On the Specification of NS and other Known On-Off Sources

Daniel R. Figueiredo, Benyuan Liu *
ratton, benyuan@cs.umass.edu

Department of Computer Science
140 Governor's Drive
University of Massachusetts
Amherst, MA 01003-4610

Computer Science Technical Report 00-25

June 14, 2000

Abstract

Traffic source is of great importance to networks modeling. Different traffic characteristics can cause other components to behave differently, resulting in different network performance measures. On-off source model has been widely adopted in networks research to capture the bursty nature of the network traffic; its behavior is precisely defined. In this work, we carefully analyzed the common structure and the behavior of on-off sources and show that the on-off source provided by ns simulator does not correspond to any of the known on-off models. This inconsistency between on-off sources has a direct impact on queueing performance measures. Modelers must be aware of this fact when using the ns simulator to compare or validate results with other modeling frameworks. We suggest that the ns on-off source be replaced by a consistent deterministic on-off source; the implementation and analysis were provided.

Keywords: traffic models, on-off source, performance analysis

*This material is based upon work supported by the National Science Foundation under Grant No. ANI-9809332. This work was also supported by CAPES (Brazil)

1 Introduction

The traffic source component is one of the most important entities in network modeling. The source behavior usually has a major impact on the overall performance evaluation of the model being studied. Different sources can easily lead to different performance measures [1]. For this very reason, it is important for sources to be well defined and have a specific behavior. This allows identical sources to be used in different frameworks, but still have the exact same behavior, which makes it possible to compare and validate the results across frameworks.

The on-off packet source is commonly used to model bursty network traffic and it has a well-defined behavior [4]. The source alters between the on and off states, remaining in each state for a certain amount of time, namely, the on period and the off period. While in the on period, the source generates packets according to some process. No packets are transmitted in off periods. This simple model requires a few parameters before it is fully described. The distribution of the on and off periods must be given, as well as the distribution for the packet inter-departure times. Besides this, it is necessary to specify whether the source generates the first packet as soon as it enters the on state, or after a inter-departure time interval.

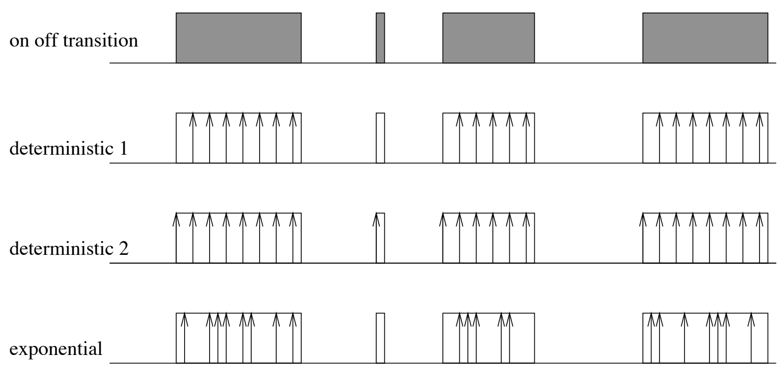

Figure 1: Different on-off sources

It is often the case that the on-off source model assigns an exponential distribution to on and off periods. From this model, we can derive three more specific models.

- Exponential inter-departure: In this model, an exponential distribution is used for the packet inter-departure process while in the on state.
- Deterministic-1: In this model, the packet inter-departure time is deterministic. Furthermore, when the source enters the on state, an inter-departure time interval must elapse before the first packet is generated.
- Deterministic-2: This model also has a deterministic packet inter-departure time. But the first packet is generated as soon as the source enters the on state.

All of these approaches are illustrated in Figure 1. Notice that both the Exponential and the Deterministic-1 sources allow on periods without packets being generated.

In the ns simulator [3], the Traffic/Expoo object([2]) models an on-off source. The manual and documentation claim that the source behaves like a deterministic on-off source. In other words, during on periods, packets are generated with a constant inter-departure time interval; on and off periods are taken from exponential distributions. However, the ns implementation of the source actually does something different.

First, the mean number of packets per on period is computed by dividing the average on period by the inter-departure time. At the outset of an on period, a random number is generated using an exponential distribution with the above mean value as the parameter. This number is then truncated to its closest integer and used as the number of packets to be generated in this on period. If the truncated number is zero, then the number of packets to be generated within this on period is set to one. Hence there is always at least one packet being sent per on period. In this implementation, the length of the on period is determined by the number of packets in that period. The off period is exponentially distributed. Figure 2 illustrates the behavior of the on-off source implemented by ns.

