A Real-Time Scheduling Framework for Packet-Switched Networks

Shirish S. Sathaye*
Network Architecture & Performance Group
Digital Equipment Corporation
Littleton, MA 01460

Jay K. Strosnider
Electrical & Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper develops a unified framework for reasoning about timing correctness of packet-switched networks. The unification is in the form of a set of consistent scheduling models *for a variety of network architectures and protocols. The unification is important as it allows heterogeneous network types to be analyzed using a consistent methodology and facilitates scheduling over multihop networks where each link is a different type of network. To demonstrate the framework, a scheduling model for a dual-link network, IEEE 802.5 token ring, and FDDI, is presented. The use of this framework to select paths in multi-hop networks is also demonstrated.*

1 Introduction

This paper develops a unified framework for reasoning about timing correctness of packet-switched networks. The unification is in the form of a set of consistent *scheduling models* for a variety of network architectures and protocols. The scheduling model can be used to analyze the schedulability of a set of real-time messages scheduled on the network. The scope of this work spans multi-access communication networks, high-speed switch-based networks, and multihop networks built from homogeneous or heterogeneous network types. A unified analysis framework allows distinct network types to be analyzed using a consistent methodology and facilitates schedulability analysis over multihop networks where each link is a different type of network.

Kamat and Zhao [4] considered the real-time schedulability and evaluated the average performance of a priority driven token-ring protocol (IEEE 802.5) and a timed-token ring protocol (FDDI) under different operating conditions. Bandwidth reservation and channel establishment procedures to guarantee deadlines in a wide area network are discussed in [2]. Scheduling real-time traffic in multi-hop networks has been discussed in [5].

The approach in this paper is to apply real-time scheduling theory developed in the context of uniprocessor scheduling to the real-time network scheduling problem. We identify major challenges in network scheduling and discuss the differences in scheduling a processor and scheduling in a distributed environment. A scheduling framework which takes the form of a generic network scheduling model is developed. We demonstrate the application of the framework using a dual-link network, IEEE 802.5 token ring and the FDDI protocol. We also demonstrate how the framework may be used to select paths in multi-hop networks by way of an example.

2 Real-Time Network Scheduling

We consider networks with arbitrary topology. There are two principal types of entities involved, *clients* at stations in the network, and the *network* itself. Clients make use of network services by establishing *calls* across the network with clients at the destination. As we discuss later, each call at the client level is transformed by the network into a *connection.* A set of calls is assumed to exist in a network on a *path* that spans a set of network *links.*

From the viewpoint of the clients a set of *calls* $\{V_1, V_2, \ldots, V_K\}$, exists in the network. Calls are carried on network links using periodic *connections.* Each call V_k can be represented by $(S_{sk}, S_{ek}, M_k, P_k, E_k, J_k)$, where each term is defined as follows. S_{sk} is the traffic source, S_{ek} the destination station, M_k the message sent every period, P_i is the period. The source generates M_k bits of information per period P_k. E_k is the end-to-end deadline J_k is the

*This work is part of the author's Ph.D dissertation at Carnegie Mellon University, which was financially supported by Digital Equipment Corporation

maximum permissible delay (jitter) between consecutive arrivals of messages from call V_k at destination S_{ek}.

The real-time scheduling problem is to schedule a new call in the network such that it can meet its end-to-end deadline, while maintaing timing guarantees made to existing calls.

3 Real-Time Scheduling Theory

Real-time scheduling theory which begins with the pioneering work by Liu and Layland [8], has been extended by [7] in the context of fixed-priority scheduling.

Consider the scheduling of n periodic tasks $\tau_1, \tau_2, \ldots, \tau_n$ arranged in priority order. Assume that tasks were independent, each task has a deadline which is not longer than the end of its period, and the system supports as many priority levels as necessary. To analyze the schedulability of a periodic task set with deadlines not longer than the period, and scheduled by any fixed priority algorithm, we can use a test developed in [7] and based on the following observations.

- The longest response time for any task τ_i occurs at its *critical instant* which occurs when it is instantiated simultaneously with all higher priority tasks [8].
- All task deadlines will be met if the first request of each task meets its deadline.
- A task set is schedulable if the following equation holds.

$$\forall\, i, \quad 1 \le i \le n, \quad \min_{0 < t \le D_i} \sum_{j=1}^{i} \frac{C_j}{t} \left\lceil \frac{t}{T_j} \right\rceil \le 1 \tag{1}$$

In the above equation, each task τ_i is evaluated over its period, up to its deadline D_i. The summation of the workload is evaluated over the interval $(0, D_i]$. If the cumulative workload's minimum value normalized by time is not greater than unity at any time t such that $0 < t \le D_i$, then the task set is schedulable.

