Exact and Efficient Analysis of Schedulability in Fixed-Packet Networks: A Generic Approach

Sergey Gorinsky* Sanjoy Baruah*† Thomas J. Marlowe*‡ Alexander D. Stoyenko*

Real-Time Computing Laboratory
Department of Computer and Information Science
New Jersey Institute of Technology
Newark, NJ 07102, USA

Abstract

A general model for traffic flows on packet-switched, virtual-circuit based, fixed-packet networks is introduced, and an exact schedulability test is obtained for systems of such flows. Rules are derived that make the evaluation of this schedulability test feasible and efficient under certain circumstances. The practical relevance of this approach is demonstrated by applying it to a number of standard traffic models.

1 Introduction

Rate-based scheduling strategies are of particular interest in the design and implementation of virtual-circuit packet-switched computer networks that provide support for multimedia applications [9, 12, 14, 15, 16, 19, 21, 22]. It is assumed that the network traffic generated by each virtual-circuit connection satisfies certain pre-specified *rate constraints*, and that the scheduling strategy at the network nodes aims to provide each connection with certain *performance guarantees*, typically expressed in terms of (end-to-end) delay, delay jitter, throughput, et cetera.

Traditionally, the study of computer network performance has had a strong stochastic flavor, and sophisticated queueing-theoretic models have been developed to characterize network traffic, and measure performance under various traffic loads, service disciplines, etc. However, performance guarantees obtained through such modeling are necessarily probabilistic, and hence of limited usefulness when *deterministic* performance guarantees are required. There has therefore recently been considerable interest in deterministic modeling of network traffic, and design and implementation of network service disciplines that work well with such deterministic traffic.

A request for a new connection is admitted by the network if and only if the network possesses adequate resources to guarantee its desired level of performance — this process of determining whether to admit a request for a new connection is referred to as a *connection admission procedure.* A cornerstone of any connection admission procedure is *schedulability analysis:* given a set of connections that share a network link, can all packets be delivered before the requested delay bounds? The subject of this paper is the design of exact and computationally efficient schedulability tests for packet-switched virtual-circuit fixed-packet networks.

Related work. Schedulability analysis is greatly affected by two factors: the service discipline at network nodes and the traffic model employed to characterize data streams. First Come First Served (FCFS) has traditionally been a scheduling discipline of choice in ATM networks. Cruz [4, 5] has obtained delay bounds in FCFS networks using the leaky bucket model for traffic characterization. While simple to implement, FCFS lacks flexibility; the policy does not discriminate between traffic from different connections and therefore is not able to provide different levels of performance guarantees. Recently, new scheduling disciplines have been investigated: VirtualClock [22], Stop-and-Go [12], and Generalized Processor Sharing (GPS) and its packet-based version — Weighted Fair Queueing (WFQ) also known as Packet-by-packet

*Real-Time Computing Laboratory, Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ 07102, USA. Research is supported in part by the U.S. ONR Grant N00014-93-1-1047 and by the NSF Grant CCR-9402827.

†Department of Computer Science and Electrical Engineering, The University of Vermont, Burlington, VT 05405, USA. Research is supported in part by the NSF Grant CCR-9596282. Dr. Baruah is on leave from the Real-Time Computing Laboratory at NJIT.

‡Department of Mathematics and Computer Science, Seton Hall University, South Orange, NJ 07079, USA.

Generalized Processor Sharing (PGPS) [7, 19, 20]. The latter policy can take into account delay dependencies along the route of a connection and may provide smaller end-to-end delays than those obtained by summing up the worst case delays in each node.

Since deadline-based service disciplines, such as the Earliest Deadline First (EDF), have been proven optimal in the context of uniprocessor scheduling [8], they have captured the attention of networking researchers and are being used in various research studies as a traffic scheduling policy. Even though the derived performance guarantees are based on per-node analysis, it was shown by Georgiadis et al [11] that when accompanied with traffic reshaping inside the network, EDF outperforms GPS in terms of end-to-end delay bounds, buffer requirements, and implementation simplicity. This makes EDF an efficient service discipline to employ at network nodes. Ferrari and Verma [9] attempted to find schedulability conditions for the case when data streams abide by the Tenet model, when the minimum inter-arrival time for any connection is not less than the summation of packet transmission times over all connections. Zheng and Shin [23] derived necessary and sufficient feasibility conditions for the sporadic traffic model; Baruah, Mok, and Rosier [1] had earlier derived very similar results in the context of hard-real-time scheduling. Georgiadis, Guerin, and Parekh [10] provided necessary and sufficient schedulability conditions for the leaky bucket model. Liebeherr et al [16, 21] have provided similar analysis for more general constraints: the maximum traffic from a connection is characterized by a concave rate-controlling function.

Although the model of Liebeherr et al permits the representation of a large class of traffic specifications, traditional real-time streams of data, as well as traffic generated by many multimedia applications, cannot be accurately characterized by concave functions. Besides, the proposed testing bound is too pessimistic. For example, when the traffic is such that the link may be busy for an unbounded amount of time, the schedulability test may not terminate. Thus, the approach may fail even if the requested performance guarantees can in principle be provided.

