection τ_L at the head station that occupies every Flink slot.

Consider a connection τ_H of higher priority than τ_L. Let τ_H generate a request at time $t=t_0$. The request of connection τ_H cannot be delayed on the Rlink by lower-priority requests due to the request preemption rule of CRP. However, since all Flink slots are being used by τ_L, τ_H is prevented from transmission. Excluding effects of preemption on the Rlink, the request of τ_H will reach the Flink slot generator at time $t_0 + d_i$. An Flink slot will be assigned to τ_H. By Theorem 7, τ_L cannot use this assigned slot. With an additional delay of d_i, the assigned slot will arrive at τ_H's station and can be used by τ_H.

Therefore after τ_H generates a request it can be delayed by lower-priority connections for a maximum of $2d_i$ slots. The theorem follows. □

Theorem 9 *A dual-link network that follows CRP is coherent.*

Proof:

This theorem follows because of Theorem 5 and Theorem 8.   □

5 Scheduling Dual-Link Networks

In this section we investigate the use of a coherent dual-link network for scheduling periodic real-time traffic. We focus on periodic traffic scheduling for the following reasons:

- Voice and video traffic sources are periodic in nature. Even compressed video may be periodic, since practical VLSI compression devices, at least those for MPEG, [Gal91] and Px64, [Lio91], may have "rate-control" buffers, so that the compressed-video output is at a constant data rate.

- Traditional real-time applications generate periodic traffic from sampled data systems. Although aperiodic real-time traffic may exist in the network, it can be handled by aperiodic server algorithms, e.g., the sporadic server [Spr90] or the deferrable server algorithm as demonstrated by Strosnider [Str88], which can be analyzed as if it is periodic.

- Non-real-time traditional aperiodic traffic such as interactive data processing. File transfers can be given either an aperiodic server or served at background priority.

Scheduling dual-link networks is different from scheduling a centralized system, since some requests are never seen by some stations. Hence we cannot directly use scheduling results from centralized systems. Nonetheless, we will show that if a set of connections is schedulable in a centralized system, it is also schedulable in a dual-link network, allowing for initial delay. We will call periodic traffic between a source station and destination station a *connection* in the rest of the paper. Each connection τ_i wishes to transmit a message of C_i fixed-size packets per period T_i. Packet size is same as the slot size on the network links. We assume that the time to transmit each slot is unity, and that each connection's period is assumed to be an integral number of slot transmission times.

Consider a set of periodic connections $\tau_1, \tau_2, \ldots \tau_n$ arranged in decreasing priority order. We are interested in the worst-case delay for a periodic connection. We first show the equivalence between relative results when its request is delayed by all higher-priority requests. A useful lemma in centralized system scheduling is the critical instant Lemma 10 [LL73].

Lemma 10 *Given a set of periodic activities in a centralized system, the longest completion time for any activity occurs when it is initiated at the critical instant. The critical instant is the time at which a task is initiated along with all tasks of higher priority.*

Lemma 11 *Consider two connections τ_H and τ_L arranged in decreasing priority order. Let τ_H and τ_L be in stations S_H and S_L respectively. Let propagation delay between S_L and S_H be d_{LH}. The preemption effect on the τ_L can be modeled as though τ_H is in the same station as τ_L with starting times modified as follows: if S_H is downstream to S_L then d_{HL} is added to the starting time of τ_H. Otherwise it is subtracted from the starting time of τ_H.*

Proof:

Case 1: τ_H is in a downstream station S_H.

τ_L will experience preemption from τ_H after $t_0 + k + d_{LH}$. This is equivalent to having τ_H in station S_L but starting after time $t_0 + k + d_{LH}$.

Case 2: τ_H is in an upstream station S_H.

τ_L will experience preemption due to τ_H after $t_0 + k - d_{LH}$. This is equivalent to having τ_H in station S_L but starting after time $t_0 + k - d_{LH}$.

Lemma 12 *Given a set of period connections in a dual-link network, the longest delay experienced by any request initiated at time $t=0$ is no greater than the delay that results when all equal- or higher-priority connections are located in the same station and generate requests at time $t=0$.*

Proof:

For any given connection τ_i at station S_i, move all higher-priority connections into S_i using the transformation technique of Lemma 11. This preserves the preemption effects on τ_i. Since τ_i and all higher-priority connections are now in the same station, the scheduling problem is a centralized one. Under this condition, Lemma 10 applies. The lemma follows.