4 Concept of Scheduling Models

Equations 1 represents ideal scheduling equations for fixed priority scheduling in real-time systems. A *scheduling model* is an abstraction that allows one to reason about the timing correctness of a set of activities that must execute on a particular resource. A generic scheduling model can be developed based on the scheduling theory introduced in Section 3. To achieve this, the schedulability conditions must be modified as follows. C_j is replaced by $C_j + Ovh_j$ which is the total amount of time task τ_j occupies the resource every period. Ovh_j represents the additional time that must be spent on behalf of the task, and includes delays such as the time to transmit the source address, destination address, etc., in case of networks. We also need to add two new terms Ovh_{sys_i} and B_i to the schedulability conditions. Ovh_{sys_i} captures task independent system level overhead encountered on some resources. B_i captures the time task τ_i is delayed by lower priority tasks. This is also called *priority inversion* [10]. Table 1 shows a generic time-based scheduling model.

4.1 Developing Network Scheduling Models

The three main issues that distinguish network scheduling from scheduling in a centralized single resource are distributed state, multi-level scheduling, and the concurrency of packets on the network.

4.1.1 Distributed State

Scheduling traffic in a network is inherently different from scheduling in a centralized environment. In a centralized system, the scheduler knows about resource requests as soon as they arrive. In contrast, bandwidth demands on a network arrive at multiple remote locations and are not immediately observable everywhere. Distributed scheduling decisions must be made with incomplete information. From the perspective of any particular station, some bandwidth requests could be delayed and some may never be seen, depending on the relative position of the station in the network.

4.1.2 Multi-Level Scheduling

Network scheduling is a multi-level scheduling problem. For example, in a multi-access network packets that become ready to transmit at each station are first placed in a local queue using the station's scheduling policy. Stations then arbitrate for access to the medium using a global media access control policy. At every round of arbitration, each station which has a non-empty local queue informs the global scheduler

Priority	Generic Scheduling Model
Time-based fixed	$\forall\, i = 1, 2, \ldots, n$ $\min_{0 < t \leq D_i} \sum_{j=1}^{i} \frac{C_j + Ovh_j}{t} \left\lceil \frac{t}{T_j} \right\rceil + \frac{Ovh_{sys_i}}{t} + \frac{B_i}{t} \leq 1$

Table 1: Generic scheduling model summary

of its intention to transmit. The station that wins this media access arbitration transmits the packet at the head of its local queue.

A variety of different global scheduling policies have been implemented in commercial networks. In IEEE 802.6 DQDB, stations make prioritized reservations for the medium and continuously monitor usage of the medium, creating a virtual global totally ordered priority queue. In the IEEE 802.5 token ring, stations make reservations at a certain priority in a token. In the FDDI synchronous mode, stations use the medium in preallocated time division multiplexed slots controlled by a token. In general, the global scheduling policy is dictated by the network protocol. The local scheduling policy is left to the implementor.

4.1.3 Concurrency: Multiple Packets on the Network

Concurrent existence of multiple packets on networks such as IEEE 802.5 with early token release and FDDI, complicates the application of scheduling theory to networks. The generic scheduling model calculates the time at which each activity completes its execution on the resource. It accomplishes this for each task by calculating the demands for the resource by the task under consideration and all higher-priority tasks. The model assumes that only one task can use the resource at any time. To reconcile this, the traditional notion of schedulability needs to be extended. In a network it is useful to consider the notion of *transmission schedulability* [9]. A connection is said to be transmission schedulable (*t-schedulable*) if its messages are transmitted before their deadlines. The maximum end-to-end latency on the link is given by :

$$\text{Max. Latency} = \text{Xmission Deadline} + \text{Prop. Delay} \quad (2)$$

The end-to-end deadline of the message on the link is satisfied if the following relation holds.

$$\text{Maximum Latency} \leq E_i \quad (3)$$

Where E_i is the end-to-end deadline. For network protocols which permit multiple packets on the network, the scheduling models check the *t-schedulability* of a set of connections on the network.