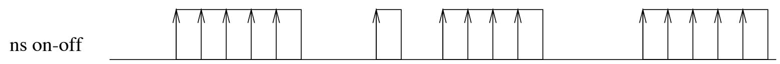

Figure 2: The ns on-off source

In this work, we show that the implementation of ns on-off source is not consistent with any of the known on-off source models, and that it behaves differently than the above three well-defined on-off sources. Furthermore, we will show that this difference can lead to different performance measures in queueing networks. The expected number of packets being transmitted per on period was derived for all the four sources. Our analysis shows that while ns claims to have a deterministic behavior, the expected number of packets per on period deviates from both known deterministic approaches. It actually approximates that of the exponential source. We also analyzed the impact of these sources feeding a FIFO queue and our results show that the performance measures of the queue obtained by ns is not consistent with any of the well-defined on-off source models.

The awkward behavior of ns on-off source and its effect on the queueing process must be kept in mind when modelers use the on-off source and compare or validate their simulation results with ns results. Different results between frameworks can just be due to the behavior of the on-off source.

Based on the fact that the ns on-off source transmits packets deterministically during on periods and always sends a packet at the onset of on periods, we suggest that the ns on-off source be replaced by the Deterministic-2 source. Besides this similarity, the Deterministic-2 source has a well-defined behavior which is easy to understand and reproduce. This recommended on-off source model was implemented in the ns framework, the source behavior and the queueing behavior using the source were validated.

The rest of the paper is organized as follows. In Section 2, we provide a careful analysis of the three well-defined on-off source models as well as the ns on-off source. The expected number of packets per on period for all models are compared. In Section 3, we investigate the impact of the different on-off sources on performance measures in a queueing network. In Section 4 we present the conclusion remarks. The implementation of the recommended source model in ns framework and the validations are presented in the appendix.

2 Analysis of source behavior

In this chapter we analyze and compare the expected number of packets generated per on period for the three well-defined on-off source models as well as the ns on-off source. The analytical results suggest a potential difference between on-off queueing models.

2.1 Exponential Inter-departure

Let T_{Int} be a random variable that denotes the time interval between packets. Since T_{Int} follows an exponential distribution with parameter λ, the packet arrival process given a fixed time interval t is just a Poisson process with expected value λt.

Let N be a random variable that denotes the number of packets transmitted in an on period. Let T_{ON} be a random variable that indicates the length of the on period. We assume the on period of all the on-off sources have an exponential distribution with parameter μ, ie, the pdf $f_{T_{ON}}(t) = \mu e^{-\mu t}$. Therefore, $E[N]$ can be computed as follows:

$$
\begin{aligned}
E[N] &= \int_0^\infty E[N = k | T_{ON} = t] f_{T_{ON}}(t) dt \\
&= \int_0^\infty \sum_{k=0}^{\infty} P(N = k | T_{ON} = t) k f_{T_{ON}}(t) dt \\
&= \int_0^\infty \sum_{k=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^k}{k!} k f_{T_{ON}}(t) dt \\
&= \int_0^\infty \lambda t f_{T_{ON}}(t) dt \\
&= \int_0^\infty \lambda t \mu e^{-\mu t} dt \\
&= \frac{\lambda}{\mu} \qquad (1)
\end{aligned}
$$

2.2 Deterministic Departure

In this case, packets are generated at a fixed rate λ, the time interval between packets is just $1/\lambda$. It is assumed that the on period is exponentially distributed with rate μ as in the previous analysis.