Lemma 13 *Consider a set of n connections $\tau_1, \tau_2, \cdots \tau_n$ arranged in decreasing priority order. In a dual-link network under CRP and the flow control protocol, a request can only be satisfied by its assigned slot.*

Proof:

Consider any connection τ_i from station S_i, that makes C_i requests every period T_i. Let each request be denoted R_i and the corresponding assigned slot be denoted A_i. There are only two cases.

Case 1: Request R_i from τ_i is not preempted before reaching the Flink slot generator.

In this case R_i is not delayed by preemptions and the slot A_i will arrive at S_i exactly $2d_i$ time units later. By the flow control protocol, A_i will be used by τ_i unless it has

been used earlier by some other station. We show that A_i cannot be used earlier by any other station.

There are three subcases to be considered:

Case 1a: τ_i is the highest-priority connection.
In this case by Theorem 7, A_i cannot be used by any other connection. By the flow control protocol, τ_i will not be ready to transmit until the first A_i arrives at S_i. Therefore τ_i will use the first C_i assigned slots (A_i). Further, since τ_i is not preempted, C_i assigned slots will arrive at S_i exactly one period T_i apart and will be used by τ_i.

Case 1b: τ_i is not the highest-priority connection and higher-priority connection τ_h from station S_h makes a request R_h such that $R_h \prec Ri$.
A_h, (slot assigned to R_h) will arrive at the source of connection τ_h before A_i arrives at the source of τ_h. Further, A_h arrives at S_h exactly $2d_h$ units later. Hence by the flow control protocol, τ_h uses A_h. Therefore τ_h cannot use A_i.

Case 1c: τ_i is not the highest-priority connection and higher-priority connection τ_h makes a request R_h such that $R_i \prec R_h$.
In this case $A_i \prec A_h$, and therefore A_i will arrive at the source of connection τ_h before the arrival of A_h. However τ_h will not use A_i since it will not be ready to transmit at this time due to the flow control protocol.

Case 2: Request from τ_i is preempted by high-priority requests.

Consider a connection τ_h which is higher priority than τ_i and has C_h packets to transmit every T_h. Let the requests by τ_h be denoted as R_h.

Let R_h preempt R_i. Therefore $R_h \prec R_i$ on the Rlink and $A_h \prec A_i$ on the Flink. By Theorem 7, connection τ_i cannot use a slot assigned to a high-priority request. Therefore τ_i cannot use A_h. By the flow control protocol, τ_h is ready to use A_h when it arrives at the source of τ_h, and cannot use more than C_h per period. Therefore τ_h will use its assigned slots. Also by the flow control protocol, τ_i is ready by the time A_i arrives.

Therefore each R_i can only be satisfied by A_i.

□

Because of the potentially long propagation delay in wide area networks, the traditional notion of schedulability needs to be extended to take the propagation delay into account. We introduce the notion of transmission schedulability.

Definition 8 Transmission Schedulability:

A connection τ_i is said to be transmission schedulable, (t-schedulable) if it can transmit C_i packets per period T_i, after an initial delay bounded by $2d_i + T_i$, where d_i is the propagation between the connection's station and the head station.

Theorem 14 *Given a set of periodic connections, if the set of periodic connections is schedulable in a centralized preemptive priority-driven system with zero propagation delay, then the set of connections is t-schedulable in a dual-link network.*

Proof:

> Since we have shown that the worst-case preemption delay experienced by a request in a dual-link network is same as the delay experienced in a centralized system with the same connection set, if the connection set is schedulable in a centralized system, then each connection τ_i will be able to make C_i requests every T_i. Therefore the Flink slot generator will receive C_i requests every T_i from connection τ_i after an initial delay bound by $d_i + T_i$. The Flink slot generator will therefore assign C_i slots to τ_i every period T_i after this initial delay. By Lemma 13, connection τ_i can always use its assigned slots. Therefore, it will be able to transmit its message every period after a delay bound by $2d_i + T_i$. Therefore the theorem follows. □

Theorem 15 *In a t-schedulable coherent dual-link network, a connection τ with C packets to transmit per period T will require $C\lceil 2d_i/T\rceil$ buffers in the source station of the connection.*

Proof:

> By the flow control protocol, the source of the connection will not transmit until $2d_i$ slot times after the arrival of the first request. In this time $C\lceil 2d_i/T\rceil$ packets will arrive at the station for connection τ and must be buffered. Therefore *at least* $C\lceil 2d_i/T\rceil$ buffers will be necessary.
>
> Since the network is t-schedulable, the source station will be able to transmit C packets every T after the initial delay of $2d_i$. Therefore the packet arrival rate at the source station is equal to the transmission rate and additional buffers are not necessary.
>
> Hence the theorem follows. □

Given a set of connections that is t-schedulable, the end-to-end delay experienced by a message of any connection τ_i is given by

$$\text{End-to-end Delay} = 2d_i + T_i + D_{prop}(i, t)$$

where $D_{prop}(i, t)$ is the propagation delay between the source station S_i and the destination station S_t of connection τ_i.

