# Delay Behavior of On-Off Scheduling: Extending Idle Periods

*Allen Roginsky[1], Ken Christensen[2], and Mehrgan Mostowfi[2]*

[1]Computer Security Division, National Institute of Standards and Technology, Gaithersburg, MD 20899
[2]Department of Computer Science and Engineering, University of South Florida, 4202 E. Fowler Ave., ENB 118, Tampa, FL 33620

**Abstract:** On-off scheduling of systems that have the ability to sleep can be used to extend system idle periods and enable greater opportunities for energy savings from sleeping. In this paper, we achieve a theoretical understanding of the delay behavior of on-off scheduling as it may apply to communications links and other systems capable of sleeping. We consider a single-server coalescing queue with a scheduler that schedules on-off periods for the server in order to extend idle periods of the downstream link. At the start of an off period (duration $T_{off}$) the server stops serving jobs immediately if idle, or after processing a job already in service. Service of any queued and arriving jobs begins at the start of the next on period (duration $T_{on}$). On and off periods are fixed. We solve for the scheduling queue behavior as a function of $T_{off}$, $T_{on}$, interarrival time $t$, service time $x$, and time of first arrival $g$ for periodic job arrivals. Our results are closed form and have both theoretical and practical significance.

## 1 Introduction

Communication systems including switches, routers, access points, links, and even entire sensor network nodes often have the capability to be placed into a low-power sleep state during idle periods to conserve energy. Computing equipment including data servers, desktop and laptop computers, and mobile computing devices can all be placed into a low-power sleep state, for example using Microsoft Windows power management capabilities. An example of a communications link that can be placed into a sleep state is XG-PON (10 Gb/s Passive Optical Network). The XG-PON standard specifically describes a cyclic sleep mode to save energy [6, 26]. Energy Efficient Ethernet (EEE) also allows for idle links to be placed into a low-power idle (LPI) mode [5]. Common to all sleep-based energy saving methods is that the transition time between sleep and awake states is non-zero. This transition time has to be accounted for in schemes or policies that determine when to enter and exit a sleep state. If we consider as an example the case of packets arriving to a switch port to be transmitted on a link (this could equally be jobs arriving to a server to be processed), these packet (or job) arrivals occur at intervals in time. Let the transition time from

awake be $t_{sleep}$ and from sleep to awake to be $t_{wake}$. The duration of an idle period, $t_{idle}$, must be longer than the sum of $t_{sleep}$ and $t_{awake}$ (that is, $t_{idle} > t_{sleep} + t_{awake}$) for sleeping to be feasible (and for energy to be saved). This motivates the idea of scheduling by coalescing – called buffer and burst in [21] and aggregation in [18] – to create extended idle periods for sleeping.

The basic idea in coalescing is to use an FCFS (First Come, First Served) queue to collect, or coalesce, multiple jobs before releasing them as a burst of contiguous jobs. We call this queue the "coalescing queue" and are interested in its behavior as a means of scheduling on and off states of a system. Coalescing effectively collects many short idle periods into a few long idle periods where the sum of the durations of the idle periods is unchanged. These extended idle periods can allow for a system to sleep when otherwise it could not. Figure 1a shows the notion of arriving jobs with interarrival times (idle periods between individual jobs) too short for sleeping, but when the jobs are coalesced (Figure 1b) the now fewer idle periods are of extended and sufficient duration for sleeping. In Figure 1a the time between job arrivals is less than $t_{sleep} + t_{wake}$ (without coalescing) and thus sleeping is not possible.

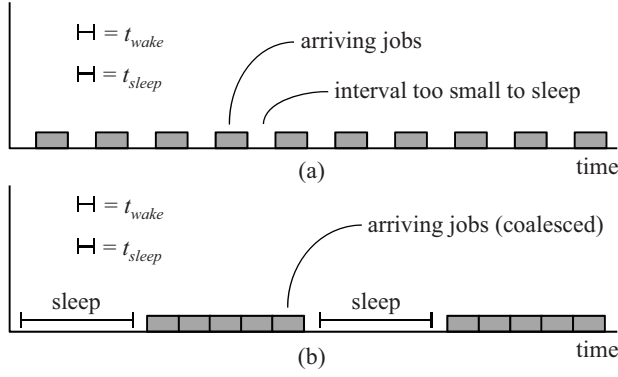* Corresponding author e-mail: christen@csee.usf.edu

**Fig. 1:** Arriving jobs (a) without and (b) with coalescing

If the durations of the coalescing period (the off period) and serving period (the on period) are predetermined, then it becomes possible to predict the energy savings that can be achieved. The energy savings is equal to the sum of all off periods less sleep and wake transition times. What is not, however, easily predicted is the increase in delay of the coalesced jobs. It is critical to have a deep understanding of the trade-off in energy savings and performance for systems that seek to be energy efficient. For example, coalescing may negatively affect the Quality of Service (QoS) of a communications system or otherwise unacceptably increase the delay of job processing in a compute server. In this paper, we develop a deterministic model to predict the bounds and mean delay for jobs in a system with periodic arrivals and fixed service times with periodic on and off periods modeling cyclic sleep. Video and voice encoding is often constant bit rate resulting in periodically generated packets. Traffic shaping also often results in packet streams in networks having periodic packet arrivals. Modeling periodic arrivals makes it possible to derive exact solutions for the parameters of interest and gain the deep insights that we seek. We will show by simulation that numerical results from our model can closely approximate those of the same system with stochastic arrivals for key cases of interest. The specific contributions of this paper are:

–A closed-form solution, as a function of the first arrival time, for the mean queue length in an interval that includes one off and one on period.
–Upper and lower bounds for the long-term mean queue length independent of the initial arrival time.
–A closed-form exact solution for the long-term mean queue length given reasonable assumptions.

The remainder of this paper is organized as follows. Section 2 describes scheduling of sleeping periods. Section 3 presents a simple fluid-flow model for periodic on and off scheduling. Section 4 develops our full model. Section 5 contains numerical (model) and simulation

results that illustrate interesting behaviors and the ability to predict delay and sleep. Section 6 describes related work in use of coalescing for energy savings and modeling of interrupted service queues. Section 7 is a summary and outlines possible future work. Finally, the appendices contain key proofs.

## 2 Scheduling by Coalescing

Coalescing can be used to aggregate individual arriving jobs into bursts of jobs to reduce the overhead of sleep and wake transitions. Scheduling of sleeping periods – which directly determines the level of energy savings and performance tradeoffs – is basically a decision of when to start and end a coalescing period. The start and end of a coalescing period for a coalescing queue can be triggered by several conditions related to queue state and/or time duration:

1. Starting a coalescing period on the arrival of the first job to an empty coalescing queue and ending the coalescing period when a predetermined number of jobs have been queued and/or a time period since the arrival of the first job has expired.
2. Starting a coalescing period when the number of jobs in a coalescing queue drops below a predetermined threshold and ending the coalescing period when a predetermined number of jobs have been queued and/or time period since the arrival of the first job has expired.
3. Starting and stopping the coalescing period based on predetermined time periods.

Each of the above approaches has been used in one or more existing communication technologies, which will be extensively reviewed in Section 6 of this paper. Namely, methods studied in [5,23] and deep sleep in EPON ONUs [3] use the first approach, the On/Off-1 algorithm in [12] uses the second approach, and Synchronized coalescing [20] and cyclic sleep in EPON ONUs [3] use the third approach. The first and the second approaches can only be used if coalescer queue state is known. But if the coalescer queue state cannot be known, then only purely time-based approaches – such as the third approach above – can be implemented. Approaches based solely on timers are especially useful in systems where the service center is remotely turned on and off with no information about the queue length (and other statistics of the queue) available at the time of making on and off scheduling decisions. In this paper, we specifically consider a time-based approach where the coalescing period is of a predetermined duration $T_{off}$, and starts periodically at times $0, T_{off} + T_{on}, 2(T_{off} + T_{on})$, and so on. At the start of an off period (duration $T_{off}$) the server stops serving immediately if idle, or when any in-service job has completed if not idle. During an on period (duration $T_{on}$), jobs are not coalesced but will
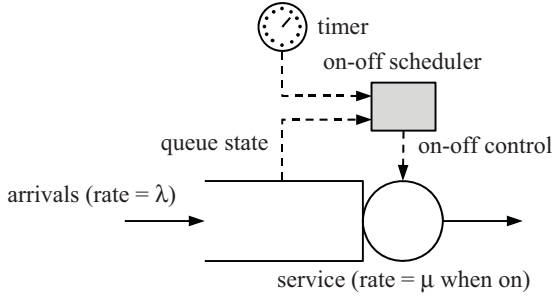
Fig. 2: Coalescing queue with scheduler



Fig. 3: Fluid flow model of periodic on and off coalescing

queue if the interarrival time between jobs is less than the service time of the coalescer queue. During the off period, the service rate is zero, and during the on period, the service rate is $\mu$ with mean service time $x = 1/\mu$. Jobs arrive at a rate $\lambda$ with mean interarrival time $t = 1/\lambda$.

