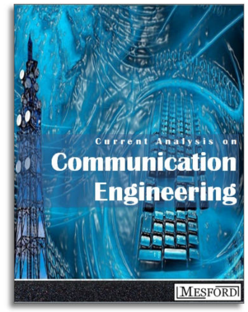
Probabilistic Timing Analysis of Periodic Traffic Streams in a One-switch Network with Random Interference



Michael Short* and Muneeb Dawood

School of Science, Engineering and Design, Teesside University, Middlesbrough, UK, TS1 3BA, UK.

Abstract:

In this paper the timing properties of a single switch LAN are analyzed. The analysis is based upon relatively simple iterative algorithms to analyze the duration of the synchronous busy period of a message set, assuming FIFO buffering in both the source nodes and switch are employed. In this paper, the real-time traffic (periodic and sporadic) is also assumed to be subject to random interference from other sources, and a probabilistic stance is taken. A number of observations are made based upon our initial analysis and investigations, and preliminary algorithms are proposed to probabilistically estimate the worst case queuing delays at source nodes and switch output ports assuming some knowledge of the (mean) interference levels are known. The work was principally motivated by the need for easy-to-apply and relatively accurate probabilistic timing analysis in distributed automation implementations; it may also be applicable to other industrial contexts. The paper concludes that the techniques may be able to provide a useful building block for larger packet switched networks with deterministic and stochastic traffic sources, and future work will consider extensions to multiple switch hierarchical networks.

Publication History: Received: 15 August 2018 | Revised: 26 September 2018 |Accepted: 01 October 2018

Keywords:
Probabilistic timing analysis; Packet switched networks; Traffic models; Random Interference.

INTRODUCTION

Recent trends have seen an increase in the use of packet-switching technologies for the implementation of simple distributed (and possibly embedded) networks for sensing and control applications [1][2]. Providing guarantees of timely delivery in packet-switched networks is a complicated problem, as the worst-case delays incurred across multiple hops in the network must be derived. Of course it is possible to employ packet scheduling techniques, e.g., Earliest Deadline First (EDF) [3]. This can simplify the overall analysis problem, and by having appropriate admission controls a flexible yet predictable network may be implemented.

On the other hand, in some applications (e.g. factory automation and domotics), the use of specific packet scheduling algorithms may not be practical since many standard packet-switching network components and technologies only support simple First Come First Served (FCFS) scheduling. Previous work has examined the timing properties of networks scheduled using FCFS under the assumption that traffic is implemented as a number of periodic streams [1][2]. For reasons discussed in [2], it can become very complex when applying techniques such as network calculus when traffic is periodic and FCFS is employed. In this paper, we wish to analyze the timing properties of simple packet switched LANs (using techniques similar to those of [2]) in which the real-time traffic is principally periodic and/or sporadic. However, it also considers that unpredictable interference - in the form of frames arriving from other sources in a random fashion with known mean – is present. Since this interference is random, probabilistic timing guarantees are appropriate; in this compact article, we report some initial investigations and findings. The remainder of the paper is structured as follows. In Section II, our assumptions on the network topology and traffic models are given, and Section III discusses the probabilistic calculation of queuing delays and buffer size requirements. Section IV concludes the article and discusses areas for future improvements.

2. NETWORK AND TRAFFIC ASSUMPTIONS AND MODELS

2.1. Single-Switch Network Section

Firstly, we assume that time is discrete and occurs in integer multiples of a global clock which has a resolution equal of δ (typically this would be the homogenous network bit-time).

*Address correspondence to this author at the School of Science, Engineering and Design, Teesside University, Middlesbrough, UK, TS1 3BA, UK.
E-mail: M.Short@tees.ac.uk

Mesford Publisher Inc
Office Address: Suite 2205, 350 Webb Drive, Mississauga, ON L5B3W4, Canada; T: +1 (647) 7109849 | E: cace@mesford.ca, contact@mesford.ca, https://mesford.ca/journals/cace/

