SCHEDULING MULTIRATE PERIODIC TRAFFIC IN A PACKET SWITCH

James Giles and Bruce Hajek

Abstract

This paper explores scheduling multiclass periodic traffic with deadlines in a time slotted system through an $N \times N$ switch. Different traffic streams may have different periods and different rates. Each packet must depart before the next packet in its stream arrives.

An algorithm is given to schedule all traffic if each period evenly divides all longer periods, and if the utilization at each link is no larger than one. Furthermore, this algorithm can schedule all traffic with no restrictions on the periods if the link utilization does not exceed 1/4. A simple greedy algorithm is presented and is shown to schedule all traffic with no restrictions on the periods if the link utilization does not exceed 1/14. These two algorithms can also be applied to more general deterministically constrained traffic when the link utilization is at most 1/4 and 1/14, respectively.

James Giles and Bruce Hajek are with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign.

Address of Corresponding Author:

Prof. Bruce Hajek
Coordinated Science Laboratory
1308 West Main Street
Urbana, IL 61801, USA
Tel: (217) 333 3605
e-mail: b-hajek@uiuc.edu

1 INTRODUCTION

The problem of scheduling multirate traffic with guaranteed quality of service through a non-blocking $N \times N$ crossbar switch has become important for systems such as Asynchronous Transfer Mode networks and satellites where voice, data, and multimedia traffic must be handled. One way to make quality of service guarantees is to require that traffic be deterministically constrained. A message source declares the constraint with which it will comply. In turn, the switch can use the constraint information to determine whether or not the message source should be admitted. Since the message source is deterministically constrained, there is no danger of having more packets to schedule than the switch can handle. Thus algorithms and simple admission control tests for scheduling deterministically constrained traffic could be useful for high speed switching systems.

This paper considers scheduling multirate traffic with deadlines through a non-blocking crossbar switch. In particular, it is shown that there is an algorithm to schedule certain classes of periodic traffic when the maximum link utilization is no larger than one. Furthermore, periodic traffic with any periods and certain more general deterministically constrained traffic can be scheduled when the maximum link utilization is no larger than 1/4. A *simple* algorithm for scheduling such traffic with maximum link utilization no larger than 1/14 is also presented.

The Multiple-Period Time Slot Assignment (MP-TSA) problem is that of scheduling periodic message streams with different periods and with deadlines through a crossbar switch such that no packet misses a deadline. A periodic message stream, M, with period p is an infinite sequence of packets with packet n arriving in time slot $j + np$. Here j is the phase of the message stream. Each packet in such a stream must be scheduled to depart in one of the p consecutive slots beginning with the slot in which the packet arrived. (A more general definition of a period p message stream would allow more than one packet arrival per period, but such a stream can be decomposed into multiple streams with one packet arrival per period, so we do not lose any generality.) We consider an $N \times N$ switch capable of transmitting one packet from each input link and receiving one packet at each output link in a given time slot. The switch is assumed to be rearrangeably non-blocking so that any set of packets at distinct inputs destined for distinct outputs can be transmitted in a single slot. The MP-TSA problem was considered by Philp and Liu [1], who conjectured that it is *always* possible to find a schedule in an $N \times N$ switch when the maximum link utilization is no more than one. While we have not resolved this conjecture, our results are consistent with it.

For the MP-TSA problem with n message streams, the lth message stream, M_l, has an associated three tuple (r_l, c_l, p_l) where $r_l \in \{1, \ldots, N\}$ specifies the input link, $c_l \in \{1, \ldots, N\}$ specifies the output link, and $p_l \in Z^+$ specifies the period length for the message stream. It is assumed that the message streams are synchronized so that the phase of each message stream is zero. Given this synchronization, a schedule need only be specified for $S = \text{lcm}(p_1, \ldots, p_n)$ time slots since all traffic in the system up through time slot $S-1$ must leave by the end of slot $S-1$, and the traffic repeats, starting at slot S. For periodic traffic, $S = \text{lcm}(p_1, \ldots, p_n)$ is called the schedule length. As an example, traffic with periods $2, 3$, and 4 requires a schedule for the first $S = \text{lcm}(2, 3, 4) = 12$ slots since the traffic will repeat after twelve slots.

For clarity, an example MP-TSA problem is provided. Consider the seven message streams for a 2×2 switch indicated in Figure 1. As seen in Figure 1, message stream M_6 has a packet arrival every eight time slots at input 1 to be transmitted to output 2. The utilization of input link 1 is 7/8, since streams M_1, M_2, and M_6 require utilizations 1/2, 1/4, and 1/8, respectively. Similarly, the utilizations of input link 2 and output links 1 and 2 are 1, 1, and 7/8, respectively. Therefore, we expect a length 8 schedule to exist if the Philp and Liu conjecture is true. It is easy to find such a schedule for this example, and one is shown in Figure 2. Each entry under a slot number indicates the output link that is required for the corresponding message stream. For example, a packet from stream M_6 is transmitted in slot 4 to output 2. In any slot column, there is at most one entry for output 1 and one entry for output 2. Looking only at the input 1 rows or the input 2 rows, there is at most one entry in any column. Therefore, at most one packet is transmitted from any given input or received by any given output in each time slot and all deadlines are met.

Several special cases of the MP-TSA problem have been studied. In particular, it has been shown by Liu and Wayland [2] that the well-known Earliest-Deadline-First algorithm applied to a 1×1 switch with multiple-period traffic provides a schedule if the total utilization for the input (and hence the output) is no larger than one. For an $N \times N$ switch and synchronized periodic traffic with one *common* period length, a classic Time-Slot Assignment (TSA) algorithm will find a schedule. In the TSA problem, the goal is to deliver all packets in a minimum number of slots. Algorithms presented by Inukai [3] and Ganz and Gao [4] are examples of classic TSA scheduling algorithms. However, these classic TSA algorithms cannot find a feasible schedule for periodic traffic with more than one period because the algorithms only guarantee that all traffic will be scheduled within a certain schedule length which is the same for all traffic. There is no guarantee with classic TSA that a particular periodic message with period shorter than the schedule length will meet its deadline.