Figure 2 shows a coalescing queue with on-off scheduling. The on-off scheduling in our case is based on timer state. A stated above, it is assumed that a job in service when an on period expires and the next off period starts completes its service. It is thus possible that an off period contains a maximum time $x$ of service time. Thus, the system could sleep for a minimum duration of $T_{off} - t_{wake} - t_{sleep} - x$ in all off period cases. We define a duty cycle as,

$$D = \frac{T_{on}}{T_{off} + T_{on}}. \tag{1}$$

The offered load to the server is then

$$\rho = \frac{1}{D} \cdot \frac{\lambda}{\mu}, \tag{2}$$

where $\rho < 1$ is required for stability. If $t_{wake}$, $t_{sleep}$, and $t$ are small compared to $T_{off}$, then $D$ models (from an engineering perspective) the percentage of time the system is on. The direct energy savings can be calculated from this known on time. The setting of $D$ is an engineering decision based on the desired energy savings with trade-off in performance. As stated previously, the setting of $D$ can only be done with a full understanding of the trade-off between increased energy savings and decreased performance.

Now we are ready to introduce the two averages that we will study in this paper. We define the *interval* mean queue length and *long-term* mean queue length as follows given periodic on and off periods of fixed duration $T_{on}$ and $T_{off}$, respectively. Let $H(s)$ be the queue length or number of jobs in the queue at time $s$.

**Definition 1.***The interval mean queue length is the mean number of jobs in the coalescing queue for a given interval defined from the start of an off period to the end of the subsequent on period (or duration $T_{off} + T_{on}$) defined as,*
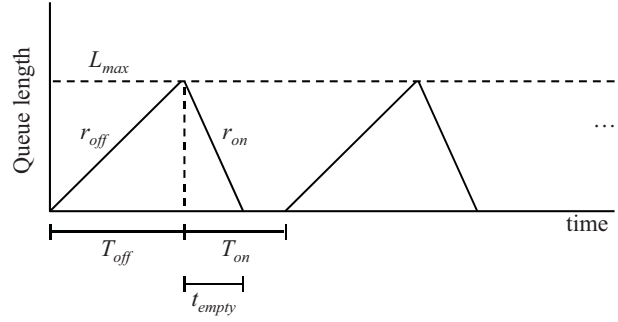
$$L = \frac{1}{T_{off} + T_{on}} \int_{0}^{T_{off}+T_{on}} H(s)\,ds. \tag{3}$$

**Definition 2.***The long-term mean queue length is the mean number of jobs in the coalescing queue over the long term for a long sequence of off and on periods.*

## 3 A Fluid Flow Queue with Periodic On and Off Service Periods

Let us model our fluid flow queue with arrivals at a rate $\lambda$ and service at a rate $\mu$. On and off periods are periodic with duration $T_{on}$ and $T_{off}$, respectively. During on periods jobs are served. During off periods the server is forced to be idle (that is, it does not serve) and queueing occurs. Here, performance is measured as mean delay, $W$. Figure 3 shows a fluid flow model of periodic on and off coalescing from which the mean delay can be determined. In this model, $r_{off} = \lambda$ is the rate of increase in queue length during the off period, and $r_{on} = \mu - \lambda$ is the rate of decrease in queue length during the on period.

In the fluid-flow model described above, the maximum queue length is $L_{max} = r_{off}T_{off}$. The time to empty the queue during an on period is $t_{empty} = L_{max}/r_{on}$. Also, $H(s)$ is the fluid accumulation (queue length) of the queue at time $s$ in the fluid-flow model. We seek to compute the interval mean queue length for a given $T_{off} + T_{on}$ interval as the area under $H(s)$ divided by the total time of an interval,

$$L = \frac{L_{max}(T_{off} + t_{empty})}{2(T_{off} + T_{on})} = \frac{\mu\lambda T_{off}^2}{2(\mu - \lambda)(T_{off} + T_{on})}. \tag{4}$$

With identical repeating on and off periods, the long-term mean queue length (as seen by a random outside observer) is given by (4). The criterion for stability is $t_{empty} \le T_{on}$. From Little's Law we can trivially determine

the mean delay,

$$W = \frac{L}{\lambda} = \frac{\mu T_{off}^2}{2(\mu - \lambda)(T_{off} + T_{on})}. \qquad (5)$$

The fluid flow model leads to our model of scheduling with periodic on and off service periods.

## 4 Scheduling Periodic On and Off Service Periods

Let us assume a queue where the first arrival is at $g$ (this is from time 0, the system always starts at time 0 in an off period) and the other parameters are as defined in Section 3. The natural assumption is that $0 \le g < t < T_{off}$. We will assume that if a job arrives exactly at time $T_{off} + T_{on}$, it will not be served until the next service interval. Figure 4 shows the behavior of the queue length for periodic on and off periods. The figure is drawn to scale and schematically shows the queue length behavior for two off-on cycles of an example where $T_{off} = 5$, $T_{on} = 2$, $g = 0$, $\lambda = 1$, and $\mu = 4$. Note the repetitive pattern, which will be used later in our analysis. Let $H(s)$ be the number of jobs in the queue at time $s$. We seek to find a good approximation or, if possible, a precise solution for the mean queue length over a long time period. Our construction is as follows. First, we produce an exact closed-form solution for the mean queue length in the most general case (arbitrary $T_{on}$, $T_{off}$, $t$, $x$, and $g$) over an interval that includes one off and one on period. Then we will show that the mean queue length on such an interval is a non-increasing function of $g$ and derive the upper and lower bounds for the mean queue length on these intervals that are independent of $g$ (the first arrival time within the periodic interval). This will give us a close estimate for the mean queue length. Next, we will give a precise closed-form expression for the mean queue length under the assumption that the ratio of $\frac{T_{on}+T_{off}}{t}$ is a rational number. Finally, in the most general case (arbitrary $T_{on}$, $T_{off}$, $t$, $x$, and $g$; and $\frac{T_{on}+T_{off}}{t}$ is not necessarily a rational number), we derive another good estimate of the mean queue length (in addition to the estimate described earlier) by making the assumption that the mean queue length in a given periodic interval is a linear function of the first arrival time $g$. Therefore, the mean queue length can be given precisely in closed-form or be very closely approximated by the two methods that we present here.

With all other parameters fixed, for $g$ in the interval $[0,t)$, denote as $L(g)$ the interval mean queue length within the interval $[0, T_{off} + T_{on})$ given that the first arrival beginning at time 0 occurs at time $g$. That is, $L(g) = \frac{1}{T_{off}+T_{on}} \int_0^{T_{off}+T_{on}} H(s)\,ds$, computed under the assumption that the first arrival time within the interval $[0, T_{off} + T_{on})$ is $g$. In other words, $L(g)$ is $L$ from Definition 1, given that the first arrival time is $g$.
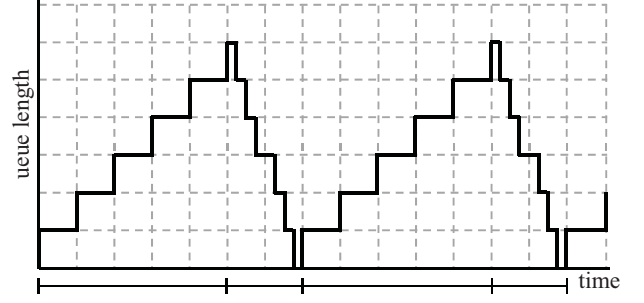