We now analyze the case when the first packet in on period is generated $1/\lambda$ time units after it starts (Deterministic-1). Given the length of the on period is between k/λ and $(k+1)/\lambda$, the number of packets transmitted in this on period is k. Therefore, $E[N]$ can be computed as follows:

$$
\begin{aligned}
E[N] &= \sum_{k=0}^{\infty} k P(k\frac{1}{\lambda} < T_{ON} \le (k+1)\frac{1}{\lambda}) \\
&= \sum_{k=0}^{\infty} k(P(T_{ON} \le (k+1)\frac{1}{\lambda}) - P(T_{ON} < k\frac{1}{\lambda})) \\
&= \sum_{k=0}^{\infty} k(e^{-\mu k \frac{1}{\lambda}} - e^{-\mu(k+1)\frac{1}{\lambda}}) \\
&= \frac{e^{-\frac{\mu}{\lambda}}}{1 - e^{-\frac{\mu}{\lambda}}} \qquad (2)
\end{aligned}
$$

We now analyze the case when the first packet in on period is generated as soon as the source enters the on state. It should be clear that the average number of packets per on period, $E[N]$, is just one more than the previous case, as shown below.

$$
\begin{aligned}
E[N] &= \sum_{k=0}^{\infty}(k+1)P(k\frac{1}{\lambda} < T_{ON} \leq (k+1)\frac{1}{\lambda}) \\
&= \sum_{k=0}^{\infty} k(P(T_{ON} \leq (k+1)\frac{1}{\lambda}) - P(T_{ON} < k\frac{1}{\lambda})) + \sum_{k=0}^{\infty}(P(T_{ON} \leq (k+1)\frac{1}{\lambda}) - P(T_{ON} < k\frac{1}{\lambda})) \\
&= 1 + \frac{e^{-\frac{\mu}{\lambda}}}{1-e^{-\frac{\mu}{\lambda}}} \\
&= \frac{1}{1-e^{-\frac{\mu}{\lambda}}} \qquad (3)
\end{aligned}
$$

Confirming the intuition, the expected number of packets per on period differs exactly by 1.

2.3 Ns on-off source

For an ns on-off source, the interval between packet transmissions is also deterministic. However, its implementation is quite different from the above models. It tries to capture the behavior of the on state using the number of packets transmitted in that period. As soon as the source enters the on state, ns computes the number of packets to be transmitted in that on period, which is also used to determine the length of the period. This random variable, Q, has an exponential distribution with average $E[Q] = \frac{E[T_{ON}]}{1/\lambda} = \frac{\lambda}{\mu}$. Notice that this average does not match any of the above deterministic models. Q should denote the number of packets generated per on period. However, Q is a continuous random variable (as opposed to N which is discrete) and ns rounds this number to its closest integer, introducing round off errors. This results in an on period whose length is no longer exponentially distributed, since a fraction of packet cannot be transmitted. Besides, the value for Q is always at least one. Therefore, this model also generates at least one packet per on period. We computed the expected value of the number of packets generated per on period, which we denote by N.

$$
\begin{aligned}
E[N] &= \sum_{k=1}^{\infty} kP((k-1)+0.5 \leq Q < k+0.5) + 1P(Q \leq 0.5) \\
&= \sum_{k=1}^{\infty} k(e^{-\frac{\mu}{\lambda}(k-0.5)} - e^{-\frac{\mu}{\lambda}(k+0.5)}) + (1-e^{-0.5\frac{\mu}{\lambda}}) \\
&= e^{0.5\frac{\mu}{\lambda}}\sum_{k=1}^{\infty} ke^{-\frac{\mu}{\lambda}k} - e^{-0.5\frac{\mu}{\lambda}}\sum_{k=1}^{\infty} ke^{-\frac{\mu}{\lambda}k} + (1-e^{-0.5\frac{\mu}{\lambda}}) \\
&= (e^{0.5\frac{\mu}{\lambda}} - e^{-0.5\frac{\mu}{\lambda}})\sum_{k=1}^{\infty} ke^{-\frac{\mu}{\lambda}k} + (1-e^{-0.5\frac{\mu}{\lambda}}) \\
&= e^{0.5\frac{\mu}{\lambda}}(1-e^{-\frac{\mu}{\lambda}})\sum_{k=0}^{\infty} ke^{-\frac{\mu}{\lambda}k} + (1-e^{-0.5\frac{\mu}{\lambda}}) \\
&= e^{0.5\frac{\mu}{\lambda}}(1-e^{-\frac{\mu}{\lambda}})\frac{e^{-\frac{\mu}{\lambda}}}{(1-e^{-\frac{\mu}{\lambda}})^2} + (1-e^{-0.5\frac{\mu}{\lambda}})
\end{aligned}
$$

$$= \frac{e^{-0.5\frac{\mu}{\lambda}}}{1-e^{-\frac{\mu}{\lambda}}} + (1 - e^{-0.5\frac{\mu}{\lambda}})$$
$$= 1 + \frac{e^{-1.5\frac{\mu}{\lambda}}}{1-e^{-\frac{\mu}{\lambda}}} \quad (4)$$

From this result, we can see that $E[N]$ of ns does not match any of the above well defined on-off source models. Therefore, measures that characterize the source behavior, such as the average number of packets per on period ($E[N]$), have different values.