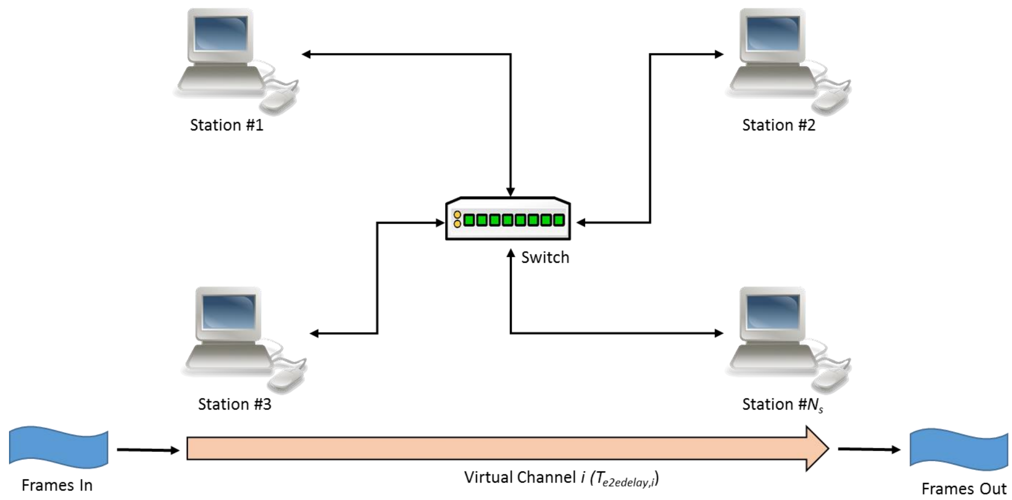


Fig. (1). Example of a single switch network such as that under consideration.

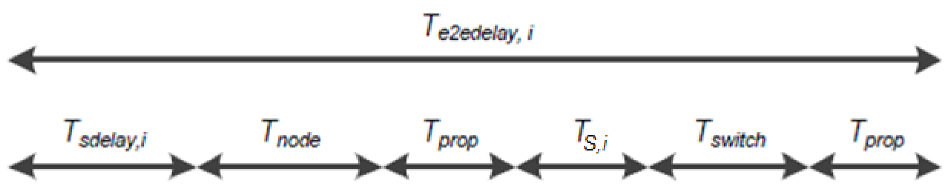


Fig. (2). Sources of delay in the single-switch network.

For simplicity, we assume henceforth that δ = 1. The next assumption is that the LAN infrastructure under analysis consists of a number of stations connected via a single network switch or router. We assume that if a router is present, it only carries out very simple low-level routing and so may be effectively treated as a standard FIFO-buffered switch. For simplicity, we will also assume that the switch is homogenous, i.e. that the incoming/outgoing bit rate of each port is identical. Although this is a relatively simple network to consider, it provides a representative starting point and allows for straightforward extension. We assume that the network segment to be analyzed consists of Ns stations connected to the single switch, which has Np active ports such that Ns ≤ Np. We assume that the system to be implemented is described as a number Nc of virtual channels, with each channel mapping a logical path between a source and destination station through the switch. This situation is depicted in Fig. (**1**).

Using similar terminology as in [1] and [2], the total worst-case end-to-end delay for any virtual channel i (denoted as Te2edelay,i) is assumed to be comprised of several sub-sources of delay as shown in Fig. (**2**).

The components making up the Te2edelay,i delay are as follows: Tsdelay,i represents the worst-case delay at the source node, and is principally due to queuing whilst awaiting access to the Network Interface Card (NIC). This delay is node-dependent as heterogeneity of nodes is assumed. Tnode represents the worst-case latency for a frame in the head of the queue to leave the source node (e.g. due to non-preemption), and depends upon several node-dependent factors which will be subsequently described. Tprop represents the propagation delay over the physical link (we assume that connection cables are of identical length; an assumption which is easily lifted if required). TS,i represents the worst-case delay at the switch, and is principally due to buffering whilst awaiting access to the output port. Finally, Tswitch is the worst case latency for a frame in the head of the queue to leave a switch/port. In this paper, as in [1] and [2], we are principally concerned with determining the source node delay Tsdelay,i and the switch delay TS,i. In the analysis that follows, we are more closely follow that developed in [2] as the work of [1] – whilst being simpler in its formulation than – makes some pessimistic assumptions (e.g. that all stations may simultaneously transmit to any other station) and restricts key parameters (e.g. that periods are all greater than the worst-case transmission delay). For the remainder of the discussion, we assume that time (which is represented by t) is continuous, real-valued and non-negative. Next, we outline the deterministic and random traffic models that are employed in this preliminary study.