It should be noted that the MP-TSA problem would be trivial if packets could be subdivided. In the MP-TSA problem, the transmission of a packet requires an entire time slot. If fractions of packets could be scheduled in a time slot, however, each packet could be transmitted at a constant rate equal to one over its period with all packets meeting their deadlines and with the utilization at each link no larger than one at all times.

This paper explores the MP-TSA problem for an $N \times N$ crossbar switch. In Section 2, it is shown that if each period evenly divides all longer periods and if the maximum link utilization is no larger than one, a schedule can be found by an algorithm we call the Nested Period Scheduling (NPS) algorithm. Using the NPS algorithm, it is also possible to schedule traffic with *arbitrary* periods and with no synchronization when link utilization is no larger than 1/4. A *simple* heuristic algorithm called the Slot-by-Slot, Earliest-Deadline-First, Earliest-Arrival-First (SS-EDF-EAF) algorithm, is presented in Section 3 for scheduling periodic traffic with any set of periods on an $N \times N$ switch. It is shown that for utilization no larger than 1/14 on any link, SS-EDF-EAF produces a feasible schedule. Scheduling traffic with more general deterministic constraints, rather than just periodic streams, is considered in Section 4. It is shown that the NPS algorithm can schedule deterministically constrained traffic when the link utilization is no more than 1/4 on any link and the SS-EDF-EAF algorithm can be used to find a schedule for deterministically constrained traffic when the link utilization is no more than 1/14.

2 NESTED PERIOD SCHEDULING ALGORITHM

In this section, the Nested Period Scheduling (NPS) algorithm and a special case of the Multiple Period Time Slot Assignment problem for which the NPS algorithm can find a schedule are presented. In particular, it is shown that synchronized periodic traffic with nested periods and with input and output utilizations no larger than one can be scheduled through an $N \times N$ switch by the NPS algorithm such that no packets miss a deadline. In addition, it is shown that the NPS algorithm can be applied to non-synchronized periodic traffic having *any* period lengths when the maximum link utilization does not exceed 1/4. Finally, the complexity of the NPS algorithm is explored.

A. Terminology

The following terminology will be used throughout this section. Let a set of synchronized periodic message streams be given with stream M_l having parameters (r_l, c_l, p_l). Let $P_1, \ldots, P_m$ denote the set of distinct period lengths, ordered so that $P_1 > P_2 > \ldots > P_m$. This set of m distinct periods is said to be *nested* if for some $\{l_1, l_2, \ldots, l_{m-1}\} \in Z^+$, the period lengths satisfy $P_i = l_i P_{i+1}$ for $1 \leq i < m$.

Let v_{ijk} be the set of streams from input i to output j having period P_k and let $x_{ijk} = |v_{ijk}|$ be the number of such streams. A *traffic matrix* is an $N \times N$ matrix where entry i, j is the set of message streams from input i to output j. The traffic demand for period P_k traffic can be represented by the traffic matrix $D_k = (v_{ijk} : 1 \leq i, j \leq N)$. A *line* of a traffic matrix refers to a row or column of the traffic matrix while a *line sum* refers to the number of message streams in a particular line. The *schedule length*, S, for synchronized traffic having m distinct periods is given by $S = \text{lcm}\{P_1, P_2, \ldots, P_m\}$.

B. Nested Period Scheduling Algorithm Description

We first provide an informal description of the Nested Period Scheduling algorithm. Next, it will be shown that the NPS algorithm can always find a schedule for periodic traffic with nested, synchronized periods when the maximum link utilization does not exceed one. Finally, an example will be worked using the NPS algorithm for the purpose of illustration.

The Nested Period Scheduling algorithm, explicitly described in Figure 3, works by iteratively decomposing traffic matrices with longer periods into sums of traffic matrices with shorter periods. Figure 4 illustrates this for a set of message streams with periods 2, 4, and 8.

Given traffic with distinct periods $P_1 > \ldots > P_m$, the NPS algorithm starts by calculating the aggregate link utilizations of the traffic with periods $P_2, \ldots, P_m$. Next, the number of slots, f_l, in each P_2 period that will *not* be used by traffic with periods $P_2, \ldots, P_m$ is found for each link, l. It satisfies $f_l = P_2(1 - u_l)$ where u_l is the aggregate utilization of link l for traffic with periods $P_2, \ldots, P_m$. The period P_1 traffic matrix is then decomposed into $l_1 = P_1/P_2$ generalized switching matrices such that for each line l, the lth line sum of each switching matrix is at most f_l. Each of the l_1 switching matrices corresponds to the period P_1 traffic that will be scheduled in one of the P_2 periods. Each of these switching matrices is then combined with a copy of the period P_2 traffic matrix, since one packet from each period P_2 stream is also to be scheduled in each P_2 period. The resulting composite matrices then represent period P_1 and period P_2 traffic. These composite matrices are traffic matrices to be scheduled in P_2 periods where the kth composite traffic matrix, $k \in \{1, \ldots, l_1\}$, corresponds to the traffic that will be scheduled in the kth P_2 period.

To summarize the steps so far, the traffic matrix for period P_1 traffic is decomposed into $l_1 = P_1/P_2$ traffic matrices, and then period P_2 traffic is added to each one. This produces $l_1 = P_1/P_2$ composite switching matrices, each representing period P_1 and P_2 traffic to be scheduled in a P_2 period. Next, the process is repeated. Each of the l_1 composite matrices is decomposed into $l_2 = P_2/P_3$ matrices, and then period P_3 traffic is added to each one. This produces a total of $l_1 l_2 = (P_1/P_2)(P_2/P_3)$ composite matrices, each representing period P_1, P_2, and P_3 traffic to be scheduled in a P_3 period.

The cycle continues until finally there are $l_1 l_2 \cdots l_{m-1} = P_1/P_m$ composite period P_m traffic matrices, each having link utilization at most one. These traffic matrices can each be scheduled in P_m or fewer slots using a classic TSA algorithm as described in Inukai [3] or Ganz and Gao [4]. Because all traffic for a given period is decomposed into matrices that are scheduled within the period length, all traffic will meet its deadline.

In this paragraph, we comment on the generalized Time Slot Assignment problem used in the decomposition step of the Nested Period Scheduling algorithm. In the basic Time Slot Assignment problem, a traffic matrix is decomposed into switching matrices having at most *one* entry in each line where a line is a row or a column. In contrast, the generalized Time Slot Assignment problem is to decompose a traffic matrix into switching matrices with at most $N_l \geq 0$ entries in each line l. It is shown that an $N \times N$ traffic matrix, D, can be decomposed into p generalized matrices under certain conditions on the row and column sums of D and the link capacities.