**Fig. 4:** Graphical model of periodic on and off coalescing

We denote as $N$ the total number of jobs arriving during the $[0, T_{off} + T_{on})$ time interval,

$$N = \left\lceil \frac{T_{off} + T_{on} - g}{t} \right\rceil. \qquad (6)$$

We also define,

$$w = max(0, g - t + x), \qquad (7)$$

and,

$$v = max\big(0, (N-1)t + g + x - (T_{off} + T_{on})\big). \qquad (8)$$

Here $w$ will be used to count the arrival overlap from the previous interval and $v$ will be used to exclude the service time of the $N$th job that occurred after $T_{off} + T_{on}$, if its service ends after $T_{off} + T_{on}$. Let us also define $k_0 \ge 1$ as the number of the first job that arrives at time 0 or later, after serving which the queue becomes empty (even if the next job arrives precisely at the same moment when the service to the $k_0$th job ends). The service of the $k_0$th job ends at time $T_{off} + k_0 x$, since the server has no idle periods between the times $T_{off}$ and $T_{off} + k_0 x$. The $(k_0 + 1)$th job arrives at time $k_0 t + g$. Thus $k_0$ is the smallest integer such that $k_0 t + g \ge T_{off} + k_0 x$. This means that

$$k_0 = \left\lceil \frac{T_{off} - g}{t - x} \right\rceil. \qquad (9)$$

Here we are presenting our main theoretical results. Theorem 1 computes the interval mean defined earlier in this paper. This computation is performed under the most general assumptions on the system parameters and with an arbitrary time of the first arrival during this interval. The next two results play an auxiliary role in our goal to estimate the long-term mean. Theorem 2 states an intuitively obvious property that the interval mean in increasing with the increase in the first arrival time within one interval. This property and the demonstrated continuity of the interval mean as a function of the first arrival time allow us to derive, in Theorem 3, the precise

upper and lower bounds for the interval mean. Finally, Theorem 4 provides the precise expression for the long-term mean when the fraction $(T_{off} + T_{on})/t$ is a rational number. In practical applications, it is always possible to assume that this is the case. The smaller the denominator in this rational fraction the fewer terms will have to be added to compute the long-term mean. The proofs for these theorems can be found in the appendices of this paper.

**Theorem 1.** *The interval mean queue length is*

$$L(g) = \frac{1}{T_{off} + T_{on}} \left( w + k_0 T_{off} - \frac{(k_0 + 1)k_0}{2}(t - x) \right.$$
$$\left. + k_0 t - k_0 g + (N - k_0)x - v \right). \tag{10}$$

**Proof.** The proof is in Appendix A.

While the interval mean queue length $L(g)$ is defined for $g$ in the interval $[0,t)$ only, we can formally define function $A(g)$ as the expression in the right hand side of (10) with $N$ and $k_0$ defined by (6) and (9) respectively. The introduction of $A(g)$ defined on the entire interval $[0,t]$ is necessary in order to state and prove some of the results that follow.

**Theorem 2.** $A(g)$ *is a continuous non-increasing function of $g$ when $g \in [0,t]$.*

**Proof.** The proof is in Appendix B.

**Theorem 3.** *The interval mean value of the queue, $L(g)$, satisfies $A(t) \le L(g) \le L(0)$ for $g \in [0,t]$, and the bounds cannot be improved.*

**Proof.** The proof is in Appendix C.

Finally, we solve for the interval mean queue length, $L(g)$, for the general case. Let $A = T_{off} + T_{on}$. If $A/t$ is rational, then $A/t = m/n$ for some mutually prime integers $m$ and $n$. We also let $\{x\}$ denote the fractional part of $x$; that is, for instance, $\{2.3\} = 0.3$, $\{5\} = 0$, and so on. We define the sequence $g_1, g_2, \dots$ by setting $g_1$ to

$$g_1 = \frac{t}{n} \left\{ \frac{gn}{t} \right\}, \tag{11}$$

and

$$g_i = g_1 + \frac{(i-1)t}{n}, \quad i = 2, 3, \cdots. \tag{12}$$

**Theorem 4.** *The long-term mean queue length for rational $A/t$ is given by*

$$L = \frac{1}{n} \sum_{i=1}^{n} L(g_i). \tag{13}$$

**Proof.** The proof is in Appendix D.

We note that if $A/t$ is rational, an exact expression for the long-term mean, $L$, is obtainable. If $A/t$ is irrational, an increasingly precise approximate result is achieved by increasing $n$.

**Corollary 1.** *Assume again that $A/t = m/n$ with mutually prime $m$ and $n$ is rational and $L(g)$ is a linear function of $g$ (this is a reasonable approximation of the behavior of $L(g)$). Then the mean of $L(g)$ can be expressed as*

$$L\left(g_1 + \frac{t(n-1)}{2n}\right). \tag{14}$$

**Proof.** If $L(g)$ can be expressed as $ag + b$ for some $a$ and $b$, then we write

$$\frac{1}{n}\sum_{i=1}^{n} L\left(g_1 + \frac{i-1}{n}t\right) = \frac{1}{n}\sum_{i=1}^{n}\left(ag_1 + \frac{a(i-1)}{n}t + b\right)$$
$$= ag_1 + \frac{1}{n^2}at\sum_{i=1}^{n}(i-1) + b$$
$$= ag_1 + \frac{atn(n-1)}{2n^2} + b$$
$$= ag_1 + \frac{at(n-1)}{2n} + b$$
$$= a\left(g_1 + \frac{t(n-1)}{2n}\right) + b$$
$$= L\left(g_1 + \frac{t(n-1)}{2n}\right). \tag{15}$$

**Corollary 2.** *Again assume that $L(g)$ is a linear function of $g$. Also assume that $A/t$ is an irrational number, so there are no $m$ and $n$. Then the mean of $L(g)$ is equal to*

$$L\left(\frac{t}{2}\right). \tag{16}$$

**Proof.** There are two ways to see that the statement of this corollary holds true. First, if $A$ is not a rational function of $t$, then the shift in $g$ between the off-on intervals is an irrational number and over long time these irrational numbers densely and uniformly cover the entire interval $[0,t)$. Hence, the linear function $L$ will average to its value in the center of this interval, that is, at $t/2$. Another way to prove the corollary is to see that an irrational $A/t$ can be closely approximated by a rational $m/n$, with the approximations (14) getting better and better as $n$ grows. As $n$ gets larger, $g_1$ tends to 0 and therefore, the value inside the large parenthesis in (14) tends to $t/2$. With the linear function $L$ being continuous, the value in (14) tends to $L\left(\frac{t}{2}\right)$.

## 5 Numerical and Simulation Results

In this section, we numerically demonstrate that (13) computes the long-term mean queue length, $L$, as a function of $g$. For the case of irrational $A/t$ we show convergence to simulation results as $n$ is arbitrarily increased in the computation of (13). We also compare results from (13) to that of the fluid flow model (4) for the case of rational $A/t$. Using a simulation model, we also show that the long-term mean queue length resulting from (13) approximates that of a similar queueing system with stochastic arrivals.
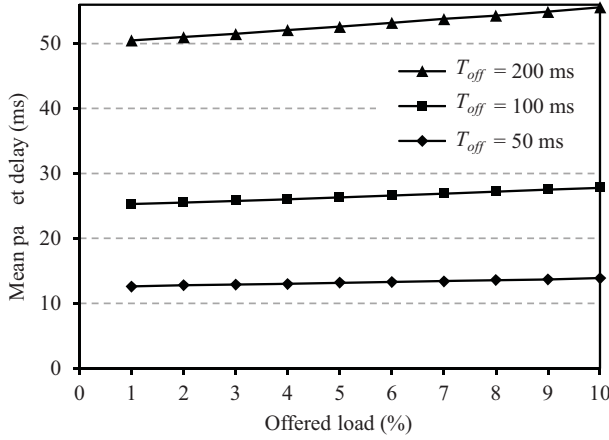
**Fig. 5:** Packet delay for 10 Gb/s EPON with cyclic sleep ($D = 0.5$)

## 5.1 Numerical results for long-term mean queue length

Table 1 shows a comparison of long-term mean queue length $L$ computed from (13) to a simulation model result for the following parameters: $n$ = 10, 100, and 1000, $T_{off}$ = 20 s, $T_{on}$ = 10 s, $g$ = 0, $D$ = 1/3, $\mu$ = $\sqrt{2}$ (irrational), and $\lambda$ computed from (2) such that $\rho$ = 0.1, 0.5, and 0.9. The simulation model was constructed using CSIM [1] as a queueing model of a single-server queue with periodic on and off periods. The results in Table 1 show that the computation from (13) converges to the simulated result as $n$ increases. This was also found to be true for cases where $g > 0$ (results not shown here). Even for small $n$ ($n$ = 10), the result from (13) was very close to the simulated actual. For cases where $A/t$ is rational, (13) and the simulation model results were found to match exactly, as should be expected.