This research. We study the problem of schedulability analysis within a very general framework, encompassing all of the work described above. Our approach is as follows:

- We define some very general constraints on traffic (Definition 1), which we refer to as the *Time-Independence Property* and the *Connection-Independence Property*. These properties are satisfied by the sporadic, Tenet, and leaky bucket models of traffic constraints and by all concave rate-controlling functions, as well as by significantly more general models of traffic constraints.
- For traffic constraints satisfying these properties (*well-formed constraints*), we define the notion of *extreme sets* (Section 4), and use this notion to derive necessary and sufficient conditions for schedulability in systems where all traffic satisfies such constraints.
- We identify guidelines (*TLTF*, *EQ*, *WIB*, and *Skip Rules* in Section 6) which assist in the efficient evaluation of these schedulability conditions. For particular kinds of traffic constraints, these rules can often be used to derive efficient schedulability tests.

This paper is organized as follows. Section 2 describes the network model and gives some definitions. Section 3 surveys existing traffic models and presents our generic traffic constraints. Section 4 introduces the notion of extreme sets. Section 5 discusses scheduling issues, and derives schedulability conditions. Section 6 provides general guidelines for identifying the smallest *testing set* and demonstrates these guidelines by examples of some specific traffic models. Finally, the results are summarized in Section 7.

2 The Network Model

In this study we consider packet-switched virtual-circuit fixed-packet networks of an arbitrary topology. Our model can be viewed as a generalization of ATM networks. Network nodes (switches) are interconnected by links. Point-to-point communication is supported by means of connections: data are transferred from a source node to a destination node along some fixed route in the network. Connection establishment takes place before transmission of user's data and succeeds only if requested performance guarantees can be provided. Traffic entering a network switch (from an input link or from an application process at the source node) is demultiplexed, and the packets are transferred to the buffers of appropriate output links (or application processes at the destination node). Every output link buffer is associated with a scheduler that decides in which order packets are sent over the link. As a general rule, the time to transfer a packet from an input link to an output link buffer (switching delay) is usually negligible in comparison with waiting time in the buffer (queueing delay) and transmission time of the packet (transmission delay) [3, 23]. In this study, we therefore assume that delay experienced by a packet in a switch comprises only queueing and transmission delays.

The basic unit of data transfer is the *packet*, and all packets are of the same size. A *message* is an ordered sequence of packets; different messages may contain different numbers of packets. Throughout this paper, we consider a model of time in which time is divided into discrete and indivisible fixed-size *time slots*. Each slot is equal to the time taken by one packet to be transmitted over a link[1]. For instance, if packets are 53 bytes long and the link bandwidth is 155 Mb/s, then the slot size equals to 2.7 μs. Messages are assumed to arrive instantaneously, and this arrival is aligned to the start of a time slot[2].

Let us define a *connection system* as the set $\{C_1, \ldots, C_n\}$ of all connections sending data over an output link of a network switch. Each connection C_i, $1 \leq i \leq n$, is characterized by an ordered pair (Z_i, d_i) where Z_i specifies a *traffic constraint* on the message arrival pattern, and d_i denotes a *delay bound*. Every message having arrived from connection C_i at time t is called an *arrival* and has to be served before time $t+d_i$ which is called the *deadline* of the arrival. Throughout the paper, without loss of generality we assume that $d_1 \leq d_2 \leq \ldots \leq d_{n-1} \leq d_n$.

A set R of arrivals is said to be *schedulable* iff it is possible to service all the arrivals of R without missing any of their deadlines. R is *legal* iff all the constraints Z_i are satisfied. The connection system is *schedulable* iff each legal set of arrivals is schedulable.

3 Traffic Constraints

In order to be able to provide useful performance guarantees, it is necessary that the traffic admitted along each connection be constrained in some manner. These constraints are formulated in terms of traffic models which can be categorized as *continuous* or *discrete*.

Traffic specification through continuous models simplifies network analysis but such models make the unrealistic assumptions that the traffic can be split into infinitesimally small packets and that multiple connections can simultaneously transmit data over the same link. As a result of these pessimistic assumptions, real traffic cannot be precisely described by continuous models; a connection admission request may be rejected even if its acceptance is safe for preserving performance guarantees in the actual (discrete-packet) network.

[1]Note that the slot size depends on the speed of data transmission and may vary for different links.

[2]In practice, this means that a message is considered to have arrived at the start of the slot following the instant that its last bit arrives. As a result, a packet can experience an additional (up to one slot) *alignment* delay in a node.

Discrete models take into account finite sizes of packets and can exactly describe network traffic. These models allow us to derive necessary and sufficient schedulability conditions for real traffic. On the other hand, performance analysis becomes more complicated.

Several kinds of constraints have been considered in the literature — we describe some of them briefly below.