Lemma 2.1 *Given input link capacities $r_i \geq 0$, for $i \in \{1, \ldots, N\}$ and output link capacities $c_j \geq 0$, for $j \in \{1, \ldots, N\}$, it is possible to decompose an $N \times N$ traffic matrix, D,* *into p matrices with at most $r_i \geq 0$ entries in row i and at most $c_j \geq 0$ entries in row j if $\sum_{j=1}^{N} x_{ij} \leq r_i p$ for each i and $\sum_{i=1}^{N} x_{ij} \leq c_j p$ for each j where x_{ij} is the number of message streams going from input i to output j.*

A proof of Lemma 2.1 and corresponding algorithms are given by Ganz and Gao [5], Sasaki and Jain [6], and Gopal and Bonuccelli [7].

Lemma 2.2 *Given an $N \times N$ switch with synchronized, nested periodic traffic with maximum link utilization no greater than one, the NPS algorithm will always decompose a composite traffic matrix, C_k, representing periodic traffic with periods $P_1, \ldots, P_k$, into at most $l_k = P_k / P_{k+1}$ generalized switching matrices.*

A proof by induction will be given. Suppose that the periodic traffic has m distinct periods. Let C_1 be equal to the traffic matrix, D_1, representing period P_1 traffic. Let y_{lk} denote the total number of message streams with traffic for link l and period P_k where $l \in \{1, \ldots, 2N\}$ indexes all possible input-output pairs. Consider the following statement

> $S(r)$: Any composite traffic matrix, C_r, representing periodic traffic with periods $P_1, \ldots, P_r$ can be decomposed into at most $l_r = P_r / P_{r+1}$ switching matrices such that the maximum link utilization for any P_{r+1} period is at most one.

First, $S(1)$ is proved. The residual line capacity per length P_2 period is given by $f_l = P_2(1 - \sum_{k=2}^{m} y_{lk}/P_k)$ for each l. Because the maximum link utilization does not exceed 1, $\sum_{k=1}^{m} y_{lk}/P_k \leq 1$ for each l. This expression can be rewritten for each l as $y_{l1} \leq P_1(1 - \sum_{k=2}^{m} y_{lk}/P_k) = l_1 f_l$. However, y_{l1} is merely the total number of message streams at each link l with period P_1 denoted by the traffic matrix C_1 and f_l is the residual line capacity per length P_2 period. Thus by Lemma 2.1, the traffic matrix, C_1, can be decomposed into at most l_1 generalized switching matrices such that the link utilization does not exceed f_l. Clearly the maximum link utilization during any P_2 period does not exceed one. Therefore $S(1)$ is true.

Assume that $S(p)$ is true. Then a composite matrix C_p representing traffic with periods $P_1, \ldots, P_p$ can be decomposed into at most $l_p = P_p/P_{p+1}$ switching matrices. Further, the utilization of link l in each P_{p+1} period satisfies $\sum_{k=1}^{m} y_{lk}/P_k \leq 1$. Thus the maximum number of streams represented by a composite matrix C_{p+1} to be scheduled in a P_{p+1} period for link l is given by $y_{l(p+1)} + \sum_{k=1}^{p} y_{lp}/P_k$. The residual line capacity per length P_{p+1} period is given by $f_l = P_{p+1}(1 - \sum_{k=p+1}^{m} y_{lk}/P_k)$ for each l. Since the link utilization for link l satisfies $\sum_{k=1}^{m} y_{lk}/P_k \leq 1$ in each P_{p+1} period, we see that $y_{l(p+1)} + \sum_{k=1}^{p} y_{lp}/P_k \leq P_{p+1}(1 - \sum_{k=p+2}^{m} y_{lk}/P_k) = l_{p+1}f_l$. Thus by Lemma 2.1, the traffic matrix, C_{p+1}, can be decomposed into at most l_{p+1} generalized switching matrices such that the link utilization does not exceed f_l. The maximum link utilization during any P_{p+1} period does not exceed one. Therefore $S(p+1)$ is true. Since $S(p)$ implies $S(p+1)$, and $S(1)$ is true, $S(p)$ is true for all $p \geq 1$ by mathematical induction. The proof is complete.

Theorem 2.1 *For an $N \times N$ switch, traffic having $m \geq 1$ synchronized, nested periods with link utilization no greater than one can be scheduled with the Nested Period Scheduling algorithm.*

By Lemmas 2.1 and 2.2, the algorithm can be executed as described. In addition, all deadlines are met because each stream of any given period P_k is guaranteed to receive a slot within each successive interval of length P_k when the period P_k traffic is put into a composite matrix. In subsequent iterations, the exact locations of the slots within the length P_k intervals are successively restricted, always remaining in the required length P_k intervals. The proof is complete.

An example of the Nested Period Scheduling algorithm is now presented for the message set in Figure 1. The following traffic matrices can be written for the message set in Figure 1 with $P_1 = 8$, $P_2 = 4$, and $P_3 = 2$

$$D_1 = \begin{pmatrix} \{\} & \{6\} \\ \{\} & \{4,5\} \end{pmatrix}$$

$$D_2 = \begin{pmatrix} \{\} & \{2\} \\ \{\} & \{7\} \end{pmatrix}$$

$$D_3 = \begin{pmatrix} \{1\} & \{\} \\ \{3\} & \{\} \end{pmatrix}$$

Clearly, the periods are nested since $8 = 2 \times 4$ and $4 = 2 \times 2$. Calculating the aggregate utilizations for period 4 and period 2 traffic, we obtain $u_1 = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$, $u_2 = \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$, $u_3 = 0 + 1 = 1$, and $u_4 = \frac{1}{2} + 0 = \frac{1}{2}$. Then the amount of residual line capacity per length P_2 period if all traffic with periods P_2 and P_3 is scheduled in a period of length P_2 is given by $f_1 = P_2(1 - u_1) = 1$, $f_2 = P_2(1 - u_2) = 1$, $f_3 = P_2(1 - u_3) = 0$, and $f_4 = P_2(1 - u_4) = 2$. Using the Ganz and Gao [5] algorithm we decompose D_1

$$D_1 = \begin{pmatrix} \{\} & \{6\} \\ \{\} & \{4\} \end{pmatrix} + \begin{pmatrix} \{\} & \{\} \\ \{\} & \{5\} \end{pmatrix}$$