2.2. Periodic/Sporadic Traffic

For periodic/sporadic channels, let each channel τi ∈ Γ be represented by the 4-tuple:

$$\tau_i = \left(S_i, D_i, T_i, C_i\right). \tag{1}$$

In which Si is the (integer) source station identifier and Di is the (integer) destination station identifier. For simplicity, assume that station identifiers are identical to the port numbers (i.e. station 1 is connected to port 1, station 2 is connected to port 2, and so on). Ti ∈ ℵ+ represents the period/minimum inter-arrival time of the channel and Ci ∈ ℵ+ is the worst-case transmission time of any message frame generated by the channel (each invocation of the channel is called a message frame or simply frame). Let the kth frame generated by channel i be denoted as τi,k. Successive frames generated by sporadic channels are always separated by at least Ti units of time; successive frames generated by periodic channels are always separated by exactly Ti time units. Since it is known that the worst-case manifestation of a sporadic message stream is the pattern in which the minimum inter-arrival times are always adhered to (and the stream effectively becomes periodic) [3], periodic/sporadic streams will henceforth be referred to as simply periodic for ease of exposition. For periodic streams, the worst-case cumulative workload generated by stream i (denoted as wi) in the interval [0, t) can be calculated using [2]:

$$w_i(t) = \left(\left\lfloor \frac{t}{T_i} \right\rfloor + 1 \right) \cdot C_i \tag{2}$$

2.3. Random Traffic

Many types of network traffic are essentially random in nature; it is well known that in some circumstances frame inter-arrival times are well-modeled as exponential or geometric distributions [4, 5]. Let us assume that each source station sends and receives random traffic, however this random traffic does not have a specific destination (in the case of traffic generated by a node) or a specific source (for traffic received by a node). In addition, we assume that the payload length is unknown, and make the assumption that any random frames processes by node i will always have length ≤ Ĉi ∈ ℵ+ (typically, Ĉi will be set to the worst-case payload length allowed by the protocol for all nodes). The final assumption that we make is that the system is not closed, in the sense that random traffic in the network is just sent and received by the stations connected to the switch; an external gateway may be present, hence the total mean traffic sent and received by all nodes is not necessarily equal. Let the inter-arrival times of the random traffic generated by node i be geometrically distributed with a mean $\bar{T}_i^+ \in \Re+$, and the inter-arrival times of traffic received by node i also be geometrically distributed with mean $\bar{T}_i^- \in \Re+$. Then for each station i, the parameters of the random traffic can be described by the 3-tuple θi ∈ Π:

$$\theta_i = \left(\bar{T}_i^+, \bar{T}_i^-, \hat{C}_i \right) \tag{3}$$

Overall, the network and source models described may occur in a small-scale home or industrial automation application where process information and control traffic may co-exist with other traffic (e.g. email internet) that is best described by random attributes. Given a set of stations, virtual channels and a probability R ∈[0.5, 1), we are interested in obtaining tight probabilistic bounds on the worst-case transmission delay that each channel may experience. Also, we are interested in the required source and switch buffer sizes such that probability that these timing or buffer bounds become violated is guaranteed to be ≤ (1-R). In order to determine this information efficiently, we require an upper bound on the expected number of packet arrivals for a random traffic stream in some interval of time, for a given confidence probability. Since time is discrete and the distribution of arrival times assumed to be geometric, the probability of a packet arrival at each individual time step - p - is equal to $1/\bar{T}$, where $\bar{T}$ is the mean inter-arrival. If the number of packet arrivals occurring in t consecutive time-steps ('independent trials') is given by the variable X, then X follows a Binomial distribution with parameters t and p. To obtain the bound with confidence R, we therefore seek to evaluate the Rth quantile of X; since obtaining the exact quantile requires a (non-trivial) iterative search over the Binomal distribution function [5], we shall instead use the following upper tail quantile inequality that was recently proven:

Theorem 1: Let η(t, p, R) represent the Rth quantile of a Binomially distributed random variable comprising t identical and independent Bernoulli variables, each having an individual probability of success p∈(0, 1). Then defining the quantity C(R) = $\sqrt{-2\text{In}(1-\text{R})}$ for R∈[0.5, 1) an easily computable and asymptotically tight upper bound on η(t, p, R) is given by:

$$\eta(t, p, \text{R}) \le \left\lceil tp + C(\text{R})\sqrt{tp(1-p)} + \frac{C(R)^2}{6} \right\rceil \tag{4}$$

Proof: Short & Proenza 2013 (see [5], Corollary 1).