The sporadic model of [1, 2, 17, 23] is a discrete traffic model. The description of a data stream is done by specifying pair of parameters (p, s), where p is the minimum message inter-arrival time and s is the maximum message size. If traffic conforms to the sporadic model, the amount of data transferred during an interval of length t is no larger than:

$$\mathsf{arrival}_s(t) = s\left\lceil \frac{t}{p} \right\rceil^+$$

where function $\lceil x \rceil^+ = n$ if $n-1 \leq x < n$ for $n = 1, 2, \ldots$, and $\lceil x \rceil^+ = 0$ when $x < 0$.

A continuous version of the model limits maximum transmission rate over an arbitrary time interval by parameter ρ. As a result, the quantity of traffic having arrived during any interval of length $t \geq 0$ can not exceed:

$$\mathsf{arrival}_s^{cont}(t) = \rho t$$

The Tenet model [9] is a discrete traffic model defined by parameters (x_{min}, x_{ave}, I, s). A data stream complies with the model if: (1) the maximum size of a message is not larger than s; (2) the message inter-arrival times are larger or equal to x_{min}; (3) over any interval of (fixed) length I, the average message inter-arrival time is not smaller than x_{ave}. Since $\frac{I}{x_{ave}}$ refers to the maximum number of messages having arrived during any interval $[t_1, t_2)$ where $t_2 - t_1 = I$, without loss of generality we may assume that $\frac{I}{x_{ave}}$ is an integer. The maximum amount of data having arrived during any period of length $t \geq 0$ is at most:

$$\begin{aligned}\mathsf{arrival}_T(t) \quad = \quad & s\Big(\left\lfloor \frac{t}{I} \right\rfloor \frac{I}{x_{ave}} \\ & + \min\Big\{\left\lceil \Big(\frac{t}{I} - \left\lfloor \frac{t}{I} \right\rfloor\Big)\frac{I}{x_{min}} \right\rceil^+, \frac{I}{x_{ave}}\Big\}\Big),\end{aligned}$$

where the first term corresponds to the maximum number of messages in complete intervals of size I, and the second to the maximum number in the current fractional period.

A continuous version of the Tenet model can be defined through parameters $(\rho_{max}, \rho_{ave}, I)$. A traffic stream satisfying the model should have the following properties: (1) the maximum arrival rate is ρ_{max}; (2) over any interval of length I, the average arrival rate is not larger than ρ_{ave}. The worst-case amount of traffic during any interval of duration $t \geq 0$ equals:

$$\text{arrival}_T^{cont}(t) = \rho_{ave}\left\lfloor\frac{t}{I}\right\rfloor I + \min\{\rho_{max}(\frac{t}{I} - \left\lfloor\frac{t}{I}\right\rfloor)I, \rho_{ave}I\}$$

Note that the sporadic model is a subcase of the Tenet model when $x_{min} = x_{ave} = p$ (or $\rho_{max} = \rho_{ave} = \rho$ for the continuous versions).

The leaky bucket model [4, 5, 6, 16] is a continuous traffic model and incorporates burstiness of data streams through parameter σ. This parameter denotes the amount of the maximum (possibly instantaneous) burst over the maximum average rate ρ. The quantity of traffic having arrived during any period of length $t \geq 0$ is not allowed to be larger than:

$$\text{arrival}_l^{cont}(t) = \sigma + \rho t$$

A discrete version of the leaky bucket model is characterized by three parameters (σ, p, s). σ packets are allowed to arrive in addition to the maximum average rate represented by s (the maximum size of a message) and p (the minimum message inter-arrival time). The amount of data received during any time interval of length $t \geq 0$ is limited by:

$$\text{arrival}_l(t) = \sigma + s\left\lceil\frac{t}{p}\right\rceil^+$$

When bursts are not allowed ($\sigma = 0$), the model reduces to the sporadic model.

While these constraints may all be a bit different from one another, they each attempt to do the same thing: restrict the rate at which traffic can arrive along any connection. In this work, we consider not only such rate-restricting traffic constraints, but also constraints that may specify to a certain degree the exact *pattern* of arrivals:

Definition 1. A traffic constraint Z_i is defined to be *well-formed* if all legal sets of arrivals under constraint Z_i possess the following two properties:

(1) *Time-Independence Property:* If R is a legal arrival set and set R′ can be derived from R by adding some integer δ to the arrival times of all messages, then R′ is also a legal set of arrivals.

(2) *Connection-Independence Property:* there is no correlation between arrivals from different connections.

According to the Time-Independence Property we only consider traffic constraints that restrict *relative* arrival times, and not absolute arrival times. For example, a constraint of the form "a message arrives at time-instant $3 + 10 \cdot k$ for all non-negative integers k" does not possess this property, while the constraint "successive arrivals are exactly 10 time units apart" does. The Time-Independence Property is not very restrictive since a priori knowledge of exact arrival times is usually unavailable. Besides, constraints based on absolute time values are necessarily location-specific and have to be different for all nodes traversed by a data stream.