Combining each of the decomposed matrices with a D_2 traffic matrix, two matrices, D_2^1 and D_2^2, corresponding to the first four slots and the last four slots of the eight slot schedule, are obtained

$$D_2^1 = \begin{pmatrix} \{\} & \{2,6\} \\ \{\} & \{4,7\} \end{pmatrix}$$

and

$$D_2^2 = \begin{pmatrix} \{\} & \{2\} \\ \{\} & \{5,7\} \end{pmatrix}$$

The residual line capacity left, if period 2 traffic is scheduled in a period of length 2, is given by $f_1 = 1$, $f_2 = 1$, $f_3 = 0$, and $f_4 = 2$. We must therefore decompose the traffic matrices D_2^1 and D_2^2 into matrices with line sums no larger than $f_1 = 1$, $f_2 = 1$, $f_3 = 0$ and $f_4 = 2$. Again using the Ganz and Gao algorithm, the following decompositions are obtained for each of the two traffic matrices

$$D_2^1 = \begin{pmatrix} \{\} & \{2\} \\ \{\} & \{4\} \end{pmatrix} + \begin{pmatrix} \{\} & \{6\} \\ \{\} & \{7\} \end{pmatrix}$$

$$D_2^2 = \begin{pmatrix} \{\} & \{2\} \\ \{\} & \{7\} \end{pmatrix} + \begin{pmatrix} \{\} & \{\} \\ \{\} & \{5\} \end{pmatrix}$$

Combining each decomposed matrix with a D_3 matrix, the following four traffic matrices, D_1^1, D_1^2, D_1^3, and D_1^4, representing the traffic for two slots each of the length eight schedule are obtained

$$D_1^1 = \begin{pmatrix} \{1\} & \{2\} \\ \{3\} & \{4\} \end{pmatrix}$$

$$D_1^2 = \begin{pmatrix} \{1\} & \{6\} \\ \{3\} & \{7\} \end{pmatrix}$$

$$D_1^3 = \begin{pmatrix} \{1\} & \{2\} \\ \{3\} & \{7\} \end{pmatrix}$$

$$D_1^4 = \begin{pmatrix} \{1\} & \{\} \\ \{3\} & \{5\} \end{pmatrix}$$

These four matrices can be scheduled in two slots each by scheduling streams 1 and 3 in the first slot, 2 and 4 in the second slot, 1 and 3 in the third slot, 6 and 7 in the fourth slot, 1 and 3 in the fifth slot, 2 and 7 in the sixth slot, 1 and 3 in the seventh slot, and stream 5 in the eighth slot. In general, a classic TSA algorithm can be used to schedule the remaining traffic matrices in this final step. The schedule found using the NPS algorithm is the same as the schedule presented in Figure 2. Because of the way traffic with longer periods is partitioned into shorter periods, all deadlines are met. We now show that it is always possible to find a feasible schedule for traffic with synchronized, nested periods and utilization no larger than one on an $N \times N$ switch.

C. A Service-Oriented View of the NPS Algorithm

The Nested Period Scheduling Algorithm can be thought of as providing a certain guaranteed service. Explicitly, if a message stream is declared to be a periodic stream with period p and initial packet arrival time slot 0, then the algorithm assigns a departure slot to the stream in each of the *type p service intervals*, which are the intervals

$$\{0, \ldots, p-1\}, \{p, \ldots, 2p-1\}, \ldots \tag{2.1}$$

One can envision *any* type of message stream declaring that it is a periodic stream with period p, and in return receiving a *potential* departure within each of the type p service intervals. If the rate of the stream is less that $1/p$, then some of the potential departure slots will not be needed. This service-oriented view is useful for applying the Nested Period Scheduling algorithm to different classes of traffic.

D. NPS Algorithm Applied to Traffic with Any Periods

It has been shown that the Nested Period Scheduling algorithm will always find a schedule for synchronized periodic traffic when the periods are nested and the link utilization does not exceed one. Here it is shown that the NPS algorithm can find a schedule for *non-synchronized* periodic traffic with *any* period lengths for link utilization up to $1/4$. The service-oriented view of the NPS algorithm discussed above is employed. The approach is for each periodic message stream to report that it actually has period 2^r with r small enough so that a type 2^r service interval is always completely contained in any interval of the actual period length. Since the reported periods must all be powers of two, they are nested, and thus the traffic can be scheduled by the NPS algorithm if for each link the sum of the reciprocals of the reported period lengths does not exceed one.

Corollary 2.1 *Given an $N \times N$ switch, traffic having $m \geq 1$ possibly asynchronous periods of any length with link utilization no greater than $1/4$ can be scheduled.*

The idea of the proof of Corollary 2.1 is that a message stream with period length P_k and arbitrary phase declares to the NPS algorithm that it is a periodic source of period $\tilde{P}_k = \lfloor (P_k+1)/2 \rfloor_2$, where $\lfloor x \rfloor_2$ is the greatest power of two less than or equal to x. It is easy to see that this choice of $\tilde{P}_k$ insures that any P_k consecutive slots contain a complete type $\tilde{P}_k$ service interval. Hence, the message stream will be assigned a potential departure by the NPS algorithm at least once in every P_k slots as required.