Another important measure that also characterizes the behavior of a source is the average packet rate generation. In the on-off source model, this measure is given by the expected number of packets generated in the on period divided by the sum of the expected length of the on and off periods: $E[\text{Packet\_Rate}] = \frac{E[N]}{E[T_{ON}]+E[T_{OFF}]}$, where N is the number of packets per on period (defined above) and T_{ON} and T_{OFF} are the random variables associated with the on and off periods. Our definition of the on-off source model assumes that the on and off periods have exponential distribution with parameters μ and γ, respectively. Therefore, one would expect that $E[\text{Packet\_Rate}] = \frac{E[N]}{1/\mu+1/\gamma}$ for all the on-off sources presented. This is true for all the three well-defined on-off models. However, this is not the case for the ns on-off source model, since the the length of its on period is determined by the number of packets generated in that period. Therefore, the distribution of the on period is not exponential and is given by the distribution of the number of packets per on period multiplied by the inter-departure packet time. Thus, for the ns on-off source we have $E[\text{Packet\_Rate}] = \frac{E[N]}{1/\lambda E[N]+1/\gamma}$, where λ is the packet rate of the on period. Notice that besides having a different value for $E[N]$, the ns on-off source also has a different formula for $E[\text{Packet\_Rate}]$, which suggests a different formula for the load, given by $E[\text{Packet\_Rate}]/E[\text{Service\_Rate}]$. This difference in the load calculation can lead to incorrect results when one is not careful and uses the wrong formula to set the load in simulations.

2.4 Comparison between the sources

We now present a graphical illustration of the analysis above and. The parameter for comparison is the rate $\frac{\lambda}{\mu}$, which in the exponential case indicates the average number of packets per on period. In this comparison, we fixed the value of μ (equal to 5) and varied λ. Note that λ indicates the packet rate of the on period and that other values of μ will also lead to similar results.

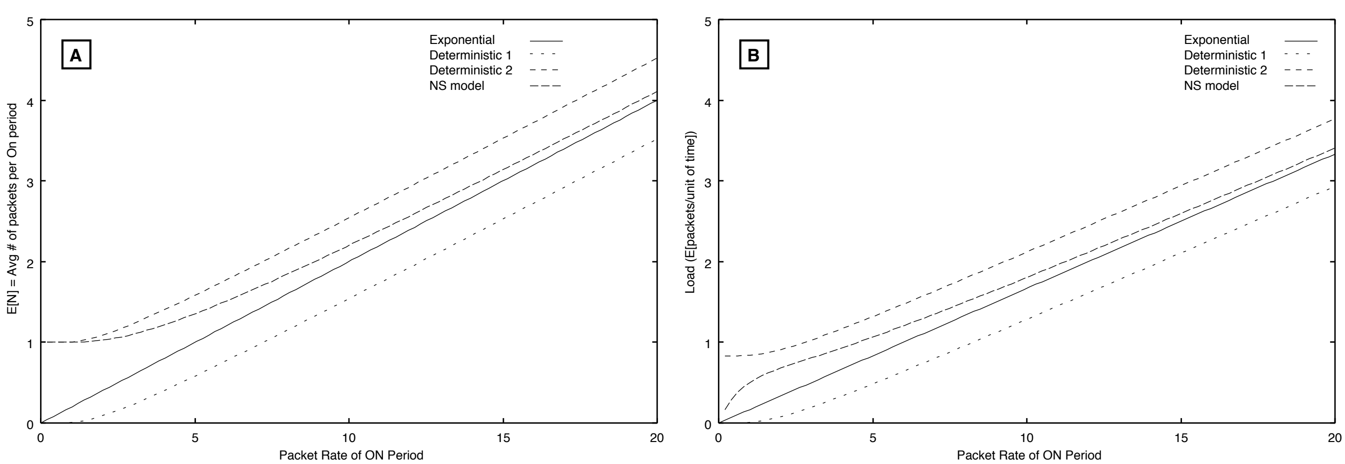

Figure 3: Comparison between different On-off sources ($\mu = 5$)