The Connection-Independence Property says that our model *cannot* represent any cross-synchronization between different connections. Hence it is not possible to specify, for example: "messages from connections C_i and C_j do not ever arrive simultaneously."

The notion of well-formed traffic constraints is extremely general and encompasses a wide variety of the kinds of traffic constraints one may encounter in practice. The standard sporadic, Tenet, and leaky-bucket models are all included. So are more sophisticated ones, such as, for example: "for messages from the same connection, successive video-messages arrive at least p_1 time units apart, and each video-message is followed by an audio-message within p_2 time units ($p_2 < p_1$)".

It is important to be able to decide whether a considered connection system satisfies the imposed traffic constraints. Moreover, an algorithm for such a verification should be known. A constraint that conforms to these requirements is referred to as *constructively solvable.* Not every well-formed constraint is constructively solvable. For example, "for messages from the same connection, the inter-arrival time is equal to 420 if the Turing machine program P halts, and 666 otherwise" is not constructively solvable. In this paper we design schedulability tests for connection systems with *well-formed constructively-solvable* traffic constraints.

4 Worst-Case Traffic

For any well-formed traffic constraint Z_i, we can define an *arrival bound function* $\text{arrival}_i(t)$, where $\text{arrival}_i(t)$ equals the maximum amount of data that may be legally delivered from connection C_i during any time interval of length t.

From $\text{arrival}_i(t)$, we can construct an *extreme set* $\mathcal{R}_i$, such that the amount of data delivered by $\mathcal{R}_i$ over the interval $[0, t]$ is equal to $\text{arrival}_i(t)$ for all $t \geq 0$. Observe that $\mathcal{R}_i$ is not necessary a *legal* set of arrivals:

Example 1. Let connection C_1 be characterized by the following traffic constraint Z_1 : "messages of size 4 arrive with period 13; each of them is followed by a message of size 3 in 3 time units and a message of size 5 in 7 time units." Figure 1 contains a legal set R_1 of arrivals, arrival bound function $\text{arrival}_1(t) = 5\left\lceil\frac{t}{13}\right\rceil^+ + 2\left\lceil\frac{t-3}{13}\right\rceil^+ + \left\lceil\frac{t-4}{13}\right\rceil^+ + \left\lceil\frac{t-6}{13}\right\rceil^+ + 3\left\lceil\frac{t-7}{13}\right\rceil^+$, and

5b.4.4

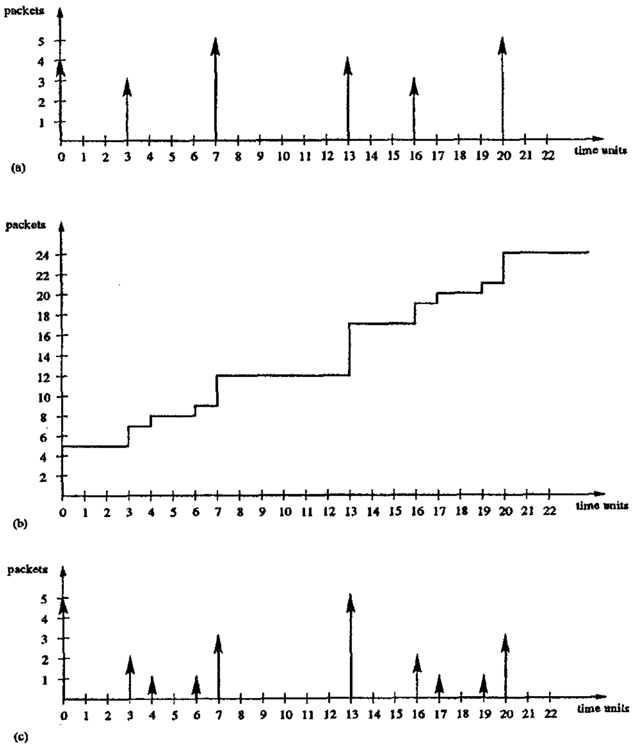

Figure 1. (a) legal set R_1 of arrivals, (b) arrival bound function $\text{arrival}_1(t)$, (c) extreme set $\mathcal{R}_1$ of connection C_1.

extreme set $\mathcal{R}_1$ of connection C_1. Note that $\mathcal{R}_1$ is *not* legal since it does not satisfy constraint Z_1.