Assume that there are m distinct periods and that the total link utilization does not exceed $1/4$. Then $\sum_{j=1}^{N} \sum_{k=1}^{m} \frac{x_{i'jk}}{P_k} \leq 1/4$ for each i' and $\sum_{i=1}^{N} \sum_{k=1}^{m} \frac{x_{ij'k}}{P_k} \leq 1/4$ for each j'. If instead we report periods of length $\lfloor (P_k+1)/2 \rfloor_2$ for P_k, then for each i' and j' we have (using the fact that $\tilde{P}_k \geq P_k/4$)

$$\sum_{j=1}^{N}\sum_{k=1}^{m}\frac{x_{i'jk}}{\tilde{P}_k} \leq 4\sum_{j=1}^{N}\sum_{k=1}^{m}\frac{x_{i'jk}}{P_k} \leq 1 \tag{2.2}$$

and

$$\sum_{i=1}^{N}\sum_{k=1}^{m}\frac{x_{ij'k}}{\tilde{P}_k} \leq 4\sum_{i=1}^{N}\sum_{k=1}^{m}\frac{x_{ij'k}}{P_k} \leq 1 \tag{2.3}$$

Thus by reporting periods with lengths that are powers of two, the utilization for all links using the reported periods is no larger than one if the actual link utilization is no larger than 1/4. By Theorem 2.1, this new traffic can be scheduled by the NPS algorithm using the service-oriented view since the new periods are nested and the link utilization with the reported periods is no larger than one. It should be emphasized that the original periodic streams do not have to be synchronized.

From the development above, it is clear that if deadlines are slightly relaxed, higher utilization can be achieved. In particular, if each link utilization does not exceed 1/2 and a message stream with period length P_k reports a period length $\tilde{P}_k = \lfloor (P_k + 1) \rfloor_2$, each packet in the stream will depart no more than $\tilde{P}_k - 1$ slots after its deadline.

E. Complexity of the Nested Period Scheduling Algorithm

There are several basic TSA algorithms that can be used to schedule the submatrices in the final step of the Nested Period Scheduling algorithm. Additionally, there are several algorithms that can be used to decompose nested period traffic matrices as no

ted in Lemma 2.1. The complexity of the NPS algorithm using the algorithms of Sasaki and Jain [6] and Ganz and Gao [5] for decomposing traffic matrices and using the algorithms of Inukai [3] and Ganz and Gao [4] for scheduling the resulting submatrices will be discussed below.

We consider two basic TSA algorithms for scheduling the traffic in the submatrices obtained after all periodic traffic is decomposed into the smallest possible period of length P_m. The algorithm of Inukai [3] finds a matching at each slot. Each matching requires $O(N^3)$ time and there are P_m such matchings required for one submatrix yielding a complexity of $O(P_m N^4)$ for each submatrix. The total number of submatrices is $l_1 \cdots l_{m-1}$ so that the total complexity for using Inukai's algorithm to schedule all the submatrices is $O(SN^3)$ where S is the schedule length, $S = \text{lcm}\{P_1, P_2, \ldots, P_m\}$.

For a given submatrix, the algorithm of Ganz and Gao[4] makes one matching that takes $O(N^3)$ time. Then, the Ganz and Gao algorithm augments the matrix up to one time for each remaining entry in the matrix for a total of $N^2 - 1$ augmentations. The augmentations take $O(N^2)$ time each. Thus the total complexity of scheduling all submatrices with the Ganz and Gao algorithm is $O(\frac{S}{P_m}N^4)$.

The decomposition steps of the NPS algorithm can be implemented using either an algorithm of Sasaki and Jain [6] or of Ganz and Gao [5]. Sasaki and Jain [6] provide an algorithm that can decompose a traffic matrix D into submatrices in $O(\bar{w}N^4)$ time where $\bar{w}$ is the average capacity of the submatrices and N is the number of nodes. When decomposing a matrix having period P_k, the average capacity of the submatrix satisfies $\bar{w} \leq l_{k+1}l_{k+2}\cdots l_{m-1}P_m$. One matrix with period P_1 is decomposed taking $O(N^4 l_2 l_3 \cdots l_{m-1}P_m)$ time, l_1 matrices with period P_2 are decomposed each taking $O(N^4 l_3 l_4 \cdots l_{m-1}P_m)$ time, and so on up through $l_1 l_2 \cdots l_{m-2}$ matrices, each taking $O(N^4 l_{m-1}P_m)$ time. Thus the total complexity for decomposition is no greater than $O(m l_1 l_2 \cdots l_{m-1}N^4)$, or equivalently, $O(mSN^4)$. Since the complexity of decomposition is greater than the complexity of applying the basic TSA algorithms to the resulting submatrices, the total complexity of scheduling traffic with nested periods is $O(mSN^4)$ when Sasaki and Jain's algorithm is used for decomposition.

Ganz and Gao [5] provide an algorithm that can decompose a traffic matrix D into submatrices in $O(z^2 \max_i\{\lceil T/w \rceil, R_i, C_i\})$ time where z is the number of nonzero entries in the matrix D, w is the maximum number of nodes that can be connected in one slot, T is the sum of all traffic denoted by the matrix D given by $T = \sum_{i=1}^{N} \sum_{j=1}^{N} D_{i,j}$, R_i is the ith row sum of D, and C_i is the ith column sum of D. For an $N \times N$ traffic matrix with period P_k, we have that $w = N$, $T \leq N l_k l_{k+1} \ldots l_m$, $R_i \leq l_k l_{k+1} \ldots l_m$, $C_i \leq l_k l_{k+1} \cdots l_m$, and $z \leq N^2$. Thus to decompose a period P_k matrix into submatrices, we require at most $O(N^4 l_k l_{k+1} \ldots l_m)$ time. To decompose all traffic matrices in a particular system, we need $O(N^4 l_1 l_2 \cdots l_m) + l_1 O(N^4 l_2 l_3 \cdots l_m) + \cdots + l_1 l_2 \cdots l_{m-2} O(N^4 l_{m-1} l_m)$, or equivalently $O(mSN^4)$ time. So, like the algorithm provided by Sasaki and Jain, the total complexity for scheduling periodic traffic with nested periods is $O(mSN^4)$.

3 A HEURISTIC SCHEDULING ALGORITHM

Due to the high complexity of the Nested Period Scheduling algorithm, it is useful to consider simpler, heuristic scheduling algorithms that can provide quality of service guarantees. One variation of a heuristic algorithm presented by Philp and Liu [1] called Slot-by-Slot, Earliest-Deadline-First, Earliest-Arrival-First, or SS-EDF-EAF, is described as follows. In the SS-EDF-EAF scheduling algorithm, packets are assigned to time slots one slot at a time. For a particular time slot, packets are considered in the following order of priority: earliest deadline first, and in the case of ties, earliest arrival time first. In the case of further ties, the order is arbitrary. When a particular packet is considered, it is scheduled in the slot if no other packets with the same input or the same output are already scheduled in the slot. This completes the specification of the algorithm. For convenience, assume that the first packets arrive at time slot 0.