5 Degree of Schedulable Saturation

Any successful analysis framework requires quantitative criteria for evaluating the performance of different design approaches. We propose a metric called the *degree of schedulable saturation* metric [9] that is consistent with our scheduling framework. It represents the degree to which the system is saturated from a schedulability viewpoint. A smaller S_{max} indicates greater remaining high-priority schedulable capacity. A particular scheduling situation is better if it results in a smaller S_{max}. We define S_{max} as follows:

$$S_{max} = \max_{1 \leq i \leq n} Saturation_i \quad (4)$$

$$Saturation_i = \min_{0 < t \leq D_i} W_i(t)/t \quad (5)$$

where $W_i(t)$ is the bandwidth demands up to time t by connection τ_i and higher-priority connections. $W_i(t)$ is given by:

$$W_i(t) = \sum_{j=1}^{i} (C_j + Ovh_j)\lceil t/T_j \rceil + Ovh_{sys_i} + B_i \quad (6)$$

Given that $S_{max} = Saturation_i$ for some $1 \leq i \leq n$, τ_i is called the *limiting connection*. If S_{max} is unity no new connections with priority greater than or equal to that of the limiting connection can be scheduled. If

S_{max} is greater than unity the system is unschedulable. As shown in [9] $(1-S_{max})$ can be considered to be the amount of high-priority work that can be added to the system per unit time without missing a deadline. S_{max} is a measure of remaining high-priority schedulable capacity. S_{max} is monotonically non-decreasing when either connection's C's increase or the number of connections to be scheduled on the system increases. S_{max} also increases when system overhead or blocking increases.

6 Scheduling Model for a Single Link

The simplest schedulable resource in a network is a single *link*. The link may be a multi-access communication network such as the IEEE 802.5 token ring, or it may be part of a switch based network. The network may either support fixed size or variable size packets. Without loss of generality we can assume that the maximum size packet on the network is denoted by P_{max}.

6.1 Sources of Overhead

Each connection τ_j scheduled on this link needs to transmit a message for C_j units of time every period. A certain amount of overhead is incurred in the transmission of this message. In the context of a network link, the components of the Ovh_j term of the generic scheduling model include:

- *Delay in transmitting packet-encapsulation:* Before transmission, the information is packetized by encapsulating portions of the information with control headers and trailers.
- *Delay in waiting for packet reception acknowledgment:* If the link protocol does not permit concurrent existence of packets from the same connection, it cannot transmit a new packet until it has determined that the previously transmitted packet has reached its destination. An example of such a protocol is the original IEEE 802.5 token ring protocol. Let us denote this delay C_{ack}.

The term Ovh_j in the generic scheduling model, which represents the total overhead in transmitting C_j units of information per period, consists of the sum of the header (C_h), trailer (C_t), and the acknowledgment (C_{ack}), multiplied by the number of packets that are needed to transmit C_j units of information. Ovh_j can be written as:

$$Ovh_j = (C_h + C_t + C_{ack}) \left\lceil \frac{C_j}{P_{max} - (C_h + C_t)} \right\rceil \quad (7)$$

Several protocols such as FDDI, IEEE 802.6, and switch-based network link protocols, permit transmission of a new packet before a previous packet has been acknowledged. In these cases C_{ack} is zero. The scheduling model for these networks is a t-schedulability model which checks only whether messages can be transmitted by their deadline. End-to-end latency of the message on the link can be determined using Equation 2. Satisfaction of the end-to-end deadline can be determined using Equation 3.

There are two sources of system-level overhead in a network. We classify Ovh_{sys_i} into the contributions due to the media access protocol, and the contributions due to unsynchronized global clocks:

- *MAC protocol penalty O_{MAC}:* Sometimes, characteristics of the protocol can cause a connection-independent overhead.
- *Unsynchronized clocks penalty O_{clock} :* If the clock in the destination station is "ahead" by an amount Δ, then messages must be delivered with a shortened deadline. This can be treated as a connection-independent overhead.

As an example of O_{MAC}, consider an FDDI network operating in synchronous mode. In this protocol, bandwidth is preallocated to stations in a time division multiplexed manner. A station may hold the token for up to its allocated bandwidth per cycle. The station may divide its assigned bandwidth among its connections using any scheduling algorithm. This creation of a TDM cycle among stations contributes to a connection independent overhead in the scheduling model. If station S_k is allocated a bandwidth H_k per cycle T_{cycle}, then O_{MAC} at any time t can be expressed as:

$$O_{MAC} = \left\lceil \frac{t}{T_{cycle}} \right\rceil (T_{cycle} - H_k) \quad (8)$$

There is also a penalty due to unsynchronized clocks. It is well known that no algorithm can exactly synchronize clocks of stations in a distributed system [6].