When carrying out a timing analysis, expression (4) will normally have to be evaluated many times for different values of t. Since we assume that both p =$1/\bar{T}$ and R are known (as a basic assumption we could take, for example, R = 0.999 for 99.9 confidence), to simplify the repeated computation of (4) for a particular link an easy simplification is to first calculate the two quantities C1 = $\sqrt{-2\text{In}(1-\text{R})(1-\rho)}$ and C2 = -ln(1-R)/3. A bound on the outgoing workload generated by random traffic originating in station j in the interval [0, t) (denoted as wj+) for a confidence probability R is thus given by:

$$w_j^+(t) = \left\lceil \frac{t}{\bar{T}_j^+} + C_1\sqrt{\frac{t}{\bar{T}_j^+}} + C_2 \right\rceil \cdot \hat{C}_j \tag{5}$$

Where Ĉi is, as discussed above, the worst-case frame length of random traffic. This is a simple closed-form expression which is easily computed for any of the input parameters in constant time; note that the probability that the actual workload exceeds that computed by (5) in the specified interval is formally guaranteed to be ≤ (1-R) [5]. Now, the worst-case incoming workload for station j (denoted as wj-) can be obtained from (5) with appropriate replacement of $\bar{T}_i^+$ with $\bar{T}_i^-$ and adjustment of C1. Expression (5) seems simple enough to allow an adaption of the 'busy period' analysis methods developed by Fan et al. in [2] to be adapted to the case of stochastic traffic; however there are several points that require attention prior to developing a suitable analysis.

3. BUSY PERIOD ANALYSIS

3.1. Analysis of Periodic Streams

For purely periodic streams, several key results were established in [2]; in the context of the current work, the two main points that were proven were as follows. When determining the delays due to FIFO buffering of frames at the source stations, the synchronous arrival case (all periodic streams arrive simultaneously at t = 0) is the worst possible. In this case, the worst-case queuing delay of stream i (denoted as

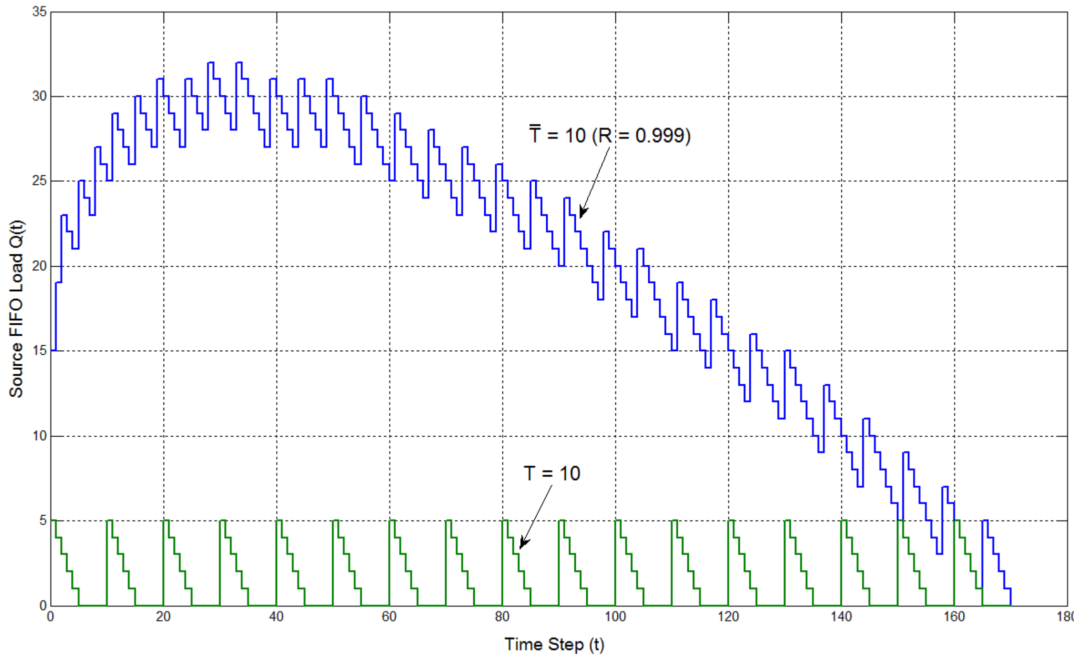


Fig. (3). Comparison of FIFO delay in a source node.

DSi) is finite if and only if the utilization of the outgoing channel from station i is not overloaded, i.e. has a utilization bounded as follows:

$$\sum_{\substack{\tau_j \in \Gamma \\ S_j = i}} \frac{C_j}{T_j} \leq 1 \tag{6}$$