Figure 6a shows a comparison of long-term mean queue length, $L$, computed from (13) to the fluid flow model of (4) for the following parameters: $D$ = 0.1, $\mu$ = 1.0, $T_{off}$ = 10, 15, ..., 100 (that is, 10 to 100 times greater than service time, $x$), $T_{on}$ computed from (1), and $\lambda$ computed from (2) such that $\rho$ = 0.1, 0.5, and 0.9. Relative error is shown as a percentage. Figure 6b shows the same comparison for $D$ = 0.5. It can be seen that as $T_{off}$ compared to $x$ increases, the relative error decreases

and the fluid flow model becomes a closer approximation. The region of small duty cycle, low offered load, and small $T_{off}$ compared to $x$ may be the region of most interest for many practical applications of coalescing. In this region, the fluid flow model is a very poor approximation.

We used simulation to compare our model to that of a similar system with exponentially distributed interarrival times (Poisson arrivals). Figure 7 shows a comparison of long-term mean queue length, $L$, computed from (13) to that obtained from the simulation with Poisson arrivals. The parameters are the same as what was used in generating Figure 6 with $\lambda$ being the mean arrival rate of the packets. Relative error is shown as a percentage for (a)$D$ = 0.1, and (b)$D$ = 0.5. As can be seen in the figure, the error is less than 10% for all values of $T_{off}$ including the small values that is the primary region of interest. The overall variability of the time between the arrival of subsequent jobs decreases in general when the samples are taken in larger time spans which is likely to be the reason for the decrease in relative error when $T_{off}$ increases. The relative error tends to zero as $T_{off}$ increases making the approximation of our model approach reality as $T_{off}$ increases. An interesting phenomenon that can happen in a system with stochastic arrivals is that the coalescing queue does not always empty completely by the time when the next off period starts. The remaining packets can be served either in 1) the next on period following the next off period, or 2) the same on period by postponing the start of the next off period until the queue empties. We chose the second policy for serving any remaining packets in our simulation and instrumented the model to measure the reduction in total off time that this policy induces. In our experiments, this reduction in time did not exceed 5% of the total off time (implying that the energy savings would be reduced by not more than 5% as well). The reduction increased, not surprisingly, as the load got higher where the maximum reduction occurred when $\rho = 0.9$.

## 5.2 Numerical results for a 10 Gb/s Ethernet Passive Optical Network

We evaluated the packet delay for a 10 Gb/s Ethernet Passive Optical Networks (EPON) with cyclic sleep (as described in [6,26]) for low utilization levels as would be typically expected in such a system. We used a 50% duty cycle, $D$ = 0.5, where the system sleeps for a time $T_{off}$, which includes both wake and sleep transition overheads. We conservatively assumed that the power draw during transitions was the same as during on, or wake, periods. For $D$ = 0.50, $T_{on}$ = $T_{off}$. We assumed that the service time corresponded to the transmission of a maximum length 1500-byte packet, which is 1.2 $\mu$s. The long-term mean queue length was computed using (13) for $T_{off}$ = 50 ms, 100 ms, and 200 ms, and the offered load ranged

**Table 1:** Effect of increasing $n$ in (13) for irrational $A/t$

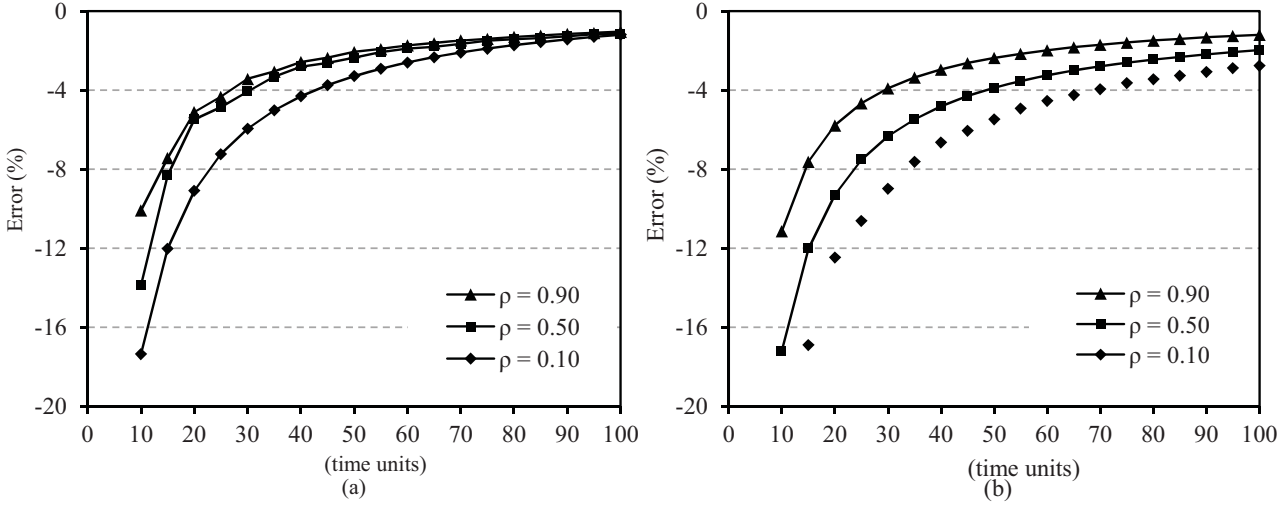| | $n$ | | | |
|---|---|---|---|---|
| $\rho$ | 10 | 100 | 1000 | Simulation |
| 0.10 | 0.382473 | 0.350958 | 0.347937 | 0.348 |
| 0.50 | 2.028797 | 1.993046 | 1.989514 | 1.989 |
| 0.90 | 4.247608 | 4.205543 | 4.201344 | 4.201 |

**Fig. 6:** Relative error for (13) versus (4) for (a)$D = 0.10$, and (b)$D = 0.50$
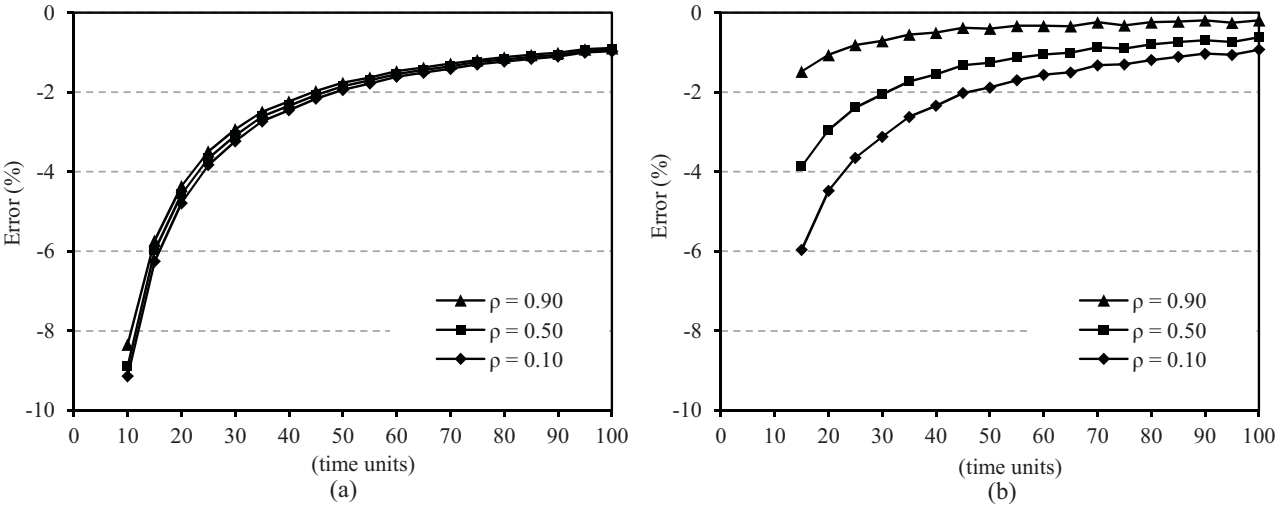


**Fig. 7:** Relative error for (13) versus that of a similar M/D/1 queue for (a)$D = 0.10$, and (b)$D = 0.50$

from 1% to 10%. Figure 5 shows the results. It can be seen, not surprisingly, that the mean delay increases as $T_{off}$ increases and as offered load increases. For a 5% offered load, the mean packet delay is about 25% of $T_{off}$ in the three cases shown. This type of evaluation can be used for determining an acceptable delay versus energy use trade-off for systems that use coalescing.

# 6 Related Work

In this section, we first review applications of coalescing and then review previous and related work in modeling of queues with service interruptions.