In Figure 3 we compare behaviors of different on-off sources by analyzing the expected number of packets per on period (graph A), and the expected packet rate of the source (graph B). The three known on-off sources are plotted together with the ns source as a function of λ. Notice that the exponential model lies about halfway between the two deterministic models. The first deterministic approach always has one packet less than the second approach for $E[N]$ (graph A) and a constant factor for $E[\text{Packet\_Rate}]$ (right graph). For small values of λ, $E[N]$ for ns approximates the second deterministic case while for larger values it approximates the exponential model of the on-off source. A similar trend can also be observed for $E[\text{Packet\_Rate}]$, where ns approximates the exponential model for high values of λ. Therefore, the implementation of the deterministic on-off source in ns has a strange behavior. It approximates the exponential on-off source as the transmission rate increases, when the number of packets generated per on period or the average packet rate are considered. This fact was also confirmed by simulation results using the ns simulator for the ns on-off source model and another simulation framework for others on-off source models.

3 Impact of sources on a queue

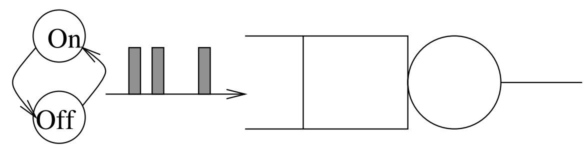

Figure 4: A single on-off source feeding a single FIFO queue

In this section we investigate how queueing performance measures are affected by the different on-off source behaviors. In the experiment, an infinite capacity FIFO queue with constant service rate was fed by a single on-off source, as depicted in Figure 4. We fixed the average on/off period, but varied the packet inter-departure time and measured the queue utilization, average queue length, and average packet waiting time for all the four on-off source models.

In Figure 5 we observe the utilization of the queue as a function of the mean number of packets per on period. We observe that the exponential model falls halfway between the two deterministic models, and the ns model approaches exponential model. This is not surprising since the queue utilization is mainly decided by the source packet rate under work conserving policies.

With respect to the average queue length, not only the average arrival rate, but the variance of the packet inter-arrival time also plays an important role. A large variance in packet inter-arrival times tends to increase the average queue length. The deterministic models all have fixed inter-arrival times, while the exponential model generate packets with a varying time interval. Figure 6 shows the average queue size for different packet rates. At 50% utilization, we observe that the ns source behaves like the Deterministic-2 source. However, this does not hold at 90% utilization, in which case the ns source behavior falls between the two deterministic sources. This indicates that the ns source does not have a consistent behavior when compared with any of the three known models.

At 90% utilization, the increase in the average queue length is due to the fact that the number of packets that must be served within the same amount of time is bigger for larger packet rate. This larger number of packets arriving at the queue contributes to the average queue size.