And has a value given by:

$$DS_i = \sum_{\substack{\tau_j \in \Gamma \\ S_j = i}} C_j \tag{7}$$

Where the summation limits of (6) and (7) are such that every valid channel in the set Γ which a source station identifier of j are included in the summation. Secondly, when determining the delay due to FIFO buffering of incoming frames at the output port i of a single switch, the synchronous arrival case (again in which all periodic streams arrive simultaneously at t = 0 in each of the source nodes) is the worst possible. The worst-case queuing delay in this situation is found during the initial busy period, which can be obtained as the smallest (positive) solution of the following equation (which is iterated from t = 0):

$$WP_i(t) = \sum_{\substack{\tau_j \in \Gamma \\ D_j = i}} w_j(t) = t \tag{8}$$

Where WPi(t) represents the workload presented to stream i t time units after the synchronous arrival pattern. This busy period has finite length if and only if the utilization of the incoming channel from station i is not overloaded, i.e. has a utilization bounded as follows:

$$\sum_{\substack{\tau_j \in \Gamma \\ D_j = i}} \frac{C_j}{T_j} \leq 1 \tag{9}$$

Clearly, for the network model presented in the previous Section, if there is no random traffic then these techniques would suffice to determine the worst-case delays. When random traffic is included in the analysis, however, it is not immediately obvious the extent to which these results are still relevant. In the next two Sections, the focus will be upon delay analysis in the source and switch output ports.

3.2. Source Node Queuing Delay

Firstly, we have observed that worst-case queuing delay in a source node no longer occurs at t = 0 due to the non-periodic nature of the workload function (5); this is highlighted by the following simple example. In Fig. (**3**), we compare the source node FIFO queue size of a single periodic channel Q(t) with {T = 10, C = 5} with a channel experiencing only random traffic with $\{\overline{T}^{+} = 10, \hat{C} = 5\}$. A confidence probability R = 0.999 was used in the latter; the constants C1 and C2 required for (5) were computed as 3.526 and 2.303 respectively. The results were obtained using a standard personal computer running a simple C++ application. The plot shows the comparison between t = 0 and t = 170, the point in time in which the queue size for the random traffic drops to zero indicating the end of the busy period. Clearly in the latter case the queue size is first increasing from Q(0) = 15 and first peaks at the maximum value of Q(28) = 32, before starting to decrease (non-monotonically) at t = 33. This indicates that it is likely to also need to employ busy period analysis in the source nodes when random traffic is present, unlike in the purely periodic case.

For the workload arrival function (5), it was shown in [5] that the workload is non-decreasing in t and eventually approaches (but does not exactly converge upon) the mean workload $t\hat{C}/\overline{T}^{+}$. This implies that the worst-case queuing delay can be obtained by examining the synchronous busy period for periodic tasks under the assumption that the random traffic also starts to arrive at t = 0. However, we first need to consider under what conditions the resulting busy period will have a finite length; a necessary condition is clearly that the utilization of the channel (including the mean utilization $\hat{C} / \bar{T}^{-}$ of the random traffic) does not exceed unity. Unfortunately this condition is not sufficient, as choosing an example in which there is no periodic traffic and any $\hat{C} = \bar{T}^{+} > 0$ with R > 0.5 can easily be verified using (5). Observing that the source traffic can be modeled by the summation of two queues (Geo/D/1 and D/D/1 in Kendall's notation [7]), it is known that such a link has a finite busy period if and only if the link utilization strictly less than one [7]. This result also holds despite the observation that the inequality (5) is not exact due to Theorem 2 in [5], which has shown that relative overestimation error in expression (4) vanishes for large t.

However, when the total link utilization is close to unity the length of the synchronous busy period becomes too large to analyze in a reasonable time and is sensitive to the choice of R. Indeed this seems to be much worse that in the purely periodic case, in which the busy period is always limited by the least common multiple (lcm) of the channel periods even for high utilizations. By limiting the effective allowable channel utilization to be less than some upper limit UM ≈ 0.99 results in tractable behavior for reliability levels in our range of interest. Therefore, in order to determine the worst-case latency the following general procedure can be used. Defining the total outgoing cumulative workload for station i at time t as Wi+(t):

$$W_i^{+}(t) = \left(\sum_{\substack{\tau_j \in \Gamma \\ S_j = i}} w_j(t) \right) + w_i^{+}(t) \tag{10}$$

Then if the busy period has length L the outgoing queue size when random traffic is also present Qi+(t), for t ≤ L, is easily computed as Wi+(t) – t. To find the worst-case delay, one finds the extrema of the function Wi+(t) – t subject to 0 ≤ t ≤ L. As it is assumed that time is discrete, then a simple iterative scheme may be used to solve the problem in a straightforward fashion with time complexity O(Ns L) and space O(Ns). Pseudo-code for such an algorithm is shown in Fig. (**4**). The operation of the algorithm may be briefly described as follows. Lines 2 and 3 initialize the variables Q and t, representing the worst-case queue size and time respectively. Line 4 checks the channel utilization and returns signaling an error if it is overloaded. There then follows a loop between lines 5 and 9, which terminates only when the cumulative workload Wi+(t) (represented by the variable W) is ≤ to t, indicating the presence of idle time and hence the end of the busy period. The workload W is updated on line 6 according to expression (10); the worst-case queuing found so far is then updated on line 7, and time advanced by the factor δ (= 1) on line 8. The worst-case queuing population is then returned on line 10.