## 6.1 Use of coalescing to reduce energy consumption

Coalescing of requests has long been used to eliminate receive livelock (a situation that can occur under heavy load in computer systems in which the processor spends all its time processing interrupts and no time processing actual jobs) [19]. Coalescing has been used in disk drives for conserving energy by reducing disk spinning operations [24]. Coalescing is used in several wireless protocols to reduce energy use. In Power Saving Mechanism (PSM) for the Distributed Coordination Function (DCF) in IEEE 802.11 wireless networks (Wi-Fi

networks), packets destined for a wireless station are coalesced in the preceding station (for example, in the access point) during predefined beacon intervals [2]. When a periodic beacon interval begins, a station listens to announcement packets from any station which has a packet to send. Upon receiving an announcement packet, a station remains powered on during the Data Transmission (DT) window in order to receive the announced packets. If the station does not receive any announcement packets, it can skip the DT interval and power down for the rest of the beacon time.

A recently developed scheme called Catnap [7] reduces the energy consumption of a mobile device by combining small gaps between packet transmissions into larger intervals during which time the device can be put to sleep and save energy. Catnap introduces a proxy on the wireless router connecting the mobile device and the Internet that performs this coalescing function for packets inbound to a wireless device. Catnap exploits the fact that the wired link between the Internet and an access point is often higher speed than the wireless link between an access point and wireless device. The speed mismatch causes idle periods between packets. These idle periods are extended with coalescing in Catnap proxy in the access point.

Sleep schemes have also been adopted for optical networks. Deep sleep and Cyclic sleep [3] in Ethernet Passive Optical Networks (EPONs) allow the Optical Network Units (ONUs) to turn off their hardware components in order to save energy. In deep sleep mode, the ONU's transmitter and receiver are powered off when traffic is neither being received nor transmitted by the ONU. In cyclic sleep mode, the ONU periodically cycles between active and sleep periods where its transceiver is powered on and off, respectively.

Recent work has focused on how packet coalescing policies for EEE can improve the energy efficiency of Ethernet ([5, 20, 23]). At low link utilizations, EEE can be very inefficient when individual packets trigger wake and sleep transitions that exceed the transmission time of the packet. In order to decrease transition time (and energy use) overhead, packets can be coalesced into bursts and then be sent as one burst of back-to-back packets. In the coalescing scheme studied in [5], when either the coalescing timer expires or the number of buffered packets reaches a defined maximum, the buffered packets were all transmitted in a single burst. This work showed that there is a trade-off between energy savings and increased packet delay from coalescing. An analytical model for EEE with packet coalescing with very general assumptions is recently developed in [14] based on the GI/G/1 queueing model with vacations. Synchronized Coalescing for EEE [20], is a timer-based packet coalescing policy for EEE which reduces the energy consumption of LAN switches. In synchronized coalescing, a LAN switch stops incoming traffic to all connected ports by periodically sending a PAUSE MAC frame on all its ports to sending hosts and/or edge routers.

When this occurs, all connected ports can enter LPI mode at the same time and internal switch components including the switching fabric can power down for this duration. Packets generated at the connected hosts or arriving at the edge routers during this pause duration are coalesced in the interface buffers until the pause interval is over.

As long as utilization of communication and computing systems remains low (and it is argued in [22] that network utilization will always be low) coalescing can be a useful means of enabling energy savings in such systems. This paper addresses the energy savings and delay trade-offs for coalescing in the case of deterministic on and off periods. These results provide a strong foundation upon which performance models of coalescing schemes can be built and a deeper understanding of this means of scheduling of jobs be gained.

## 6.2 Queues with service interruptions

Queues with service interruptions, modeled as service pre-emption, vacations, and breakdowns have been studied since the 1950s. This previous work gives us insights towards scheduling by coalescing. Service interruption can result from *unscheduled* breakdown of a server (for instance, when the arrival of a high priority customer temporarily stops service to other customers), or a *scheduled* service stop (for instance, between shifts in a product line). In both cases, typically, the duration of the interruption is randomly distributed. In our scheme, however, the start time and duration of off periods is known.

The first work in queues with service interruptions dates back to 1958 where priority classes for jobs with a pre-emptive resume service discipline were studied in [29]. From the perspective of a lower-priority job, service pre-emption appears as if the server breaks down and is repaired. Poisson arrivals and negative-exponential repair times were assumed in this work and the expected time spent in the system and the generating function for the delay distribution were derived. Preemptive-resume and non-resume disciplines were further studied in [11, 15] where an exponentially distributed time to the next breakdown was assumed. Special cases of the same system, such as when the server breaks down only when a job wishes to receive improved service from the server, or when the server never breaks down while serving a job were studied in [4].

Server interruption is closely related to server vacations and breakdowns. A server takes a vacation after completing the service to all queued jobs and the queue is empty. The vacation model has a fundamental difference with server breakdowns where the service can be interrupted while there are jobs in the queue or in service. Queuing systems with vacations were reviewed in [8, 27]. None of these previous works explicitly considered a

server that independently of job arrivals cycles between serving (on) and not serving (off) jobs.

More recent work has addressed servers that independently vary between on and off modes. In [10] the M/G/1 decomposition property is used to model any M/G/1 system with vacations. The mean waiting time in the queue, the probability of delay, and the steady-state distribution of the number of jobs in the system for a FCFS queue with Poisson arrivals and general service process for a variety of distributions for the on and off periods (where the distribution of either the on or off period length is not periodic) were approximated in [9]. A queuing system with server interruptions in which the interruptions were semi-Markovian was studied in [17]. The case where the on and off periods were governed by an alternating renewal process was considered in [25]. A queuing system with Markovian Arrival Process (MAP) and a very general random interruption distribution was solved (including numerical results) in [28]. The probability distribution of jobs in the system for the M(t)/M(t)/1 queue is solved in [16] using numerical methods. Additional recent work has addressed the characteristics of queues with fluctuation of loads (where server breakdowns are a special case where server service time is set to infinity) to determine fundamental results [13]. This work, as all previous work, also assumes stochastic on and/or off periods, which do not map to the case of timer-based coalescing as critically considered in our work.


# 7 Summary and Future Work

Our work is, we believe, the first to develop a model for timer-based on-off scheduling in the context of a coalescing queue. Our work has direct relevance to understanding increased delay in systems that use coalescing to extend idle periods in order to increase opportunities for the system to sleep. Theoretical results are important for deeper understanding of real systems. The model we have developed produces a result that is easily computable. Exact results are possible for many cases, approximate results are possible for all cases. Comparison to simulation shows that the approximations are very good. Numerical results show that coalescing converges to a fluid flow case as $T_{off}$ increases relative to $x$ and as $T_{on}$ increases relative to $t$. An interesting case is where the on period is of duration equal to the time to serve all packets in the queue and the next off period begins at the time of the first arrival immediately after the queue becomes empty. Future work should consider 1) modeling the M/G/1 queue with periodic on and off periods, 2) the analysis of on and off policies other than periodic (for example, as described in Section 2), and 3) studying the effects of the output process from one (or more) coalescing queue(s) on a downstream queue.

# Appendix A Proof of Theorem 1

To compute $L(g)$, we assume that the process begins before the interval where we are computing the interval mean queue length, and there may be an overlap from the previous interval. The last arrival before time 0 was at time $g - t$ and it took time $x$ to process this job. Hence, $H(s) = 1$ between time 0 and $w$, and $H(s) = 0$ between $w$ and $g$. Therefore,

$$\int_0^g H(w)\,ds = 1 \cdot w = w. \tag{17}$$

where $w$ is defined in (7).

We now evaluate $H(s)$ when $s \geq g$. Let us define two functions $F(s)$ and $G(s)$, where $F(s)$ is the number of jobs that have arrived by time $s$ and $G(s)$ is the number of jobs that have been completed by time $s$. When $s \geq g$,

$$F(s) = \left\lfloor \frac{s-g}{t} \right\rfloor + 1. \tag{18}$$

For $g \leq s \leq T_{off}$, $G(s) = 0$, and for $T_{off} < s \leq T_{off} + k_0 x$,

$$G(s) = \left\lfloor \frac{s - t_{off}}{x} \right\rfloor, \tag{19}$$

where $k_0$ is defined in (9). The above expression for $G(s)$ does not hold for $s > T_{off} + k_0 x$ since the server might have some idle periods after that time. We can see that $H(s) = F(s) - G(s)$ for all $s$. We further note that $H(s) = 0$ for $s \in (T_{off} + k_0 x, k_0 t + g)$, since this interval starts after the completion of service to job $k_0$ and ends before job number $(k_0 + 1)$ arrives. After that, when the job number $j$ arrives $(j = k_0 + 1, \ldots, N)$, the service queue length is 1 between the times $(j-1)t + g$ and $(j-1)t + g + x$, and 0 between the times $(j-1)t + g + x$ and $jt + g$. For the $N$th job, it is not known if its service gets completed before $T_{off} + T_{on}$. If it does not, we cannot include the part of service that occurs after $T_{off} + T_{on}$ in the calculation of the interval mean queue length in the $[0, T_{off} + T_{on})$ interval, so we will subtract $1 \cdot v = v$ from the total, where $v$ is defined in (8).