Therefore including the penalty due to unsynchronized clocks we get:

$$Ovh_{sys_i} = O_{MAC} + O_{clock} \quad (9)$$

6.2 Sources of Blocking

Blocking can occur in a network due to one or more of the following:

- *Resource is not fully preemptable:* Connections in a network are preemptable only at packet boundaries. A high-priority packet must wait until the end of transmission of a lower-priority packet. The maximum amount of blocking due to non-preemptability of packets is P_{max}, the maximum packet size (expressed in time).

- *Global priority scheduling is imperfect:* The arrival of a high-priority packet is not immediately detected by the scheduler. This imperfect arbitration can lead to blocking, which may be denoted as B_{gs}.

- *Priority levels are insufficient:* Blocking can occur when a set of connections with a large number of *natural priorities* is scheduled on a network that supports a smaller number of priority levels. The natural-priority level is the priority that would have been assigned to the connection on a system with sufficient priority levels. Multi-access communication networks typically support very few priority levels. For example, IEEE 802.5 token ring supports only eight priority levels. Blocking due to insufficient priority levels can be denoted as B_l.

Considering the blocking contributions of non-preemptability of the network, effects of global priority arbitration, and limited priority levels, the B_i term of the generic scheduling model can be written as:

$$B_i = P_{max} + B_{gs} + B_l \qquad (10)$$

The scheduling model of a single network link is summarized in Table 2. As stated earlier, if the network permits concurrency concurrency, the model is a t-schedulability model, and the end-to-end deadline condition must also be checked.

6.3 Example: Link Scheduling

In this Section we will show how the framework developed here can be used to analyze three very different network types:

6.3.1 Dual-Link Networks

We briefly discuss the operation of the dual-link network. The IEEE 802.6 DQDB MAC operating in opposite directions. The links may be referred to as Flink and Rlink respectively as shown in Figure 1.

Fixed-length slots are generated by slot generators of the corresponding links. Each station is able to transmit and receive messages on both links. Reservation for a slot on the Flink is made on the Rlink via a **request** and vice versa.

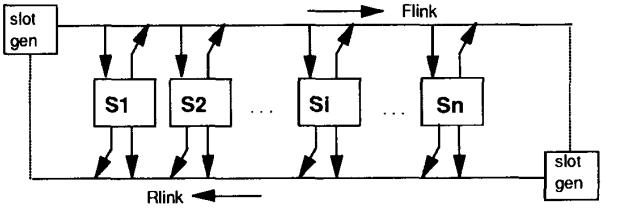

Figure 1: IEEE 802.6 DQDB Network

The operation of the protocol is based on a single busy bit, indicating whether the slot is used or free, and a request bit per slot for each priority level. Four priority levels are supported. Each priority level represents a separate access queue. A station wishing to transmit at a certain priority on Flink, issues a request in a slot on Rlink by setting the proper request bit. It also places its own request into its access queue at the correct priority. Each station on seeing a request, enqueues it in its access queue at the correct priority. Every station on seeing a free slot discards the top request from its highest priority non-empty access queue, because the slot has been previously reserved by another station. If the top request is the station's request then it transmits in the slot on the Flink in addition to removing the request from its access queue.

The current IEEE 802.6 protocol does not have adequate mechanisms to ensure correct operation in a real-time environment [9], and it exhibits unpredictable behavior under certain conditions [11]. In [9] a dual-link protocol that was shown to be predictable was presented. The protocol is based on the IEEE 802.6 architecture with the modifications that: (i) requests on the Rlink can be preempted by stations based on their priority, and (ii) Stations can have multiple outstanding requests. Since the dual-link protocol was explicitly designed to be predictable, its scheduling model is simple. Let d_i be the propagation delay between the source of connection τ_i and the Flink slot generator.

Sources of Overhead

The term Ovh_j in the generic scheduling model represents the total overhead in transmitting C_j units of information per period. This consists of the sum of the header (C_h) multiplied by the number of packets that are needed to transmit C_j units of information. Ovh_j can be written as follows:

Scheduling Model

$\underset{1 \le i \le n}{max} \quad \underset{0 < t \le D_i}{min} \sum_{j=1}^{i} \frac{C_j + Ovh_j}{t} \left\lceil \frac{t}{T_j} \right\rceil + \frac{Ovh_{sys}}{t} + \frac{B_i}{t} \le 1$ if t-schedulability model, check $E_i \ge D_i$ + Propagation Delay