Example 1: Suppose we have the following traffic characteristics in a source node: two periodic/sporadic streams {T = 20, C = 3} and {T = 30, C = 5} combined with a random outgoing stream { $\bar{T}^{+}$ = 10, Ĉ = 5}. Application of the algorithm described above (assuming R = 0.999) yields a worst case delay estimate of 69, which occurs at t = 240 assuming the start of the busy period at t = 0. The end of the busy period in this case occurs at t = 1131.

```
01 INPUT(i, R, Γ, Π, δt, Ns);
02 t:=0;
03 Q:=0;
04 IF (Σ_{τj∈Γ, Sj=i} Ci/Ti) + Ĉi/T̄i+ > U_Max RETURN(∞);
05 DO:
06    W:= Σ_{τj∈Γ, Sj=i} wj(t) + wi+(t);
07    Q:=MAX{Q, (W-t));
08    t:=t+1;
09 WHILE((W-t)>0);
10 RETURN(Q);
```

Fig. (4). Algorithm to determine the maximum queuing delay for a single source station.

Note that these measures of delay not only represent the latency incurred by a frame when exiting the source node, they also imply a bound on the buffer size required by the node to implement the queue [2]. If the random traffic characteristics have been correctly modeled, then the probability of either of these bounds being violated is guaranteed to be ≤ (1-R).

3.3. Switch Output Port Queuing Delay

Turning attention now to a switch output port, the analysis may progress upon the following lines. Given our assumptions upon the network topology (Fig. **1**), we may observe that the traffic leaving the switch via a given output port is essentially the input traffic destined for the corresponding station. Since Fan *et al.* [2] have shown that for periodic streams the synchronous arrival case in each of the source nodes is the worst-possible (note that this considers only the traffic to be delivered to this specific output port; other port traffic is omitted from the analysis), and we have that the interference from random traffic is maximized over smaller intervals, let us again define the total incoming cumulative workload for station i at time t as Wi-(t):

$$W_i^{-}(t) = \left(\sum_{\substack{\tau_j \in \Gamma \\ D_j = i}} w_j(t) \right) + w_i^{-}(t) \tag{11}$$

The analysis to obtain the worst-case delay may then proceed along similar lines as that developed for the traffic leaving a source node, with the following caveat; the rate at which work from the input ports of the switch can be transferred to any single output port is limited by the physical design of the switch [2]. In the case of periodic and random traffic streams operating with a discrete clock having resolution δ, this restriction can be (pessimistically) captured as follows:

Observation 1: In the case where a station does not transmit frames directly to itself (i.e. no direct loop-back), then the worst-case workload that can be transferred to the output queue of any switch port with every clock tick δ is (Np-1)δ.

Proof: Assume that during an interval of time having length δ each active port of the switch is busy processing incoming traffic. Consider any output port j. Since there is no loop-back, assuming the worst-case then at most (Np-1) ports can have incoming traffic destined to be transferred to port j. In the worst-case all input ports will be busy processing incoming traffic for port j simultaneously, hence the maximum workload transferred to queue j is (Np-1)δ.

Again observing that the switch output traffic can be represented by the summation of multiple queues (one Geo/D/1 queue and one D/D/1 queue for each channel with this destination port), the busy period will be finite if and only if the link utilization is strictly less than one [7]. The same effective allowable channel utilization limit of UM ≈ 0.99 seems to result in tractable behavior for reliability levels in our range of interest, and taking these factors into consideration leads to the simple iterative scheme for delay estimation shown in Fig. (**5**), again requiring time O(Ns L) and space O(Ns). The operation of the algorithm is almost identical to that of Fig. (**4**), with the main exception that the rate at which the workload W is updated on line 6 is rate-limited to (Np-1) per iteration (since we assume δ = 1). Note that this rate-limit does not affect the length of the busy period if Np > 1, since the same total workload is eventually delivered out of the port, but has the effect of modulating (and potentially reducing) the peaks in the queue size.