To compute the average size of the queue we need to integrate $H(s)$ over the interval from 0 to $T_{off} + T_{on}$ and then divide the result over $T_{off} + T_{on}$. We have,

$$\int_0^{T_{off}+T_{on}} H(s)\,ds = w + \int_g^{T_{off}+k_0x} H(s)\,ds +$$
$$\int_{T_{off}+k_0x}^{k_0t+g} H(s)\,ds + \int_{k_0t+g}^{T_{off}+T_{on}} H(s)\,ds. \tag{20}$$

The second integral in the right hand side is 0, since, as we saw earlier, there are no jobs in the queue at this time. The third integral is equal to $(N - k_0)x - v$, since the value of

$H(s)$ is 1 during the $(N - k_0)$ intervals of length $x$ (each interval corresponding to serving a job beginning with job number $k_0 + 1$), and 0 at other times, except that the service of the last job may end outside this interval and thus the adjustment by $v$. Now,

$$\int_g^{T_{off}+k_0x} H(s)\,ds = \int_g^{T_{off}+k_0x} F(s)\,ds - \int_g^{T_{off}+k_0x} G(s)\,ds. \quad (21)$$

We can see that $F(s) = 1$ between $g$ and $g + t$, it is equal to 2 between $g + t$ and $g + 2t, \ldots$, and equal to $k_0 - 1$ between $g + (k_0 - 2)t$ and $g + (k_0 - 1)t$. It is equal to $k_0$ between the times $g + (k_0 - 1)t$ and $T_{off} + k_0x$. Note that by our definition of $k_0$, the next job, $(k_0 + 1)$, arrives at or after the time $T_{off} + k_0x$. Therefore, the first integral in the right hand side of (21) can be calculated as

$$\int_g^{T_{off}+k_0x} F(s)\,ds = \frac{(k_0 - 1)k_0}{2}t$$

$$+ k_0\big(T_{off} + k_0x - (k_0 - 1)t - g\big). \quad (22)$$

$G(s)$ is equal to 0 between the times 0 and $T_{off} + x$, equal to 1 between the times $T_{off} + x$ and $T_{off} + 2x, \ldots$, and equal to $k_0 - 1$ between the times $T_{off} + (k_0 - 1)x$ and $T_{off} + k_0x$. Therefore,

$$\int_g^{T_{off}+k_0x} G(s)\,ds = \frac{(k_0 - 1)k_0}{2}x. \quad (23)$$

Hence,

$$\int_g^{T_{off}+k_0x} H(s)\,ds = \int_g^{T_{off}+k_0x} F(s)\,ds - \int_g^{T_{off}+k_0x} G(s)\,ds$$

$$= \frac{(k_0 - 1)k_0}{2}(t - x)$$

$$+ k_0(T_{off} + k_0x - (k_0 - 1)t - g)$$

$$= k_0 T_{off} - \frac{(k_0 + 1)k_0}{2}(t - x) + k_0t - k_0g. \quad (24)$$

Combining (20) and (24) we calculate the interval mean queue length as,

$$L(g) = \frac{1}{T_{off} + T_{on}}\left(w + k_0 T_{off} - \frac{(k_0 + 1)k_0}{2}(t - x)\right.$$

$$\left. + k_0t - k_0g + (N - k_0)x - v\right). \quad (25)$$

This is (10). □

## Appendix B Proof of Theorem 2

We will demonstrate that for every $g \in [0, T_{on} + T_{off}]$, the function $A(g)$ is non-increasing and continuous at $g$. From the definition of $A(t)$, it follows, that $A(g)$ is a continuous function of $g$, $w$, $v$, $k_0$ and $N$. Note that the other parameters, $T_{off}$, $T_{on}$, $t$ and $x$ are independent of $g$. From (6) and (9), it follows that $N$ and $k_0$ are both right-continuous functions of $g$. From Lemma 1 stated below in this appendix it follows that $A(g)$ is a right-continuous function of $g$. Now, we need to prove that $A(g)$ is also left-continuous and non-increasing. We divide our proof into four cases based upon the continuity of $N$ and $k_0$ at $g$.

**Case 1.** Suppose that neither $\frac{T_{off}-g}{t-x}$ nor $\frac{T_{off}+T_{on}-g}{t}$ is an integer. From this it follows that $g$ is such that there is no discontinuity of either $k_0$ or $N$ (as functions of $g$) at this value of $g$. Then $A(g)$ is continuous at this $g$ as $k_0$ and $N$ are the same when evaluated at points near $g$. To prove the non-increasing property of $A(g)$ in this case, it would suffice to show that $w - k_0g - v$ is a non-increasing function of $g$. Note that all other terms in the parentheses in (10) remain unchanged near $g$. Since $-v$ can only decrease or remain the same as $g$ increases, it is sufficient to show that $w - k_0g$ does not increase. Indeed, when $g$ increases by some $\delta > 0$, $w$ increases by at most $\delta$, while $-k_0g$ decreases by $k_0\delta \geq \delta$. Thus, $A(g)$ is non-increasing at this $g$.

**Case 2.** Let $g$ be such that $\frac{T_{off}+T_{on}-g}{t}$ is an integer while $\frac{T_{off}-g}{t-x}$ is not. To prove the left-continuity of $A(g)$ at this $g$, we decrease the value of $g$ to $g - \delta$. This causes the value of $N$, when evaluated at $g - \delta$ to change to $N + 1$. Since $N$ is only present in the $(N - k_0)x$ term and in $v$ inside the parentheses in (10), and the value of $k_0$ remains the same if $\delta$ is sufficiently small, we should only look at the effect of the change on these two terms, as other terms are continuous in $g$. Clearly, the $(N - k_0)x$ term increases by $x$ when $N$ increases to $N + 1$. Using $Nt = T_{on} + T_{off} - g$ we obtain the value of $v$ at $g$ as

$$max\big(0, Nt - t + g + x - (T_{on} + T_{off})\big) = max\big(0, -t + x\big) = 0. \quad (26)$$

So, $v(g) = 0$. The value of $v$ at $g - \delta$ is ($N$ becomes $N + 1$)

$$max\big(0, Nt + g - \delta + x - (T_{on} + T_{off})\big) = max\big(0, x - \delta\big)$$

$$= x - \delta, \quad (27)$$

therefore, $v(g - \delta) = x - \delta$. Then, replacing $w(g)$, $v(g)$, $w(g - \delta)$ and $v(g - \delta)$ with $w$ and $v$ evaluated at $g$ and

$g - \delta$ correspondingly, we obtain

$$(T_{off} + T_{on})(A(g-\delta) - A(g)) =$$
$$\left( w(g-\delta) + k_0 T_{off} - \frac{k_0(k_0+1)}{2}(t-x) \right.$$
$$+ k_0 t - k_0(g-\delta) + (N+1-k_0)x - v(g-\delta)$$
$$\left. - \left( w(g) + k_0 T_{off} - \frac{k_0(k_0+1)}{2}(t-x) \right. \right.$$
$$+ k_0 t - k_0 g + (N - k_0)x - v(g)$$
$$= \left( w(g-\delta) - w(g) \right) + k_0 \delta + x - \left( v(g-\delta) - v(g) \right)$$
$$= \left( w(g-\delta) - w(g) \right) + k_0 \delta + x - (x - \delta)$$
$$= \left( w(g-\delta) - w(g) \right) + (k_0 + 1)\delta. \qquad (28)$$

This shows, using the fact that $-\delta \leq w(g-\delta) - w(g) \leq 0$, that at this value of $g$ the function $A(g)$ is left-continuous (and therefore, continuous) and non-increasing.