The following theorem gives a performance guarantee for the SS-EDF-EAF algorithm, and is proven in the appendix.

Theorem 3.1 *For an $N \times N$ switch, periodic traffic having link utilization no larger than* 1/14 *can be scheduled by the SS-EDF-EAF algorithm.*

While the utilization, 1/14, required by the theorem is relatively small, the algorithm could be useful in a multiple priority system where a relatively small amount of periodic, high priority traffic could be scheduled with SS-EDF-EAF, while lower priority traffic is placed in slots left empty by the SS-EDF-EAF algorithm.

4 MORE GENERAL TRAFFIC

The previous sections focused on scheduling periodic traffic, a special case of deterministically constrained traffic. In this section, the application of the Nested Period Scheduling algorithm and the Slot-by-Slot, Earliest-Deadline-First, Earliest-Arrival-First algorithm to more general deterministically constrained traffic is explored. In particular, the requirement that packets arrive periodically is relaxed so that any number of packets up to the maximum allowed for a particular stream can arrive in any single slot. This special form of deterministically constrained traffic called (α, S) is described below. Then it is shown that the NPS algorithm can be used to find a schedule for (α, S) traffic when the maximum link utilization is at most 1/4. Finally, it is shown that the SS-EDF-EAF algorithm can find a schedule for (α, S) traffic when the maximum link utilization is at most 1/14.

A. Deterministically Constrained (α, S) *Traffic Model*

The (α, S) traffic model is described as follows. Each message stream, M_l, has a corresponding vector $(\alpha_l, S_l, r_l, c_l)$. The interpretation is that M_l is a stream of packets originating at input r_l to be sent to output c_l such that at most $\alpha_l S_l$ packets arrive in any S_l consecutive slots. It will be assumed that S_l and $\alpha_l S_l$ are integers. For (α, S) traffic, the maximum utilization at any input link, i, is given by $\sum_{\{l|r_l=i\}} \alpha_l$. The maximum utilization at any output link is defined similarly. Packets must depart in one of the S_l consecutive slots, beginning with the slot in which they arrive.

We assume without loss of generality that $\alpha_l S_l = 1$ for each l because if $\alpha_l S_l = C > 1$ for some l, then the message stream M_l could be replaced by C streams each having $(\alpha_l', S_l, r_l, c_l)$ traffic where $\alpha_l' = \alpha_l / C$. A message stream with $\alpha_l S_l = C > 1$ can be thought of as C interleaved message streams with $\alpha_l S_l = 1$. A schedule exists for (α, S) traffic when no deadlines are missed. Thus the (α, S) traffic is different from periodic traffic only in that at *most* one arrival occurs every S slots rather than *exactly* one arrival occurring every S slots. Equivalently, the interarrival times for (α, S) traffic are at least S slots rather than exactly S slots.

B. Nested Period Scheduling Algorithm Applied to Deterministically Constrained Traffic

As in the case of periodic traffic with asynchronous, non-nested periods, the service-oriented view of the Nested Period Scheduling algorithm discussed in Section 2 can be applied to (α, S) traffic. A corollary to Theorem 2.1, provided below, shows that the NPS algorithm can find a schedule for (α, S) traffic when the maximum link utilization does not exceed 1/4. As in Section 2, the approach is for each message stream to report to the NPS algorithm that it is a periodic message stream having period 2^r with r small enough that at least one type 2^r service interval is completely contained in any interval of length S_l. Since the reported periods must all be powers of two, they are nested, and thus the traffic can be scheduled by the NPS algorithm if the utilization calculated with the reported period lengths does not exceed one.

Corollary 4.1 *For an $N \times N$ switch, (α, S) traffic with maximum link utilization no greater than $1/4$ can be scheduled by the NPS algorithm.*

The idea of the proof of Corollary 4.1 is that a message stream with parameter S_l declares to the NPS algorithm that it is a periodic source with period $\tilde{P}_l = \lfloor (S_l + 1)/2 \rfloor_2$, where $\lfloor x \rfloor_2$ is the greatest power of two less than or equal to x. As before, it is easy to see that this choice of $\tilde{P}_l$ insures that any S_l consecutive slots contain a complete type $\tilde{P}_l$ service interval. Hence, the message stream will be assigned a potential departure by the NPS algorithm at least once in every S_l slots as required.

Assume that the maximum link utilization does not exceed 1/4 for the (α, S) traffic. Then, in particular, $\sum_{\{l|r_l=i\}} \alpha_l \leq \frac{1}{4}$ for each input link i. Using the fact that $S_l/\tilde{P}_l$, the utilization at each input link i can be written

$$\sum_{\{l|r_l=i\}} \frac{1}{\tilde{P}_l} \leq 4 \sum_{\{l|r_l=i\}} \alpha_l \leq 1 \qquad (4.1)$$

The same results hold for each output link.

Thus by having each message stream request an appropriate periodic message stream, the resulting utilization for all links is no larger than one when the (α, S) link utilization is at most 1/4. Therefore, by Theorem 2.1, the (α, S) traffic can be scheduled using the service-oriented interpretation of the NPS algorithm.

C. A Bound on Utilization for Scheduling (α, S) Traffic with SS-EDF-EAF

The results of Section 3 can be extended to a switching system with (α, S) traffic. As in the proof of Theorem 3.1 provided in the Appendix, it is required that a packet from message stream M_l departs within $\lceil \gamma S_l \rceil$ slots of its arrival time slot for some fixed $0 < \gamma < 1$. The maximum number of packets that can block a particular packet from being scheduled with SS-EDF-EAF is calculated and shown to be less that $\lceil \gamma S_l \rceil$ for each l when the maximum link utilization does not exceed $\frac{1}{14}$ and $\gamma = 1/2$.

Theorem 4.1 *For an $N \times N$ switch having (α, S) traffic with maximum link utilization no larger than* 1/14*, the SS-EDF-EAF algorithm will produce a schedule.*

We assume that there are m message streams with $\alpha_l S_l = 1$ for each $l \in \{1, \ldots, m\}$. Using the fact that (α, S) traffic has interarrival times of S or more slots, the proof of Theorem 3.1 can be directly applied to Theorem 4.1 by noting that the bound (A.1) holds for (α, S) traffic as well.