This latter point related to the effect of Np is illustrated by Fig. (**6**) below, in which the queuing delay for the same random traffic model { $\bar{T}^{-}$ = 10, Ĉ = 5} with R = 0.999 employed to create Fig. (**3**) is displayed for values of Np equal to 2, 3 and 6. The results were obtained using a standard personal computer running a simple C++ application. For Np = 2, the queue size is never greater than 1 as the rate of delivery into the buffer is the same as the rate of exit. For Np = 3, the rate of delivery into the buffer is twice the rate of exit leading to a steady increase and a peak of Q(29) = 30. As can be seen, for Np = 6 the worst-case queue delay of Q(28) = 32 is still achieved, and the overall evolution of the queuing delay begins to approach that shown in Fig. (**3**). For Np > 10, the evolution becomes identical as the worst-case rate of delivery into the buffer is no longer affected by the limit.

```
01 INPUT(i, R, Γ, Π, δt, Ns, Np);
02 t:=0;
03 Q:=0;
04 IF (\sum_{τ_j∈Γ, D_j=i} C_j/T_j) + \hat{C}_i/\bar{T}_i^- > U_{Max} RETURN(∞);
05 DO:
06  W:=W+MIN(t*(Np-1),(W-\sum_{τ_j∈Γ, D_j=i} w_j(t)+w_i^-(t)));
07  Q:=MAX{Q,(W-t));
08  t:=t+1;
09 WHILE((W-t)>0);
10 RETURN(Q);
```

Fig. (5). Algorithm to determine the maximum queuing delay at a switch output port.

Example 2: Suppose we have the following traffic characteristics in a switch with 4 ports (Np = 4): three periodic/sporadic streams {T = 30, C = 2}, {T = 50, C = 5} and {T = 100, C = 10} combined with a random incoming stream { $\bar{T}^{-}$ = 20, Ĉ = 10}. Application of the algorithm described above (assuming R = 0.999 giving C1 = 3.623 and C2 = 2.303) yields a worst case delay estimate of 114, from the extrema which occurs at t = 312 assuming the start of the busy period at t = 0. The end of the busy period in this case occurs at t = 1468.

Note again that these measures of delay not only represent the latency incurred by a frame when transiting through a switch, they also imply a bound on the buffer size required at the switch output port [2]. If the random traffic characteristics have been correctly modeled, then the probability of either of these bounds being violated is again guaranteed to be ≤ (1-R).

4. CONCLUSIONS AND FURTHER WORK

This paper has focused upon the timing properties of real-time traffic (periodic and sporadic) in a simple one-switch network that is subject to random interference. A probabilistic stance was taken, and preliminary algorithms proposed to estimate the worst-case queuing delays for a given confidence level at source nodes and switch output ports assuming the mean interference levels are known. In particular, the simple algorithms of Fig. (**4**) and Fig. (**5**) provide the means to determine delay bounds to a given confidence level. Examples 1 and 2 have been presented to illustrate (Fig. **3** and Fig. **6**) the usefulness of the developed techniques. Although the network architecture is presently somewhat restrictive, the simplicity of the algorithms is indicative that the techniques may be able to provide a useful building block for analysis of larger packet switched networks with deterministic and stochastic traffic sources.

In terms of future work, aspects of queuing theory and network calculus (see e.g. [2] and [6]) may provide a means to extend the ideas contained within this paper to obtain similar probabilistic bounds on more complex networks. The analysis techniques and algorithms presented in this paper will benefit greatly from a number of simple improvements; principally, to improve time complexity, a discrete-event solution approach (as opposed to iterating an increasing time variable) will be adopted for the solution of the algorithms of Fig. 4 and Fig. 5. Obtaining accurate estimates of the next arrival of work from the workload function (5) in a discrete-event simulation can be obtained by re-arranging the expression to predict the next arrival time of a periodic or random frame event. It is also clear that some pessimism could potentially be removed from the estimation of the port output delay by maintaining a separate FIFO for the incoming workload originating in each source node in Fig (**5**). Also, to enable more realistic structures of network to be analyzed, it is required to be able to analyze switches in which part (or all) of the incoming traffic originates from other switches and the transmission rate is non-homogenous. One further improvement that is also required is the need to model situations in which the probability of a packet arrival at each time-step is time-varying (e.g. to cater for