6 Engineering Considerations

In this paper we have developed a model of a dual-link network which allows us to achieve a high degree of schedulability and exhibits predictable timing behavior. In this section we first

discuss implementation considerations for a dual-link network architecture. We then briefly compare our implementation with IEEE 802.6.

6.1   Implementation Considerations

The dual-link network abstraction in previous sections was designed to facilitate analysis. We now reconsider this model from an implementation standpoint. First, station queues can be replaced by a set of counters similar to those in IEEE 802.6 [Sta90]. We considered Flink slots to consist of the BUSY bit and data, and Rlink slots to consist of the REQ bit, and the priority field. Since each link is actually used for reservation of the opposite link and transmission of data, slots on each link should consist of the BUSY bit, data, a REQ bit, and the priority field. A further optimization might be to omit the REQ bit and let the zero value in the priority field denote the lack of a request.

We now discuss the priority field. A significant aspect of priority-based scheduling in real-time systems is the number of priorities that should be supported by the arbitration logic. Ideally, there should be as many priority levels as the different connection periods. When the priority levels are fewer than the number of different periods, schedulability is reduced as discussed in [SRL91]. In this paper our t-schedulability definition requires that each connection must be able to transmit one message every period. This is equivalent to centralized scheduling in which each periodic activity must meet its end-of-period deadline. [SRL91] shows that the schedulability loss is negligible with 256 priority levels. Ideally a dual-link network for real-time applications should have 256 priority levels, although it may be possible to meet the t-schedulability requirement with fewer priority levels, depending on the characteristics of connections in the network.

We proposed a protocol in which priority is implemented as an 8-bit encoded field to yield 256 priority levels. This protocol allows preemption of lower-priority slots by higher-priority slots. Each slot contains an 8-bit encoded priority field in the header. Higher numbers are used to indicate higher priorities. All zeros in the priority field can indicate the absence of a request. A station that wants to make a request at priority i behaves as follows:

> If the next slot received contains a request at priority j, then
>
> - If $j > i$, the station waits for the next slot.
> - If $j \leq i$, the station replaces the priority field j of the slot with i and stores j in a prioritized request queue.
>
> Therefore the station preempts lower-priority reservations with higher-priority reservations.

The advantage of the encoded priority field is that it allows the implementation of 256 priority levels with only 8 bits of overhead in each slot. A simplified implementation of request preemption logic is shown in Figure 8. The slot priority from the link is passed through a single-bit

Figure 8: *Proposed Request Preemption Circuit*

delay and compared bit by bit with the station priority that is stored in the shift register. As long as the priority bits match, the output of the exclusive-OR gate is zero and the link priority is output. As soon as the priority bits differ, the station priority bits are output if they have a higher priority. Otherwise the link priority bits continue to be transmitted. Note that the logic assumes that the most significant priority bits in the slot are received first.

6.2 Implications to IEEE 802.6

There are two main functional differences between our dual-link network model and IEEE 802.6 DQDB. First, in an attempt to achieve fairness, in IEEE 802.6 a station cannot make a new request on the Rlink if its previous request is outstanding. This makes the request traffic non-autonomous and dependent on the traffic on the Flink. As we have shown, this may result in unbounded priority inversion and make the system incoherent. The second less serious difference is that a station in IEEE 802.6 can use a slot on the Flink before making a request on the Rlink, provided its CD counter is zero. This is acceptable when the system is schedulable. When the system is overloaded, it is not possible to predict which station will miss deadlines.

IEEE 802.6 implements priority by having a separate REQ bit for each priority level. Because of this implementation, it is not possible to implement a large number of priority levels without excessive overhead. Hence IEEE 802.6 implements only 4 priority levels. As we have shown, this may not be sufficient and may result in low schedulable utilization, depending on the characteristics of the connections in the network.

7 Conclusions and Future Work

We have developed a general model of reservation-based dual-link networks and used it to reason about the relationship between station request patterns and slot usage patterns. We introduced the concept of system coherence and examined the properties of coherent systems. We showed that a coherent dual-link network can be analyzed similarly to an equivalent centralized system in terms of its schedulability for periodic message traffic.