5 CONCLUDING REMARKS

This paper has explored the Multiple-Period Time Slot Assignment problem. In particular, the Nested Period Scheduling algorithm has been presented and shown to find a schedule for periodic traffic with nested periods and link utilization at most one. While the Philp and Liu conjecture remains open, we have shown that the Philp and Liu conjecture holds for synchronized, nested periodic traffic.

More importantly, the Nested Period Scheduling algorithm has been shown to schedule (α, S) traffic and periodic traffic with *any* periods when the link utilization is at most 1/4. A simple algorithm, Slot-by-Slot, Earliest-Deadline-First, Earliest-Arrival-First, has been shown to schedule (α, S) traffic and periodic traffic when the link utilization is at most 1/14. The link utilization conditions may seem unacceptably small when applied to traffic that is not periodic with nested periods. However, the algorithms could be useful if applied to a two-priority system with a small amount of high priority traffic that must be guaranteed to meet deadlines. The high priority traffic could be scheduled with NPS or SS-EDF-EAF and the low priority traffic could take any remaining available slots.

Future work might include finding a more efficient algorithm for scheduling nested periodic traffic. Also, other simple algorithms capable of finding a schedule for periodic traffic with arbitrary period lengths or (α, S) traffic with link utilization larger than 1/14 could be investigated. Finally, further work should be done to determine if the Philp and Liu conjecture is true.

A Appendix

Theorem 3.1 is proved in this appendix.

For some $0 \leq \alpha \leq 1/14$, assume that the maximum link utilization is less than or equal to α. Fix some $\gamma, 0 < \gamma < 1$. For $r \in Z^+$, consider the following statement

> $S(r)$: Any packet of period P_k for $k \in \{1, \ldots, m\}$ arriving in slot r or earlier is scheduled by the SS-EDF-EAF algorithm no later than slot $r + \lceil \gamma P_k \rceil - 1$.

Trivially, $S(-1)$ is true. We assume that $S(r-1)$ is true for the sake of a proof by induction. A packet is said to be *blocked* in a particular slot if another packet originating at the same input or terminating at the same output has priority and is scheduled in the slot. For a tagged packet arriving in slot r going from input i_o to output j_o with period l, the maximum number of slots in the set $\{r, r+1, \ldots, r + \lceil \gamma l \rceil - 1\}$ in which the tagged packet could be blocked by a period p message stream that originates at input i_o or terminates at output j_o, denoted $B_{lp\gamma}$, is bounded by

$$B_{lp\gamma} \leq \begin{cases} 1 + \gamma + \frac{\gamma l + 1}{p} & : \quad p - \lceil \gamma p \rceil < l \\ 0 & : \quad p - \lceil \gamma p \rceil \geq l \end{cases} \tag{A.1}$$

This bound for $B_{lp\gamma}$ is found as follows. By the induction hypothesis, a packet from the period p message stream arriving in slot $r - \lceil \gamma p \rceil$ or earlier will depart before slot r, and cannot block the tagged packet. A packet from the period p message stream that arrives in slot $r + \lceil \gamma l \rceil$ or later also cannot block the tagged packet. This leaves only $\lceil \gamma p \rceil + \lceil \gamma l \rceil - 1$ possible slots for the arrival of blocking packets, and these slots are consecutive, so that

$$B_{lp\gamma} \leq \left\lceil \frac{\lceil \gamma p \rceil + \lceil \gamma l \rceil - 1}{p} \right\rceil \tag{A.2}$$

which by the identities, $\lceil x \rceil \leq x+1$ and $\lfloor x \rfloor \leq x$, yields (A.1) in the case $p - \lceil \gamma p \rceil < l$. As noted above, any packet from the period p stream that can block the tagged packet must arrive in slot $r - \lceil \gamma p \rceil + 1$ or later. But if $p - \lceil \gamma p \rceil \geq l$, then the deadline of such a packet would be strictly later than the deadline of the tagged packet, so no such packet exists. Thus (A.1) is established.

Now examine the total number of slots in which the tagged packet could be blocked by summing over all streams using either input i_o or output j_o. This total number of potential blocking slots, B_T, for the tagged packet with period l is bounded by

$$B_T \leq \left(\sum_{i=1}^{N} \sum_{\{k|P_k - \lceil P_k \gamma \rceil < l\}} x_{ij_oP_k} B_{lP_k\gamma} \right) + \left(\sum_{j=1}^{N} \sum_{\{k|P_k - \lceil P_k \gamma \rceil < l\}} x_{i_ojP_k} B_{lP_k\gamma} \right) - 2B_{ll\gamma} \qquad \text{(A.3)}$$

Using bound (A.1) for $B_{lp\gamma}$ yields

$$\begin{aligned} B_T &\leq (\gamma+1) \left(\sum_{i=1}^{N} \sum_{\{k|P_k \leq \frac{l}{1-\gamma}\}} x_{ij_oP_k} + \sum_{j=1}^{N} \sum_{\{k|P_k \leq \frac{l}{1-\gamma}\}} x_{i_ojP_k} \right) \\ &+ \left(\sum_{i=1}^{N} \sum_{k=1}^{m} x_{ij_oP_k} \frac{\gamma l + 1}{P_k} \right) + \left(\sum_{j=1}^{N} \sum_{k=1}^{m} x_{i_ojP_k} \frac{\gamma l + 1}{P_k} \right) - 2(1 + 2\gamma + \frac{1}{l}) \qquad \text{(A.4)} \end{aligned}$$

Since the maximum link utilization is less than or equal to α, it is true that

$$\sum_{i=1}^{N} \sum_{k=1}^{m} \frac{x_{ij_oP_k}}{P_k} \leq \alpha \quad \text{and} \quad \sum_{j=1}^{N} \sum_{k=1}^{m} \frac{x_{i_ojP_k}}{P_k} \leq \alpha$$