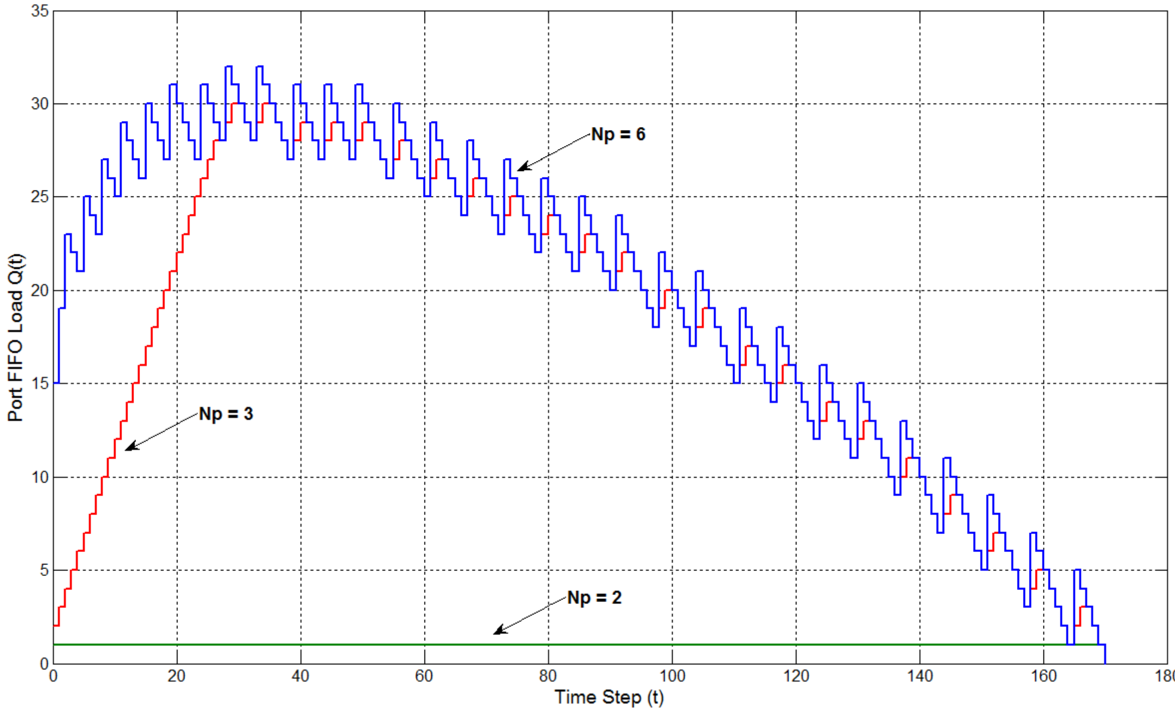


Fig. (6). FIFO delay in a switch output port with random traffic for various values of Np.

bursty arrivals of packets). For this latter point, we note that both Theorem 1 in [5] and Corollary 7 in [7] may be adapted to suit this purpose, and we leave this to future work. The quantile inequalities in this latter work will also clearly provide sharper results than provided by Equation (4), for reasons discussed in [7]. In addition, the novel methods investigated for probabilistic analysis as developed in [8] may provide scope for application in the context of the current paper.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous reviewers for their insightful comments and suggestions, which helped to improve the final quality of the paper.

REFERENCES

[1]. Lee, KC, Lee, S, Lee, MH. Worst case communication delay of real-time industrial switched Ethernet with multiple levels. IEEE Transactions on Industrial Electronics 2006; 53(5): 1669-1676.

[2]. Fan, X, Jonsson, M, Jonsson, J. Guaranteed real-time communication in packet switched networks with FCFS queuing. Computer Networks 2009; 53: 400–417.

[3]. Zhang Q, Shin, KG. On the ability of establishing real-time channels in point-to-point packet-switched networks. IEEE Transactions on Communications 1994; 42(2/3/4): 1096-1105.

[4]. Johnson, NL, Kemp, AW, Kotz, S. Univariate Discrete Distributions: Third Edition. Wiley-Interscience, 2005.

[5]. Short M, Proenza, J. Towards efficient probabilistic scheduling guarantees for real-time systems subject to random errors and random bursts of errors. In: Proc. 25th Euromico Conference On Real-Time Systems (ECRTS 2013); 2013 July; Paris, France.

[6]. Kleinrock, L. Queueing Systems. Vol I: Theory. Wiley Interscience, 1975.

[7]. Short, M. Improved Inequalities for the Poisson and Binomial Distribution and Upper Tail Quantile Functions. Probability & Statistics 2013, 2013(412958): 1-6. https://doi.org/10.1155/2013/412958.

[8]. Short, M. Bounds on Worst-Case Deadline Failure Probabilities in Controller Area Networks. Journal of Computer Networks and Communications 2016, 2016(560625): 1-12. https://doi.org/10.1155/2016/5196092.