(A connection is called *greedy* if beginning from the first message it sends as much data as allowed by the corresponding traffic constraints. Constraints Z_i are referred to as *zero-severe* if the greedy connection C_i generates its extreme set. The class of zero-severe constraints includes, for example, the sporadic, Tenet, and leaky bucket models; however, observe that constraint Z_1 of Example 1 is *not* zero-severe. For constraints that are not zero-severe, the arrival bound function yields an extreme set that is not a legal set of arrivals)

Because of the Time-Independence and Connection-Independence Properties, connections may send their maximum traffic starting at the same time. Thus, the worst-case traffic from the connection system is a union of the connections' extreme sets, called a *system extreme set* $\mathcal{R}$:

$$\mathcal{R} \stackrel{\text{def}}{=} \bigcup_{i=1}^{n} \mathcal{R}_i \tag{1}$$

The latter is represented by a *system arrival bound function:*

$$\text{arrival}(t) = \sum_{i=1}^{n} \text{arrival}_i(t)$$

The definition of the system extreme set $\mathcal{R}$ implies that for any $t_1, t_2 \geq 0$ we have[3]:

$$\text{arrival}(t_1 + t_2 + 1) \leq \text{arrival}(t_1) + \text{arrival}(t_2) \tag{2}$$

5 Scheduling and Schedulability

Result 1 (Dertouzos 74). If set R of arrivals is schedulable, then it can be scheduled without missing any deadlines by the Earliest Deadline First (EDF) scheduling algorithm.

This optimality of EDF implies that EDF is a good scheduling discipline to employ at network nodes, at least from the view of satisfying delay bounds. Furthermore, remarkably efficient implementations of EDF have been devised, in both hardware and software [18]. However in this paper, we do not require that EDF be the scheduling discipline employed at the network nodes — *any* optimal discipline can be used.

If $\lim_{t\to\infty} \frac{\text{arrival}(t)}{t} > 1$, then the amount of arrived traffic is asymptotically larger than a scheduler can serve, and bounded delay guarantees cannot be provided to all the connections. So, schedulability requires satisfaction of the *Utilization Condition:*

$$\lim_{t\to\infty} \frac{\text{arrival}(t)}{t} \leq 1$$

Note that we do not exclude the cases when $\lim_{t\to\infty} \frac{\text{arrival}(t)}{t} = 1$. Even if there are infinite busy periods (time intervals when the scheduler queue is not empty), it may be possible to schedule traffic from all the connections. For example, when the arrival bound function of the only connection of a connection system is $\text{arrival}(t) = 2 + t$, any delay bound larger or equal to 2 can be guaranteed to this connection.

Let us define a *demand function* of an arrival set R as:

$$\text{demand}_{\text{R}}(t) \stackrel{\text{def}}{=} \sum_{i=1}^{n} A_i^{\text{R}}(t - d_i)$$

where t is time passed after the arrival of the first message, d_i is the delay bound of connection C_i, $A_i^{\text{R}}(t-d_i)$ is the amount of data delivered from connection C_i by set R by time $t - d_i$.

Theorem 1. A set R of arrivals is schedulable if and only if $(\forall t \geq 0)[\text{demand}_{\text{R}}(t) \leq t]$.

Proof: Given in [13]. ∎

Let us define a *system demand function* as the demand function of the system extreme set $\mathcal{R}$ (as defined in (1)): $\text{demand}(t) \stackrel{\text{def}}{=} \text{demand}_{\mathcal{R}}(t) = \sum_{i=1}^{n} \text{arrival}_i(t - d_i)$.

[3] For continuous traffic models this condition becomes $\text{arrival}(t_1 + t_2) \leq \text{arrival}(t_1) + \text{arrival}(t_2)$.

Theorem 2. A connection system is schedulable if and only if the system extreme set is schedulable.

Proof: Given in [13]. ∎

Corollary 1 (Schedulability Conditions). A connection system is schedulable iff

$$(\forall t \geq 0)[\mathsf{demand}(t) \leq t] \quad (3)$$

Proof: Follows from Theorems 1 and 2. ∎

Checking condition (3) directly will take infinite time for any schedulable system. Fortunately, when the Utilization Condition is satisfied, it is usually possible to determine a finite *testing bound* t_B and a finite *testing set* S such that $\mathsf{demand}(t) \leq t$ for all $t \in S \subset \{d_1, d_1+1, \ldots, t_B\}$ implies condition (3) (Note that for $0 \leq t < d_1$ we always have $\mathsf{demand}(t) = 0 \leq t$). This fact allows us to derive the more practical schedulability test:

Result 2. A connection system is schedulable iff

$$(\forall t \in S)[\mathsf{demand}(t) \leq t] \text{ and}$$
the Utilization Condition is satisfied

6 Determination of the Testing Set

Even though the smallest testing set S depends upon the specific traffic constraints being considered, there are several general guidelines for detecting S:

(1) The *Too-Late-To-Fail (TLTF) Rule:* if condition (3) is false, then there exists $t_f \geq 0$ such that $\mathsf{demand}(t_f) > t_f$. Sometimes, the latter inequality may be transformed to $t_f < B$ where B does not depend on t_f. In such cases, the failure of $\mathsf{demand}(t) \leq t$ cannot occur for any $t \geq B$, and $t_{TLTF} = \lceil B \rceil - 1$ can be used as a testing bound.