    **Case 3.** Suppose that $\frac{T_{off}-g}{t-x}$ is an integer while $\frac{T_{off}+T_{on}-g}{t}$ is not. We will show that $A(g)$ is continuous and non-increasing at this value of $g$. Noting that $k_0(t-x) - T_{off} + g = 0$ in this case, we have

$$(T_{off} + T_{on})(A(g-\delta) - A(g)) =$$
$$\left( w(g-\delta) + (k_0+1)T_{off} - \frac{(k_0+1)(k_0+2)}{2}(t-x) \right.$$
$$+ (k_0+1)t - (k_0+1)(g-\delta) + (N-k_0-1)x$$
$$- v(g-\delta) \quad - \left( w(g) + k_0 T_{off} - \frac{k_0(k_0+1)}{2}(t-x) \right.$$
$$+ k_0 t - k_0 g + (N - k_0)x - v(g)$$
$$= \left( w(g-\delta) - w(g) \right) + T_{off} - (k_0+1)(t-x)$$
$$+ t - g + (k_0+1)\delta - x - \left( v(g-\delta) - v(g) \right)$$
$$= \left( w(g-\delta) - w(g) \right) + (k_0+1)\delta - \left( v(g-\delta) - v(g) \right).$$
$$\qquad (29)$$

From (29), it immediately follows that $A(g)$ is left-continuous at this value of $g$ and hence continuous. Also, since $\left( w(g-\delta) - w(g) \right) \geq -\delta$ and $-\left( v(g-\delta) - v(g) \right) \geq 0$, the value in the right hand side of (29) is no smaller than $k_0 \delta > 0$. Therefore, $A(g)$ is non-increasing at this value of $g$.

    **Case 4.** Finally, suppose that both $\frac{T_{off}-g}{t-x}$ and $\frac{T_{off}+T_{on}-g}{t}$ are integers. Noting that in this case $k_0(t-x) - T_{off} + g = 0$, and that $v(g-\delta) = x - \delta$ and

$v(g) = 0$, we have

$$(T_{off} + T_{on})(A(g-\delta) - A(g)) =$$
$$\left( w(g-\delta) + (k_0+1)T_{off} - \frac{(k_0+1)(k_0+2)}{2}(t-x) \right.$$
$$+ (k_0+1)t - (k_0+1)(g-\delta) + (N+1-k_0-1)x$$
$$- v(g-\delta) \quad - \left( w(g) + k_0 T_{off} - \frac{k_0(k_0+1)}{2}(t-x) \right.$$
$$+ k_0 t - k_0 g + (N - k_0)x - v(g)$$
$$= \left( w(g-\delta) - w(g) \right) + T_{off} - (k_0+1)(t-x)$$
$$+ t - g + (k_0+1)\delta - \left( v(g-\delta) - v(g) \right)$$
$$= \left( w(g-\delta) - w(g) \right) + (k_0+2)\delta \geq (k_0+1)\delta.$$
$$\qquad (30)$$

This shows that $A(g)$ is continuous and non-increasing at this value of $g$ as well, and completes the proof of Theorem 2.     □

**Lemma 1.** *If $F$ is a continuous function and $f$ is a right-continuous function (both are functions of one real variable) then $F(f)$ is a right-continuous function.*

**Proof.** We need to show that for any $\varepsilon > 0$ there exists $\delta > 0$ such that for any $x$ where $F(f)$ is defined (and hence, where $f$ is defined), if $y$ satisfies the inequalities $0 < y - x < \delta$, the following holds: $|F(f(y)) - F(f(x))| < \varepsilon$. Since $F$ is continuous at $f(x)$, for this specific $\varepsilon$ there exists $\xi > 0$ such that if $|f(y) - f(x)| < \xi$ then $|F(f(y)) - F(f(x))| < \varepsilon$. Since $f$ is right-continuous at $x$, there exists $\delta > 0$ such that $|f(y) - f(x)| < \xi$ for any $y$ such that $0 < y - x < \delta$. This $\delta$ hence satisfies the conditions of the lemma for the chosen $\varepsilon > 0$.

## Appendix C  Proof of Theorem 3

Since, according to Theorem 2, $L(g)$ is non-increasing, it is bounded by $A(t)$ below and $L(0)$ above. The upper bound is reached at $g = 0$ and therefore cannot be improved.

    The variable $g$ is allowed to take all real values as it increases towards $t$. This means that $L(g)$ can be any value between $L(0)$ and $\lim_{g \uparrow t} L(g)$. This latter limit exists and is equal to $A(t)$. Due to the property of the limit, the lower bound in this theorem cannot be improved.     □

## Appendix D  Proof of Theorem 4

We seek the mean of $L(g)$. The mean is,

$$L = \lim_{T \to \infty} \frac{1}{T} \int_0^T L(g, s)\, ds, \qquad (31)$$

where $L(g,s)$ is the queue length at time $s$ of the process with the very first arrival time at $g$, for an arbitrary set of parameters. $L(g,s)$ is the same as $H(s)$ in Appendix A except $g$ is now not fixed.

In the usual notation, we are assuming that $A/t = m/n$ for some mutually prime integers $m$ and $n$ and we seek to find the precise value for

$$\frac{1}{T}\int_0^T L(g,s)\,ds \text{ as } T \to \infty, \tag{32}$$

where $L(g,s)$ is the queue length at time $s$ of the process with the very first arrival time at $g$, for an arbitrary set of parameters (here, as before, $A = T_{off} + T_{on}$). We will denote by $g^{(1)}$ the first $g$ upon the beginning of the process, by $g^{(2)}$ the $g$ at the beginning of the second interval, that is, the first arrival time in the interval that starts at $A$, by $g^{(3)}$ the $g$ at the beginning of the third off-on interval and so on.

**Lemma 2.** *For each $i > 1$, $g^{(i)} - g^{(1)} = \frac{M_i}{n}t$ for some integer $M_i$.*

**Proof.** When computing $g^{(i)}$ we need to add to $g^{(1)}$ some integer multiple of $t$ (one for each new arrival) and subtract $(i-1)A$, that is, subtract $A$ every time we go over the end of the previous off-on interval. Hence, for some integers $K$ and $M_i$,

$$g^{(i)} = g^{(1)} + Kt - (i-1)A$$
$$= g^{(1)} + \frac{Kn}{n}t - \frac{(i-1)m}{n}t$$
$$= g^{(1)} + \frac{M_i}{n}t, \tag{33}$$

where $M_i = Kn - (i-1)m$.

Let $m_i$, $i = 1,2,\ldots$ be the integer between 0 and $n-1$ such that

$$\frac{m_i}{n}t \le g^{(i)} < \frac{m_i+1}{n}t, \tag{34}$$

and $a_i \in [0,1)$ be such that

$$g^{(i)} = \frac{m_i + a_i}{n}t. \tag{35}$$

**Lemma 3.** *For each $i > 1$, the $a_i$'s are all equal.*

**Proof.** For each $i > 1$,

$$g^{(i)} - g^{(1)} = \frac{m_i - m_1}{n}t + \frac{a_i - a_1}{n}t. \tag{36}$$

Since $|a_i - a_1| < 1$, it follows from Lemma 2 that $a_i - a_1 = 0$.

Denote by $a$ the common value of $a_i$.

**Lemma 4.** *The following is true:*

$$a = \left\{\frac{gn}{t}\right\}. \tag{37}$$

*where $g$ is the same as $g^{(1)}$ and $\{\}$ denotes the fractional part.*

**Proof.**

$$\left\{\frac{gn}{t}\right\} = \left\{\frac{\frac{m_1+a}{n}tn}{t}\right\} = \{m_1 + a\} = a. \tag{38}$$

It is now clear that the only difference in the values of $g^{(i)}$ is the value of $m_i$. Since $m_i$ can take no more than $n$ different values $(0, 1, \ldots, (n-1))$, there are no more than $n$ possible different values of $g^{(i)}$.

**Lemma 5.** *For each pair of distinct $i$ and $j$ such that $|i - j| < n$, $g^{(i)} \ne g^{(j)}$. If $|i - j|$ is a multiple of $n$, then $g^{(i)} = g^{(j)}$.*

**Proof.** Suppose $|i - j| < n$ and $i > j$. Getting from $g^{(i)}$ to $g^{(j)}$ takes some number $M$ (integer, obviously) of new arrivals, that cause a total of $(i-j)$ overlaps over the end of an off-on interval. Hence,

$$g^{(i)} = g^{(j)} + Mt - (i-j)A. \tag{39}$$

Therefore,

$$g^{(i)} - g^{(j)} = Mt - \frac{(i-j)m}{n}t. \tag{40}$$

If $g^{(i)} = g^{(j)}$, then $\frac{(i-j)m}{n}$ must be an integer. However, $m$ and $n$ are mutually prime and $0 < (i - j) < n$. Contradiction. This proves the first part of the lemma.