Model Parameters

Ovh_j	$(C_h + C_t + C_{ack}) \left\lceil \frac{C_j}{P_{max} - (C_h + C_t)} \right\rceil$
Ovh_{sys}	$O_{MAC} + O_{clock}$
B_i	$P_{max_k} + B_{gs} + B_l$

Table 2: Link scheduling model summary

$$Ovh_j = C_h \left\lceil \frac{C_j}{C_{packet} - C_h} \right\rceil \tag{11}$$

Sources of Blocking

We have shown that each packet of τ_i is transmitted in its *assigned* slot after an initial delay of $2d_i$. Due to the flow control protocol, the first packet of the connection has to wait for $2d_i$ after making a request before it can transmit. If the connection is schedulable, it will be able to transmit C_j units of information every period. Each message will be first delayed at least for $2d_i$ before it is transmitted, irrespective of its priority. During this time, lower priority transmissions can occur. Hence each message suffers a blocking of $2d_i$.

Hence we can write the scheduling model of a dual-link network as follows:

$$\forall\, i = 1, 2, \ldots, n \quad \underset{0 \le t \le D_i}{min} \sum_{j=1}^{i} \frac{C_j + Ovh_j}{t} \left\lceil \frac{t}{T_j} \right\rceil + \frac{2d_i}{t} \le 1 \tag{12}$$

Where Ovh_j is given by Equation 11. As stated before this model gives the worst case time at which each message from a connection τ_i completes transmission and checks if one message can be transmitted per period of the connection. Hence it checks if the set of connections is t-schedulable.

6.3.2 IEEE 802.5 Token Ring Scheduling Model

Scheduling models for two options of the IEEE 802.5 protocols are now presented. The conventional token release (CTR) protocol is the one in which the source station must wait for the claimed token to arrive before it can generate a new token. In the early token release (CTR) protocol the source station can generate a free token as soon as packet transmission ends. We demonstrate the usefulness of the model, to select between the two protocols for a given message set. The models are and described in detail in [9].

Our model can be used to compare CTR and ETR scheduling calculate S_{max} for a particular connection set at different operating regions. The graphs in Figure 2 and Figure 3 plot S_{max} as a function of walk-time W_T for different P_{max} values for both CTR and ETR. The performance of the protocols depends both on network parameters such as W_T and P_{max}, and on the characteristics of the connection set. For a given connection-set, walk-time, and maximum packet size the model can therefore be used to select between the CTR and ETR protocol.

6.3.3 FDDI Scheduling Model

We demonstrate the use of a scheduling model for FDDI in determining bandwidth allocation to sta-

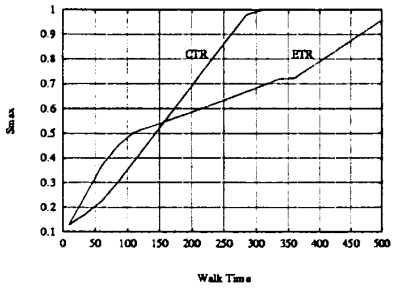

Figure 2: S_{max} vs. walk-time for $P_{max} = 75$, CTR and ETR protocol

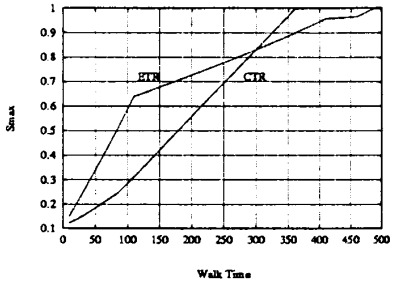

Figure 3: S_{max} vs. walk-time for $P_{max} = 100$, CTR and ETR protocol

tions. The scheduling model is summarized in Table 3.

The FDDI MAC protocol does not provide rules or guidelines governing allocation of synchronous bandwidth among stations, and bandwidth allocation algorithms have recently received much attention. A normalized proportional allocation scheme has been proposed in by Agrawal *et al.* [1] that allocates each station a bandwidth proportional to its relative network utilization. Under this scheme the allocation to station S_k is given by :

$$H_k = \frac{U_k}{U_{net}}(TTRT - W_T) \qquad (13)$$

where H_k is the bandwidth allocated to station S_k. U_k is the network utilization of station S_k and the network utilization $U_{net} = \sum_{i=1}^{n} U_i$. $TTRT$ is the target token rotation time and W_T is the walk time (the token rotation time when the network is idle). Agrawal *et al* have shown that provided this allocation scheme is used, message deadlines can be guaranteed