(2) The *Empty-Queue (EQ) Rule:* if for some t_{EQ} we have $\mathsf{arrival}(t_{EQ}) \leq t_{EQ} + 1$ and $\mathsf{demand}(t) \leq t$ for all $d_1 \leq t \leq t_{EQ}$, it means that all messages received during interval $[0, t_{EQ}]$ are served by time $t_{EQ} + 1$ without violation of their deadlines. Then the same is true for the messages arriving after t_{EQ}, since, according to (2), we have $\mathsf{arrival}(t_{EQ} + t + 1) - \mathsf{arrival}(t_{EQ}) \leq \mathsf{arrival}(t)$ for any $t \geq 0$. Thus, t_{EQ} can be used as a testing bound.

(3) The *Worst-Is-Behind (WIB) Rule:* there may exist some t_{WIB} such that $(\forall t > t_{WIB})$ $(\exists t' \leq t_{WIB})[\mathsf{demand}(t) \leq \mathsf{demand}(t') + (t - t')]$. If for all $d_1 \leq t \leq t_{WIB}$ the condition $\mathsf{demand}(t) \leq t$ is satisfied, then so is it for all $t \geq 0$. Thus, t_{WIB} can be employed as a testing bound.

(4) The *Skip Rule:* if there exist some $t_1 \in S$ such that $(\forall t \in \{t_1 + 1, t_1 + 2, \ldots, t_2\})$ $[\mathsf{demand}(t) \leq \mathsf{demand}(t_1) + (t - t_1)]$, then numbers $t_1 + 1, t_1 + 2, \ldots, t_2$ may be excluded from the testing set S.

Example 2. Let us consider connection system $\{C_1, C_2\}$: C_1 is characterized by Z_1 (as introduced in Example 1) and $d_1 = 7$; C_2 conforms to the leaky bucket model with $\sigma = s = 1$, $p = 13$ and has delay bound $d_2 = 7$. The test presented in [16] runs forever and fails to assure schedulability of the system. However, applying the WIB and Skip Rules yields a testing bound $t_B = 19$ and testing set $S = \{7, 10, 14\}$. Therefore, checking condition $\mathsf{demand}(t) \leq t$ only for $t \in \{7, 10, 14\}$ suffices to verify schedulability of system $\{C_1, C_2\}$.

We have used the TLTF, EQ, WIB, and Skip Rules to derive necessary and sufficient schedulability conditions for the traffic models described in Section 3. The results are given in Figure 2. A detailed derivation of the obtained conditions for the leaky bucket model is provided in [13]; other may be derived similarly.

Comparative efficiency of the rules varies for different connection systems. In general, the TLTF Rule outperforms the EQ Rule; however, the latter may provide lower testing bounds when there exists a connection with a very large delay bound. Using the WIB Rule may yield smaller testing sets when the link utilization is high. A quantitative analysis given in [13] demonstrates the usefulness of each rule. [13] also offers recommendations for improving the efficiency of the connection admission in actual networks and provides efficient schedulability tests for the generic traffic model and the leaky bucket model.

7 Conclusions and Future Work

We have investigated the problem of schedulability at nodes of packet-switched virtual-circuit fixed-packet networks. We introduced very general constraints on traffic (*well-formed constraints*). For connection systems that satisfied those constraints, we defined the notion of *extreme sets*, and used this notion to derive necessary and sufficient schedulability conditions. We identified guidelines (*TLTF*, *EQ*, *WIB*, and *Skip Rules*) which could assist in the efficient evaluation of the obtained schedulability conditions.

In the near future, we plan on extending the research described here in several directions:

- We will explore the issue of designing additional rules, similar to the ones in Section 6, that help to improve the efficiency of the admission control.
- We will use the developed methodology for deriving efficient schedulability tests for different network and traffic models. For instance, this work can be extended in a straightforward manner to schedulability testing of systems where messages of the same connection are allowed to have different delay bounds.