Further, from (40) it is clear that if $i - j$ is a multiple of $n$, then $g^{(i)} - g^{(j)}$ is the multiple of $t$. However, each $g^{(i)}$ is non-negative and less than $t$. Therefore the difference in (40) should be 0.

We can now see that $g^{(1)}, g^{(2)}, \ldots, g^{(n)}$ take all possible values $\frac{a}{n}t, \frac{1+a}{n}t, \ldots, \frac{n-1+a}{n}t$, where $a$ is derived in (37). We denote as $g_1$ the smallest of these values. So,

$$g_1 = \frac{t}{n}\left\{\frac{gn}{t}\right\}. \tag{41}$$

We denote the other values, in ascending order as

$$g_2 = g_1 + \frac{t}{n}, \cdots, g_n = g_1 + \frac{(n-1)t}{n}. \tag{42}$$

Now, to prove Theorem 4 we break an interval $[0,T)$ for large $T$, that is a multiple of $An$, into $M$ intervals of length $An$. Then, we can write

$$\lim_{T\to\infty}\frac{1}{T}\int_0^T L(g,s)\,ds = \lim_{M\to\infty}\frac{1}{Mn}\sum_{f=1}^M\int_{An(f-1)}^{Anf}\frac{1}{A}L(g,s)\,ds. \tag{43}$$

Further,

$$\int_{An(f-1)}^{Anf}\frac{1}{A}L(g,s)\,ds = \sum_{i=1}^n\int_{An(f-1)+(i-1)A}^{An(f-1)+iA}\frac{1}{A}L(g,s)\,ds. \tag{44}$$

Each integral in the sum in the right hand side of (44) is the mean queue length in an off-on interval from $An(f-1) + (i-1)A$ to $An(f-1) + iA$. The $g$ in (44) is equal to one of the $g^{(i)}$'s computed above and hence equal to one of the $g_i$'s in (42).

Changing $i$ between 1 and $n$ with $f$ fixed, we will run, as shown in Lemma 5, through all possible values of $g^{(1)}, g^{(2)}, \ldots, g^{(n)}$ and hence through all possible values of $g_1, g_2, \ldots, g_n$. Each integral in the right-hand side of (44) becomes by Theorem 1 equal to $L(g_i)$ for some $i$, and the right hand side in (44) becomes $\sum_{i=1}^{n} L(g_i)$ and is independent of the $f$ and $M$ in (43). Thus the right hand side in (43) is equal to $\frac{1}{n} \sum_{i=1}^{n} L(g_i)$. $\qquad\square$

## Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

## References

[1] CSIM 20 – Development toolkit for simulation & modeling. Mesquite Software Inc., URL: http://www.mesquite.com/ 2013.

[2] The Editors of IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification 1999.

[3] ITU-T Recommendation G.Sup45, GPON power conservation 2009.

[4] Avi-Itzhak B, Naor P. Some queuing problems with the service station subject to breakdown. Operations Research 1963;11(3):303–320.

[5] Christensen K, Reviriego P, Nordman B, Bennett M, Mostowfi M, Maestro J. IEEE 802.3az: The road to Energy Efficient Ethernet. IEEE Communications Magazine 2010;48(11):50–56.

[6] Dhaini A, Ho PH, Shen G. Toward green next-generation passive optical networks. IEEE Communications Magazine 2011;49(11):94–101.

[7] Dogar FR, Steenkiste P, Papagiannaki K. Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys) 2010;107–122.

[8] Doshi BT Queueing systems with vacations – a survey. Queueing Systems 1986;1:29–66.

[9] Federgruen A, Green L. Queueing systems with service interruptions. Operations Research 1986;34(5):752–768.

[10] Fuhrmann SW, Cooper RB. Stochastic decompositions in the M/G/1 queue with generalized vacations. Operations Research 1985;33:1117–1129.

[11] Gaver DP. A waiting line with interrupted service, including priorities. Journal of the Royal Statistical Society Series B (Methodological) 1962;24(1):73–90.

[12] Gupta M, Singh S. Using low-power modes for energy conservation in Ethernet LANs. Proceedings of the 26th IEEE International Conference on Computer Communications 2007;2451–2455.

[13] Gupta V, Harchol-Balter M, Wolf AS, Yechiali U. Fundamental characteristics of queues with fluctuating load. Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems 2006;203–215.

[14] Herreria-Alonso S, Rodriguez-Perez M, Fernandez-Veiga M, Lopez-Garcia C. A GI/G/1 model for 10Gb/s Energy Efficient Ethernet links. IEEE Transactions on Communications 2012;60(11):3386–3395.

[15] Keilson J. Queues subject to service interruption. The Annals of Mathematical Statistics 1962;33(4):1314–1322.

[16] Knessl C, Yang YP. An exact solution for an M(t)/M(t)/1 queue with time-dependent arrivals and service. Queueing Systems: Theory and Applications 2002;40:233–245.

[17] Lee D. Analysis of a single server queue with semi-markovian service interruption. Queueing Systems: Theory and Applications 1997;27:153–178.

[18] Li H, Yu H, Yang B, Liu A. Timing control for delay-constrained data aggregation in wireless sensor networks. International Journal of Communication Systems 2007;20:875–887.

[19] Mogul J, Ramakrishnan K. Eliminating receive livelock in an interrupt-driven kernel. ACM Transactions on Computer Systems 1997;15:217–252.

[20] Mostowfi M, Christensen K. Saving energy in LAN switches: New methods of packet coalescing for Energy Efficient Ethernet. Proceedings of the International Green Computing Conference and Workshops 2011, 1–8.

[21] Nedevschi S, Popa L, Iannaccone G, Ratnasamy S, Wetherall D. Reducing network energy consumption via sleeping and rate-adaptation. Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation 2008;323–336.

[22] Odlyzko AM Data networks are lightly utilized, and will stay that way. Review of Network Economics 1999;2(3):210–237.

[23] Reviriego P, Maestro J, Hernandez J, Larrabeiti D. Burst transmission for Energy-Efficient Ethernet. IEEE Internet Computing 2010;14(4):50–57.

[24] Ruan X, Manzanares A, Yin S, Zong Z, Qin X. Performance evaluation of energy-efficient parallel I/O systems with write buffer disks. Proceedings of the International Conference on Parallel Processing 2009 164–171.

[25] Sengupta B. A queue with service interruptions in an alternating random environment. Operations Research 38(2):308–318.

[26] Skubic B, Hood D. Evaluation of ONU power saving modes for gigabit-capable passive optical networks. IEEE Network 2011;25(2):20–24.

[27] Takagi H. Queueing Analysis, Volume 1: Vacation and Priority Systems. Holland, Amsterdam. 1991.

[28] Takine T, Sengupta B. A single server queue with service interruptions. Queueing Systems: Theory and Applications 1997;26:285–300.

[29] White H, Christie LS. Queuing with preemptive priorities or with breakdown. Operations Research 1958;6(1):79–95.

**Allen Roginsky** works as Mathematician at the National Institute of Standards and Technology (NIST) in Gaithersburg, MD. His present research interest is Cryptography and its application in information security in distributed systems. Prior to switching to Cryptography, he spent a number of years with IBM Corporation analyzing and improving computer networks' performance. Allen received his Ph.D. in Statistics from the University of North Carolina at Chapel Hill in 1989. He has more than 30 publications and over a dozen US patents.

**Ken Christensen** is Professor and Director of the Undergraduate Program in the Department of Computer Science and Engineering at the University of South Florida. Ken received the Ph.D. in Electrical and Computer Engineering from North Carolina State University in 1991. His research interest is in performance evaluation of computer networks with a particular emphasis on power management of networks – "green networks". His research has been funded by NSF, KETI, Cisco, and Google. Before joining the faculty at USF, Ken was an Advisory Engineer at IBM in the Research Triangle Park. Ken has over 100 journal and conference publications and 13 U.S. patents. Ken is a licensed Professional Engineer in the state of Florida, a senior member of IEEE, and a member of ACM and ASEE.

**Mehrgan Mostowfi** is a doctoral candidate in the Department of Computer Science and Engineering at the University of South Florida. He received his B.S. degree in Computer Science from Beheshti University, Tehran, Iran, in 2005 and his M.S. degree in Computer Science from the University of South Florida in 2010. His research interests include power management and performance evaluation of computer networks.