Scheduling Model

$$\begin{array}{c} max \quad max \quad min \\ 1 \le k \le n \quad 1 \le i \le m_k \quad 0 < t \le D_{ik} \\ \sum_{j=1}^{i} \frac{C_{jk}+Ovh_{jk}}{t} \left\lceil \frac{t}{T_{jk}} \right\rceil + \frac{Ovh_{sys_k}}{t} + \frac{B_i}{t} \le 1 \end{array}$$

Model Parameters

Ovh_{jk}	$\left\lceil \frac{C_{jk}}{P_{max_k} - C_{enc}} \right\rceil C_{enc}$
Ovh_{sys_k}	$(TTRT - H_k)\left\lceil \frac{t}{TTRT} \right\rceil$
B_i	P_{max_k}

Table 3: FDDI scheduling model summary

provided the network utilization is below 33%, and that stations service packets in priority order.

We have modeled the FDDI network by developing a scheduling model for each station in the network. Each station in the network has a distinct S_{max}. It is desirable to allocate bandwidth such that each station has approximately the same S_{max}, since this results in the smallest schedulability saturation across the network. Our model can be used to iteratively solve for a bandwidth allocation that minimizes S_{max} since a smaller S_{max} implies a smaller degree of schedulability saturation.

For example in a network with three stations, Figure 4 shows S_{max} as a function of maximum packet size using normalized proportional bandwidth allocation. By reducing the allocations to stations S_1 and S_2 and increase the allocation to station S_3 we tend to equalize the values of S_{max} for the three stations as shown in Figure 5. This iterative bandwidth-allocation heuristic (termed S_{max} driven allocation) is preferable to the original normalized proportional allocation. In the normalized proportional allocation, since station S_3 is very close to being unschedulable, even a small increase in size of any message from S_3 may make the network unschedulable. The new allocation scheme reduces the saturation of station S_3.

In an FDDI network, stations negotiate a target

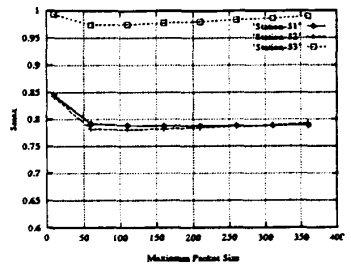

Figure 4: S_{max} vs P_{max}: Normalized prop. allocation

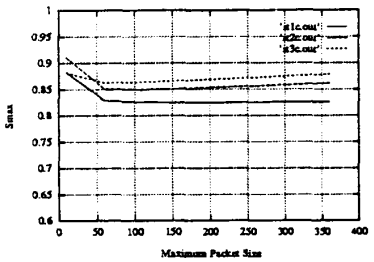

Figure 5: S_{max} vs. P_{max}:S_{max} driven allocation

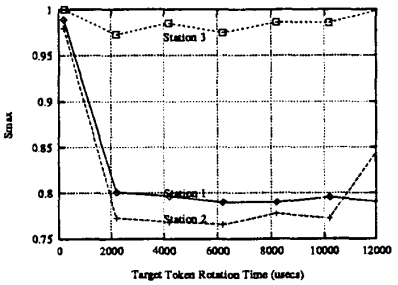

Figure 6: S_{max} vs. $TTRT$: Normalized prop. allocation

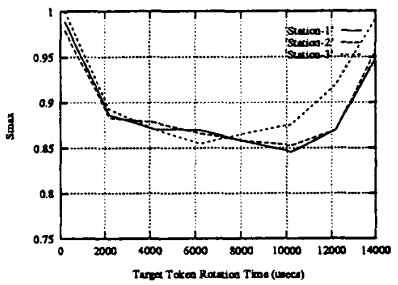

Figure 7: S_{max} vs. $TTRT$: S_{max} driven allocation

token rotation time. The model may also be used to search the design space for the values of TTRT to allocate bandwidth such that the equalized value of S_{max} is minimized. For the three station network of the previous example, the first graph in Figure 6 plots S_{max} as a function of $TTRT$ with normalized proportional bandwidth allocation. Figure 7 plots S_{max} as a function of $TTRT$ for the connections at three stations in an FDDI network. Bandwidth is allocated to stations using S_{max} driven bandwidth allocation. As expected, S_{max} driven allocation lowers the saturation of station S_3. Observe that the S_{max} curve in both figures remains flat for a wide range of values. Other studies have shown that for non real-time traffic a larger value of $TTRT$ is desirable for higher throughput.