Traffic models	Schedulability conditions
The sporadic model	$\sum_{i=1}^{n} \frac{s_i}{p_i} \leq 1$ and $(\forall t \in S)[\mathsf{demand}(t) \leq t]$ where $S = \bigcup_{i=1}^{n} \{t:\ t \leq t_B,\ t = d_i + mp_i,\ m \in \mathrm{N}\}$, $t_B = \min\left\{ \max\left\{ d_n, \left\lceil \frac{\sum_{i=1}^{n} s_i(1-\frac{d_i}{p_i})}{1-\sum_{i=1}^{n} \frac{s_i}{p_i}} \right\rceil -1 \right\},\ \left\lceil \frac{\sum_{i=1}^{n} s_i - 1}{1-\sum_{i=1}^{n} \frac{s_i}{p_i}} \right\rceil -1,\ d_n + \operatorname{lcm}_{i=1,\ldots,n} \{p_i\} - 1 \right\}$, and $\mathsf{demand}(t) = \sum_{i=1}^{n} s_i \left\lceil \frac{t-d_i}{p_i} \right\rceil^{+}$
The continuous version of the sporadic model	$\sum_{i=1}^{n} \rho_i \leq 1$
The Tenet model	$\sum_{i=1}^{n} \frac{s_i}{x_{ave,i}} \leq 1$ and $(\forall t \in S)[\mathsf{demand}(t) \leq t]$ where $S = \bigcup_{i=1}^{n} \{t:\ t \leq t_B,\ t = d_i + mI_i + kx_{min,i},\ m \in \mathrm{N},\ k = 0, 1, \ldots, \frac{I_i}{x_{ave,i}} - 1\}$, $t_B = \min\left\{ \max\left\{ d_n, \left\lceil \frac{\sum_{i=1}^{n} \frac{s_i}{x_{ave,i}}(I_i-d_i)}{1-\sum_{i=1}^{n} \frac{s_i}{x_{ave,i}}} \right\rceil -1 \right\},\ \left\lceil \frac{\sum_{i=1}^{n} \frac{s_i}{x_{ave,i}} I_i - 1}{1-\sum_{i=1}^{n} \frac{s_i}{x_{ave,i}}} \right\rceil -1,\ d_n + \operatorname{lcm}_{i=1,\ldots,n} \{I_i\} - 1 \right\}$, and $\mathsf{demand}(t) = \sum_{i=1}^{n} \mathsf{threshold}(s_i \left(\left\lfloor \frac{t-d_i}{I_i} \right\rfloor \frac{I_i}{x_{ave,i}} + \min\left\{ \left\lceil (\frac{t-d_i}{I_i} - \left\lfloor \frac{t-d_i}{I_i} \right\rfloor) \frac{I_i}{x_{min,i}} \right\rceil^{+}, \frac{I_i}{x_{ave,i}} \right\} \right), t - d_i)$
The continuous version of the Tenet model	$\sum_{i=1}^{n} \rho_{ave,i} \leq 1$ and $(\forall t \in S)[\mathsf{demand}(t) \leq t]$ where $S = \bigcup_{i=1}^{n} \{t:\ t \leq t_B,\ t = d_i + (m + \frac{\rho_{ave,i}}{\rho_{max,i}})I_i,\ m \in \mathrm{N}\}$, $t_B = \min\left\{ \max\left\{ d_n, \frac{\sum_{i=1}^{n} \rho_{ave,i}(I_i-d_i)}{1-\sum_{i=1}^{n} \rho_{ave,i}} \right\},\ \frac{\sum_{i=1}^{n} \rho_{ave,i} I_i}{1-\sum_{i=1}^{n} \rho_{ave,i}},\ d_n + \operatorname{lcm}_{i=1,\ldots,n} \{I_i\} - 1 \right\}$, and $\mathsf{demand}(t) = \sum_{i=1}^{n} \mathsf{threshold}(\rho_{ave,i} \left\lfloor \frac{t-d_i}{I_i} \right\rfloor I_i + \min\left\{ \rho_{max,i}(\frac{t-d_i}{I_i} - \left\lfloor \frac{t-d_i}{I_i} \right\rfloor) I_i, \rho_{ave,i} I_i \right\}, t - d_i)$
The leaky bucket model	$\sum_{i=1}^{n} \rho_i \leq 1$ and $(\forall j, 1 \leq j \leq n)[\sum_{d_i \leq d_j} (\sigma_i + \rho_i(d_j - d_i)) \leq d_j]$
The discrete version of the leaky bucket model	$\sum_{i=1}^{n} \frac{s_i}{p_i} \leq 1$ and $(\forall t \in S)[\mathsf{demand}(t) \leq t]$ where $S = \bigcup_{i=1}^{n} \{t:\ t \leq t_B,\ t = d_i + mp_i,\ m \in \mathrm{N}\}$, $t_B = \min\left\{ \max\left\{ d_n, \left\lceil \frac{\sum_{i=1}^{n} (\sigma_i + s_i(1-\frac{d_i}{p_i}))}{1-\sum_{i=1}^{n} \frac{s_i}{p_i}} \right\rceil -1 \right\},\ \left\lceil \frac{\sum_{i=1}^{n} (\sigma_i + s_i) - 1}{1-\sum_{i=1}^{n} \frac{s_i}{p_i}} \right\rceil -1,\ d_n + \operatorname{lcm}_{i=1,\ldots,n} \{p_i\} - 1 \right\}$, and $\mathsf{demand}(t) = \sum_{i=1}^{n} \mathsf{threshold}(\sigma_i + s_i \left\lceil \frac{t-d_i}{p_i} \right\rceil^{+}, t - d_i)$

Notations:
(1) threshold function: $\mathsf{threshold}(x, y) = x$ for $y \geq 0$, and $\mathsf{threshold}(x, y) = 0$ if $y < 0$;
(2) N is the set of natural numbers;
(3) $\operatorname{lcm}_{i=1,\ldots,n} \{x_i\}$ is the least common multiple of numbers x_i where $i = 1, \ldots, n$.

Figure 2. Necessary and sufficient schedulability conditions.