Thus B_T is bounded by

$$\begin{aligned} B_T &\leq (\gamma+1) \left(\sum_{i=1}^{N} \sum_{\{k|P_k \leq \frac{l}{1-\gamma}\}} x_{ij_oP_k} \frac{l}{P_k(1-\gamma)} + \sum_{j=1}^{N} \sum_{\{k|P_k \leq \frac{l}{1-\gamma}\}} x_{i_ojP_k} \frac{l}{P_k(1-\gamma)} \right) \\ &+ 2\alpha(\gamma l + 1) - 2 \\ &\leq 2\alpha(\gamma+1)\frac{l}{1-\gamma} + 2\alpha(\gamma l + 1) - 2 \qquad \text{(A.5)} \end{aligned}$$

Since we want to show that the tagged l period packet can be scheduled in one of the $\lceil \gamma l \rceil$ slots beginning with its arrival time slot, we must show that $B_T \leq \lceil \gamma l \rceil - 1$, which is true if

$B_T \leq \gamma l - 1$ because $\gamma l \leq \lceil \gamma l \rceil$. So if

$$\alpha \leq \frac{\gamma + \frac{1}{l}}{2\gamma + \frac{2(\gamma+1)}{1-\gamma} + \frac{2}{l}} \tag{A.6}$$

then the tagged l period packet can be scheduled within $\lceil \gamma l \rceil$ slots of its arrival time slot. Noting that $l \geq 1$,

$$\alpha \leq \frac{\gamma}{2\gamma + \frac{2(\gamma+1)}{1-\gamma} + 2} \tag{A.7}$$

implies α satisfies Equation (A.6). We want the highest possible utilization so we maximize α in Equation (A.7) and obtain $\gamma = 1/2$, the optimal value of γ.

Substituting $1/14$ for α and $1/2$ for γ in Equation (A.5) yields

$$B_T \leq \frac{1}{14}l + \frac{1}{7} + \frac{3}{7}l - 2 = \frac{1}{2}l - \frac{13}{7} \leq \lceil \gamma l \rceil - 1 \tag{A.8}$$

The tagged period l packet thus departs within $\lceil \gamma l \rceil$ slots of its arrival time slot. Therefore, $S(r)$ is true. Thus $S(r)$ is true for all r by induction when the maximum link utilization is at most $1/14$ and $\gamma = 1/2$. Therefore, for an $N \times N$ switch, periodic traffic having maximum link utilization at most $1/14$ can be scheduled by the SS-EDF-EAF algorithm. Theorem 3.1 is proved.

References

[1] I. Philp and J. Liu, “A switch scheduling problem for real-time periodic messages,” UIUC Preprint 1996.

[2] C. L. Liu and J. Wayland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *IEEE Transactions on Communications*, vol. 20, pp. 46–61, January 1973.

[3] T. Inukai, "An efficient SS/TDMA time slot assignment algorithm," *IEEE Transactions on Communications*, vol. 27, pp. 1449–1455, October 1979.

[4] A. Ganz and Y. Gao, "Efficient algorithms for SS/TDMA scheduling," *IEEE Transactions on Communications*, vol. 40, pp. 1367–1374, August 1992.

[5] A. Ganz and Y. Gao, "Time-wavelength assignment algorithms for high performance WDM star based systems," *IEEE Transactions on Communications*, vol. 42, pp. 1827–1836, February/March/April 1994.

[6] G. Sasaki and R. Jain, "Scheduling data transfers in preemptive hierarchical switching systems," Preprint 1994.

[7] I. S. Gopal and M. A. Bonuccelli, "An optimal switching algorithm for multibeam satellite systems with variable bandwidth beams," *IEEE Transactions on Communications*, vol. 30, pp. 2475– 2481, November 1982.

Stream (M_l)	r_l - input	c_l - output	p_l - period
M_1	1	1	2
M_2	1	2	4
M_3	2	1	2
M_4	2	2	8
M_5	2	2	8
M_6	1	2	8
M_7	2	2	4

Figure 1: A periodic message set

		Slot Number							
Input	Stream	1	2	3	4	5	6	7	8
1	M_1	1		1		1		1	
1	M_2		2				2		
1	M_6				2				
2	M_3		1		1		1		1
2	M_4	2							
2	M_5							2	
2	M_7			2		2			

Figure 2: A feasible schedule for the message set in Figure 1

```
Input: Distinct periods $P_1 > \ldots > P_m$ and corresponding traffic matrices $D_1, \ldots, D_m$
Output: Schedule meeting all deadlines
begin:
          Let there be exactly one *composite* traffic matrix containing period $P_1$ traffic
          and let the matrix equal $D_1$.
          For $k = 1$ to $m - 1$ do:
          begin:
              Calculate the aggregate link utilization, $u_l$ where $l \in \{1, \ldots, 2N\}$ indexes
              the rows first and then the columns, for $D_{k+1}, \ldots, D_m$.
              Set $f_l = P_{k+1}(1 - u_l)$ for $l \in \{1, \ldots, 2N\}$. (Then $f_l$ is the residual line
              capacity per length $P_{k+1}$ period if all traffic with periods $P_{k+1}, \ldots, P_m$
              is scheduled in a period of length $P_{k+1}$).
              Use a generalized TSA algorithm to decompose each of the composite
              traffic matrices representing traffic with periods $P_1, \ldots, P_k$ into at most
              $P_k/P_{k+1}$ matrices with maximum line sums $f_l$, $l \in \{1, \ldots, 2N\}$. (See
              Lemmas 2.1 and 2.2.)
              Combine each of these traffic matrices with a copy of the matrix having
              period $P_{k+1}$ to obtain a composite matrix representing traffic with pe-
              riods $P_1, \ldots, P_{k+1}$. (This traffic will eventually be scheduled in a $P_{k+1}$
              period.)
          end
          Schedule each of the composite matrices representing traffic with periods
          $P_1, \ldots, P_m$ with a classic TSA algorithm.
end
```

Figure 3: The Nested Period Scheduling algorithm

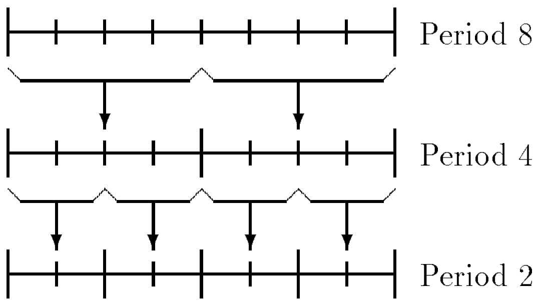

Figure 4: Traffic arrangement under the NPS algorithm