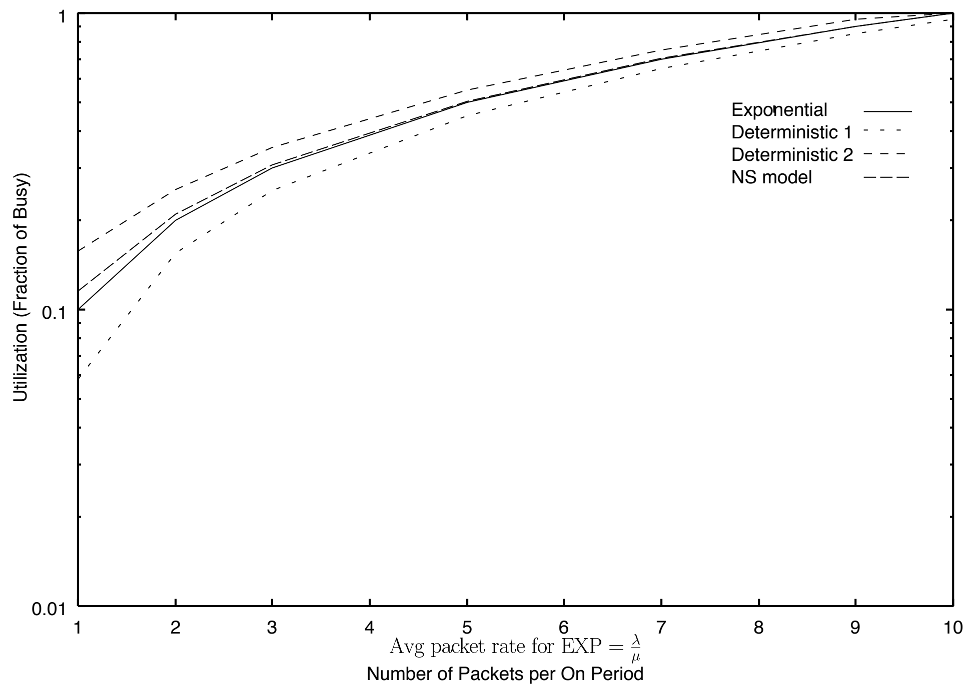

Figure 5: Utilization of the queue

To summarize, the ns on-off source approximates the exponential model in terms of queue utilization, approximates the deterministic models in terms of average queue length and queueing delay in the median utilization range, but deviates away and approximate exponential model in the low and high utilization regions. Hence the queueing performance measures using the ns on-off source is not consistent with any of the well-defined three on-off source model. We conclude that the behavior of the ns on-off source does not correspond to any known model.

We now investigate the effect of aggregated sources feeding a single FIFO queue. Intuitively, the effect of different sources in queueing performance measures should diminish as the number of sources increases. This is expected because multiplexing a large number of sources tends to smooth out the packet arrival process. Figure 7 depicts the average queue size and average waiting time for the three well-defined on-off sources and the ns on-off source for different queue utilizations. The average waiting time is shown for utilizations of 20%, 50% and 90%. To obtain