- We plan to investigate providing performance guarantees to heterogeneous connection systems where different connections are described by different traffic models.
- The problem of verifying whether adding a new connection to a schedulable connection system is safe for system schedulability is important from a practical perspective. Often this problem can be solved more efficiently than by *re*testing the entire new system for schedulability. We plan to apply the techniques introduced in this paper to find exact and efficient solutions for this problem as well as for the problem of determination of the minimum delay bound that may be associated with the new connection.

References

[1] S. Baruah, A. Mok, and L. Rosier. "The Preemptive Scheduling of Sporadic, Real-Time Tasks on One Processor," Proceedings of the 11th Real-Time Systems Symposium, pp. 182-190, 1990.

[2] S. Baruah. "The Uniprocessor Scheduling of Sporadic Real-Time Tasks," Ph.D. Thesis, The University of Texas at Austin, 1993.

[3] I. Cidon, I. Gopal, G. Grover, and M. Sidi, "Real-Time Packet Switching: A Performance Analysis," IEEE Journal on Selected Areas in Communications, 6(9):1576-1586, 1988.

[4] R.L. Cruz. "A Calculus for Network Delay, Part I: Network Elements in Isolation," IEEE Transactions on Information Theory, 37(1):114-131, January 1991.

[5] R.L. Cruz. "A Calculus for Network Delay, Part II: Network Analysis," IEEE Transactions on Information Theory, 37(1):132-141, January 1991.

[6] R.L. Cruz. "Quality of Service Guarantees in Virtual Circuit Switched Networks," IEEE Journal on Selected Areas in Communications, 13(6):1048 - 1056, August 1995.

[7] A. Demers, S. Keshav, and S. Shenker. "Analysis and Simulation of a Fair Queueing Algorithm," Proceedings of SIGCOMM '89, pp. 1-12, 1989.

[8] M. Dertouzos. "Control Robotics: the Procedural Control of Physical Processors," Proceedings of the IFIP Congress, pp. 807-813, 1974.

[9] D. Ferrari and D. C. 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.

[10] L. Georgiadis, R. Guerin, and A. Parekh. "Optimal multiplexing on a single link: delay and buffer requirements," Proceedings of INFOCOM '94, pp. 524-532, 1994.

[11] L. Georgiadis, R. Guerin, V. Peris, and K. N. Sivarajan. "Efficient Network QoS Provisioning Based on Per Node Traffic Shaping," IEEE/ACM Transactions on Networking, 4(4):482-501, August 1996.

[12] S. J. Golestani. "A Stop-and-Go Queueing Framework for Congestion Management," Proceedings of SIGCOMM '90, pp. 8-18, September 1990.

[13] S. Gorinsky, S. Baruah, T. J. Marlowe, and A. D. Stoyenko. "Exact and Efficient Analysis of Schedulability in Fixed-Packet Networks: A Generic Approach," Technical Report CIS-96-26, Department of Computer and Information Science, NJIT, 1996.

[14] P. Goyal, S. S. Lam, and H. M. Vin. "Determining End-to-End Delay Bounds In Heterogeneous Networks," Proceedings of NOSSDAV '95, pp. 287-298, April 1995.

[15] C.R. Kalmanek, H. Kanakia, and S. Keshav. "Rate-Controlled Servers for Very High-Speed Networks," Proceedings of GLOBECOM '90, pp. 300.3.1 - 300.3.9, December 1990.

[16] J. Liebeherr, D. E. Wrege, and D. Ferrari. "Exact Admission Control in Networks with Bounded Delay Sevices," to appear: IEEE/ACM Transactions on Networking, December 1996.

[17] A. Mok. "Fundamental Design Problems of Distributed Systems for the Hard Real-Time Environment," Ph.D. Thesis, MIT Laboratory for Computer Science, Technical Report MIT/LCS/TR-297, May 1983.

[18] A. Mok. "Task Management Techniques for Enforcing ED Scheduling on a Periodic Task Set," Proceedings of 5-th IEEE Workshop on Real-Time Software and Operating Systems, May 1988.

[19] A. K. Parekh and R. G. Gallager. "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," IEEE/ACM Transactions on Networking, 1(3):344-357, June 1993.

[20] A. K. Parekh and R. G. Gallager. "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case," IEEE/ACM Transactions on Networking, 2(2):137-150, April 1994.

[21] D. E. Wrege, E. W. Knightly, H. Zhang, and J. Liebeherr. "Deterministic Delay Bounds for VBR Video in Packet-Switching Networks: Fundamental Limits and Practical Tradeoffs," IEEE/ACM Transactions on Networking, 4(3):352-362, June 1996.

[22] Lixia Zhang. "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks," ACM Transactions on Computer Systems, 9(2):101-124, May 1991.

[23] Q. Zheng and K. G. Shin. "On the Ability of Establishing Real-Time Channels in Point-to-Point Packet-Switched Networks," IEEE Transactions on Communications, 42(2/3/4):1096-1105, February/March/April 1994.