7 Multi-Hop Scheduling

Scheduling a call over a path in a multi-hop network is similar to scheduling a task over a set of serially connected resources. The end-to-end deadline of the call must be partitioned, and intermediate deadlines must be assigned to each link. The call is schedulable if it is schedulable at each link.

When packets from a call are scheduled on multiple links in series, they may arrive at the next link before their deadline on the current link. If we schedule packets upon their arrival, calls lose their periodic characteristics. This undesirable effect can be controlled by a simple rule: messages from a call that arrive in a given period become eligible for transmission only at the beginning of a new period. This rule has been called *double buffering* or *stop-and-go queueing* [3].

Let the total number of links in the network be N_L. Connections (τ_{ij}) on each link in the network can be specified using a double subscript notation where the first subscript indexes connections on the same link and the second subscript indexes on different links. Hence, the connections on the network can be described as:

$$\forall\, i,\ 1 \le i \le n_j \quad \forall\, j,\ 1 \le j \le N_L \qquad \tau_{ij} : (C_{ij}, T_{ij}, D_{ij}) \tag{14}$$

where n_j is the number of connections on link j. Consider a particular link j on a path P of a call V_k mapped to connection τ_{ij}^k on link j, and let a set of real-time connections $\{\tau_{1j}, \ldots, \tau_{n_j j}\}$ exist on that link.

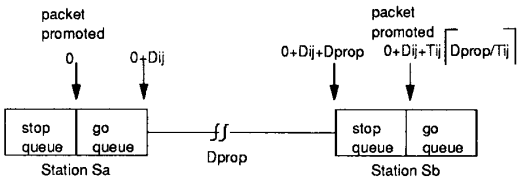

Figure 8: Delay between consecutive promotions

Each distinct period is assigned a pair of queues, the *stop-queue* and the *go-queue*. Packets from the incoming link are stored in the stop-queue. Packets in the go-queue are eligible for transmission. The transferring of packets for a connection from its *stop-queue* to its *go-queue* is called *promotion*. A connection's packets are promoted at the end of its period. Traffic smoothness is maintained by promoting at most C_{ij}^k packets from τ_{ij}^k per promotion. Packets in go-queues with smaller periods are serviced before those with larger periods. The end-to-end delay for a periodic message depends directly on the time between successive promotions. Assuming that the time spent by a packet of connection τ_{ij}^k in each go-queue can be bounded by D_{ij}^k, the maximum delay between successive promotions can be computed as follows.

Consider any packet of connection τ_{ij}^k in the stop-queue of station S_a in Figure 8. Let it be promoted to the go-queue of S_a at time 0. It will be transmitted from S_a, at a time no greater than $0+D_{ij}^k$. The packet will arrive at S_b's stop-queue at $0+D_{ij}^k+D_{prop_j}$, where D_{prop_j} is the propagation delay of link j. D_{prop_j} is defined as the difference in the time at which a packet is sent from one end of the link, and time that the packet is received at the other end of the link, time being measured independently at each end. The packet waits in S_b's stop-queue until it is promoted. This time can be T_{ij} in the worst case. The summation of the propagation delay D_{prop_j} and the time spent in station S_b's stop queue is given by $T_{ij}^k\lceil D_{prop_j}/T_{ij}^k\rceil$. The ceiling function accounts for the fact that the propagation delay may not be an integral multiple of the period and hence the packet may arrive at station S_b's stop queue after S_b has just done a promotion, delaying the packet for a complete period. The time between promotions is given by:

$$\text{Time between Promotions} = D_{ij} + T_{ij}\left\lceil \frac{D_{prop_j}}{T_{ij}^k} \right\rceil \tag{15}$$

The end-to-end delay for call V_k on a path with h_k links is given by:

$$\text{End-to-end Latency} = h_k D_{ij}^k + \sum_{j=1}^{h_k} T_{ij}^k \left\lceil \frac{D_{prop_j}}{T_{ij}^k} \right\rceil \tag{16}$$

and the condition for messages of the call meeting their end-to-end latency requirement (deadline E_k) is

$$\text{End-to-end Latency} \leq E_k \tag{17}$$

The latency is the time between promotions multiplied by L, the number of links on the path. This is a simplified version of the expression for end-to-end delay from [3]. The stop-and-go queueing technique, combined with the assumption that the connection is t-schedulable, guarantees that the destination receives C_i units of information every period.