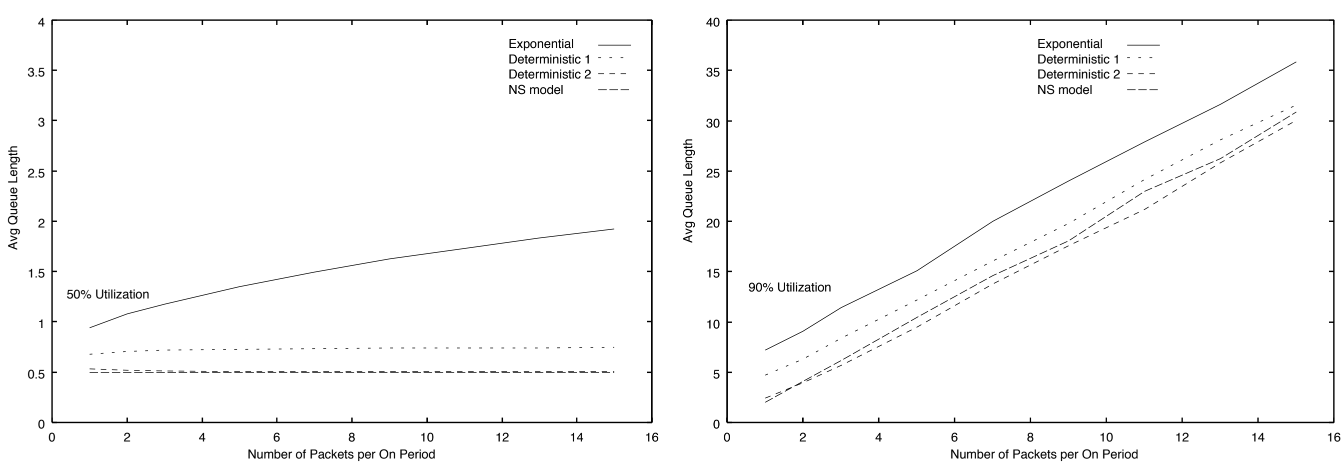

Figure 6: Average queue length

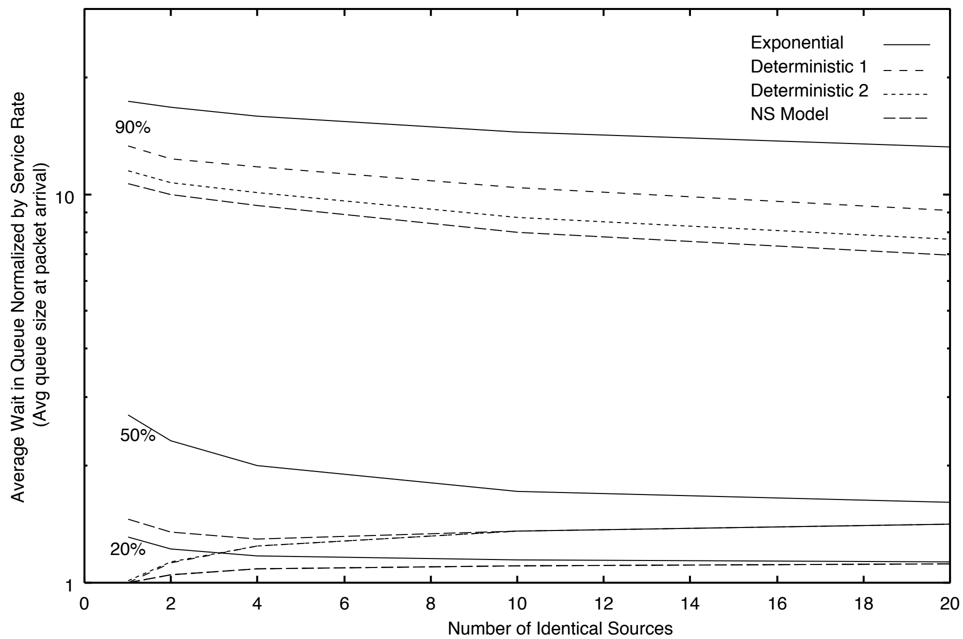

Figure 7: Comparison between different aggregated sources

this results we used the ns queueing monitor, which doesn't account for the packet in service in its measures. Due to this incompatibility in the performance measures among the simulators used, only the waiting time is presented.

Notice that the differences in performance measures between the three source models becomes smaller as more sources are aggregated. Also note that for higher utilization, the convergence rate is smaller and the differences between the sources persist. In this scenario, the ns on-off source model performs in between the two well-defined deterministic sources. However, for low utilizations, the ns on-off source behaves very close to the Deterministic-2 on-off source, which always generates a packet at the onset of the on period. Recall that the queueing behavior of a single ns on-off source also falls between the two deterministic sources at high utilizations, and is very close to the Deterministic-2 source at low utilizations, as illustrated in Figure 6.

Confirming our intuition, the aggregation of identical sources smoothes out the difference between the three source models. However, for high utilization, the difference in performance measures can still be significant, especially if used for validation purposes among different frameworks.

4 Conclusions

In this work we analyzed three well-defined on-off source models as well as the on-off source implemented in ns simulator. Although the ns source model generates packets at a deterministic time interval, the expected number of packets per on period is quite different from the two deterministic model under the same parameter setting. It actually approximates the exponential model when packet rate is large. Different sources have direct impacts on queueing performance measures of a network model, and it turns out that the ns on-off source has a unique behavior different than any of the three well-defined on-off sources. Modelers must be aware of this fact when they use the ns on-off source to compare or validate the results with ones obtained from other frameworks. Due to the similarity between the ns on-off source and the Deterministic-

2 source in the sense that they both transmit packet deterministically and always sends a packet at the beginning of an on period, we suggest that the ns on-off source be replaced by the Deterministic-2 source model. The implementations and validations were provided in the appendix.

Appendix

A Suggestion for the ns on-off source

The ns source should have a well-defined behavior to avoid multiple interpretations of its functionality. Based on the results obtained and the definition in the manual, the ns on-off source should behave like the Deterministic-2 source. We now propose a change in the source code of the on-off source to implement the deterministic 2 source. This modification does not introduce any additional computational cost, since no new random number generation is required.

The required modifications take place in the *next_interval()* procedure, which determines when the next packet is going to be generated. We also introduce two new attributes to the EXPOO_Traffic class. One of them is the state_ variable and keeps track of the current state of the source. This variable can assume values 1 (on state) or 0 (off state). The other variable is time_rem_, which indicates the amount of time between the last packet transmitted in the on period and the end of the on period. Maintaining this value is necessary since the on period does not end exactly after its last packet is sent.