A number of important issues remain to be addressed.

Bandwidth allocation and overload management: Overload management is a challenging problem in a metropolitan area network because scheduling decisions are made in a distributed manner. Nevertheless, we must have the the ability to specify an arbitrary subset of traffic sources that meet deadlines even under overload.

Integration between periodic and aperiodic messages: We need to extend this analysis to address both periodic and aperiodic traffic in a unified framework.

Effect of introducing erasure nodes: When a station receives a packet, the slot, in principle, can be "erased" and be used again. The use of erasure nodes and their effect on network predictability needs to be considered.

8 Acknowledgements

References

[CGL91] M. Conti, E. Gregori, and L. Lenzini. A methodological approach to an extensive analysis of DQDB performance and fairness. *IEEE Journal on Selected Areas in Communications*, 9(1):76–87, January 1991.

[Gal91] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[Lio91] M. Liou. Overview of the px64 kbits/s video coding standard. *Communications of the ACM*, 34(4):59–63, April 1991.

[LL73] C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 30(1):46–61, January 1973.

[Spr90] Brinkley Sprunt. *Aperiodic Task Scheduling for Real-Time Systems*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA 15213, August 1990.

[SRL90] Lui Sha, R. Rajkumar, and J.P Lehoczky. Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on Computers*, 39(9):1175–1185, September 1990.

[SRL91] L. Sha, R. Rajkumar, and J. Lehoczky. Real-time computing using Futurebus+. *IEEE Micro*, June 1991.

[SS90] K. Sauer and W. Schodl. Performance aspects of the DQDB protocol. *Computer Networks and ISDN systems*, 20(1–5):253–260, December 1990.

[Sta90] IEEE 802.6 Distributed Queue Dual Bus - Metropolitan Area Network - Draft Standard - Version P802.6/D15, October 1990.

[Str88] J.K Strosnider. *Highly Responsive Real-Time Token Rings*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, August 1988.

[vAWZ90] H.R van As, J.W. Wong, and P. Zafiropulo. Fairness, priority and predictability of the DQDB MAC protocol under heavy load. *Proceedings of the International Zurich Seminar*, pages 410–417, March 1990.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS None			
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited			
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A						
4. PERFORMING ORGANIZATION REPORT NUMBER(S CMU/SEI-92-TR-10			5. MONITORING ORGANIZATION REPORT NUMBER(S) ESD-TR-92-10			
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute		6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office			
6c. ADDRESS (City, State and ZIP Code) Carnegie Mellon University Pittsburgh PA 15213			7b. ADDRESS (City, State and ZIP Code) ESD/AVS Hanscom Air Force Base, MA 01731			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION SEI Joint Program Office		8b. OFFICE SYMBOL (if applicable) ESD/AVS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003			

8c. ADDRESS (City, State and ZIP Code)
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO 63756E	PROJECT NO. N/A	TASK NO N/A	WORK UNIT NO. N/A

11. TITLE (Include Security Classification)
Analysis of Reservation-Based Dual-Link Networks for Real-Time Applications

12. PERSONAL AUTHOR(S)
Lui Sha, Shirish S. Sathaye, and Jay K. Strosnider

13a. TYPE OF REPORT Final	13b. TIME COVERED FROM TO	14. DATE OF REPORT (Yr., Mo., Day) June 1992	15. PAGE COUNT 28 pp.

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse of necessary and identify by block number)
coherence
dual-link networks   real-time networks
IEEE 802.6 metropolitan area network standard

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Next-generation networks are expected to support a wide variety of services. Some services such as video, voice, and plant control traffic have explicit timing requirements on a per-message basis rather than on the average. In this paper we develop a general model of reservation-based dual-link networks to support real-time communication. We examine the desirable properties of this network and the difficulties in achieving these properties. We then introduce the concept of coherence and develop a theory of coherent dual-link networks We show that a coherent dual-link network can be analyzed as though it is a centralized system. We then discuss practical considerations in implementing a dual-link network, and implications of this work to address problems observed in the IEEE 802.6 metropolitan area network standard.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED SAME AS RPT DTIC USERS ■	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution	
22a. NAME OF RESPONSIBLE INDIVIDUAL John S. Herman, Capt, USAF	22b. TELEPHONE NUMBER (Include Area Code) (412) 268-7631	22c. OFFICE SYMBOL ESD/AVS (SEI)

ABSTRACT —continued from page one, block 19