In the above discussion we assumed that each link is t-schedulable. Therefore the schedulability condition of a call V_k on a path of length h_k in a multi-hop network can be considered to be a combination of the t-schedulability model for connections on each link in the path, and the end-to-end schedulability checks developed above. Consider a path P of length h_k which is the set of all the links in the path. Then connections on the path are schedulable if the following conditions holds.

$$\forall\, j \in P \quad \forall\, i = 1, 2, \ldots, n_j$$
$$\min_{0 < t \leq D_{ij}} \sum_{m=1}^{i} \frac{C_{mj}+Ovh_{mj}}{t}\left\lceil \frac{t}{T_{mj}} \right\rceil + \frac{Ovh_{sys}}{t}\frac{B_{ij}}{t} \leq 1 \tag{18}$$

j refers to a link in the path and n_j is the number of connections on link j. If the above condition holds then a call V_k that is scheduled on the path is t-schedulable on each link. In combination with end-to-end conditions developed above for each call, this is the basis of a test to determine whether a new call should be accepted by the network.

7.1 Example: Multi-Hop Scheduling

Figure 9 shows an FDDI network, an IEEE 802.5 network operating at in CTR mode, and other networks that have been shown as links to keep this example simple. These links L_1 through L_7 can be different network types for which we have a scheduling model in the framework presented in this paper. Let IEEE 802.5 network have an S_{max} of 0.2 for the connection set. The values of S_{max} for the other links in the network are shown in Figure 9 (for now, ignore the numbers in parentheses). Let the S_{max} of the FDDI network be 0.85.

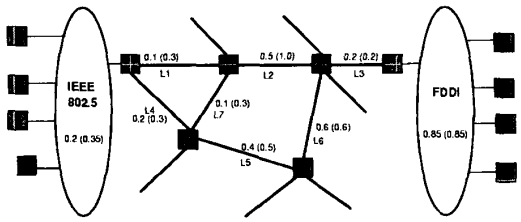

Figure 9: Example illustrating multi-hop scheduling & routing

Let a new connection need to be established between a station on the IEEE 802.5 network and a station on the FDDI network. The numbers in parentheses indicate the values of S_{max} of the links assuming that this connection is established. For example the establishment of this connection would increase the S_{max} of the 802.5 network from 0.2 to 0.35. Observe that the values of S_{max} either stay the same or increase, since S_{max} is monotonically non-decreasing with increased load, [9].

There are two paths that are schedulable between the source and destination: path
802.5 $\rightarrow L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow$ FDDI has a cost of 1.0 and
802.5 $\rightarrow L_4 \rightarrow L_5 \rightarrow L_6 \rightarrow L_3 \rightarrow$ FDDI has a cost of 0.85

We can consider the *cost* of a path to be the maximum S_{max} of all links in the path [9]. Hence a routing algorithm that chooses a minimum cost path will avoid overloaded links in the network. Such a routing algorithm will then choose the path 802.5 $\rightarrow L_4 \rightarrow L_5 \rightarrow L_6 \rightarrow L_3 \rightarrow$ FDDI between the source and destination stations to establish the new connection. The S_{max} values of the links in the path are updated.

8 Conclusion

This paper developed a unified framework for reasoning about timing correctness of packet-switched networks in the form of consistent *scheduling models* for a variety of network protocols. The scope of the work spans multi-access communication networks, high-speed switch-based networks, and multi-hop networks built from homogeneous or heterogeneous network types. The unification is important as it allows the heterogeneous network types to be analyzed using a consistent methodology and facilitates scheduling over multihop networks where each link is a different type of network.

References

[1] G. Agrawal, B. Chen, W. Zhao, and S. Davari. Guaranteeing synchronous message deadlines in high speed token ring networks with timed token protocol. *Proceedings of IEEE International Conference on Distributed Computing Systems*, 1992.

[2] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[3] S. Golestani. A stop-and-go queueing framework for congestion management. *Proceedings of SIGCOMM'90*, June 1990.

[4] S. Kamat and W. Zhao. Real-time schedulability of two token ring protocols. *13th International Conference on Distributed Computing Systems*, pages 347–355, May 1993.

[5] D. Kandlur, K. Shin, and D. Ferrari. Real-time communication in multi-hop networks. *11th International Conference on Distributed Computing Systems*, pages 300–307, May 1991.

[6] H. Kopetz and W. Ochsenreiter. Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers*, C-36(8):933–940, August 1987.

[7] J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. *10th IEEE Real Time Systems symposium*, 1989.

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

[9] S. Sathaye. *Scheduling Real-Time Traffic in Packet-Switched Networks.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, June 1993.

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

[11] 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.