It's important to know that every time the function *next_interval()* is called by the core of the simulator a packet is generated. This function also returns the time that the next call should take place. The objective of the function is then to determine when will the next packet be generated. We do this by keeping track of the number of packets in the on period and the state of source. When this number reaches one, the last packet of the current on period is being transmitted and the next transmission (call to the function) should take place after the remaining of the current on period plus the off period. In this case, the state is set to zero to indicate a off period. If the on period is very short (less than the inter-arrival packet time), then we have no remaining packet to send and the next packet will be generated after the remaining of this on period plus the subsequent off period. This algorithm is implemented in the EXPOO_Traffic class.

This new C++ class was compiled, linked and tested under ns. We measured the behavior of this new source and compared with the Deterministic-2 source using another simulator. As illustrated in Figure 8 the two sources have exactly the same behavior. In this figure, the number of packets generated by both sources in 100000 simulation time units is presented for different packet rates of the on period. We also measure the waiting time of a packet in the queue when fed by these sources. Notice that there are no differences in the measures obtained. Other measures also agreed very well. This indicates that the new implementation is well-defined and can be easily reproduced by other simulation frameworks.

Below is the implementation of the EXPOO_Traffic class with the suggested modifications. We would like to note that minimal change was done to the already existing class and some

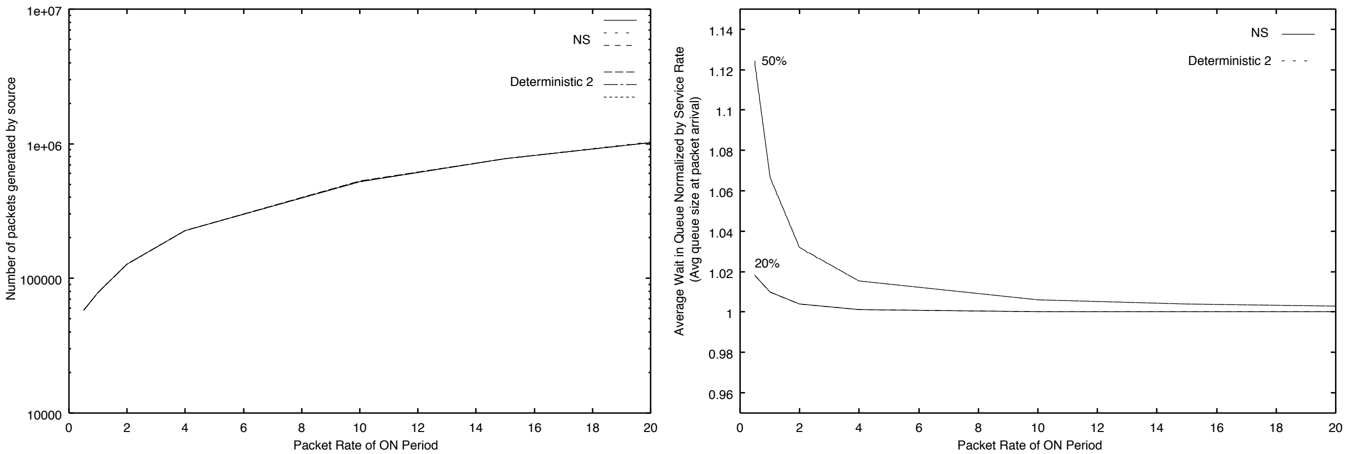

Figure 8: Comparison between the modified ns source and the Deterministic-2 source

attributes not being used were maintained. No code review, concerning other attributes and methods was done.

```
class EXPOO_Traffic : public TrafficGenerator {
 public:
   EXPOO_Traffic();
   virtual double next_interval(int&);
   virtual void timeout();
 protected:
   void init();
   double ontime_;   /* average length of burst (sec) */
   double offtime_;  /* average length of idle time (sec) */
   double rate_;     /* send rate during on time (bps) */
   double interval_; /* packet inter-arrival time during burst (sec) */
   unsigned int rem_; /* number of packets left in current burst */
   unsigned int state_; /* state of the on-off source */
   double time_rem_; /* time remaining at the end of the on period */

   /* new stuff using RandomVariable */
   ExponentialRandomVariable Ontime_;
   ExponentialRandomVariable Offtime_;

};


static class EXPTrafficClass : public TclClass {
 public:
   EXPTrafficClass() : TclClass("Application/Traffic/Exponential") {}
   TclObject* create(int, const char*const*) {
         return (new EXPOO_Traffic());
   }
} class_expoo_traffic;

EXPOO_Traffic::EXPOO_Traffic() : Ontime_(0.0), Offtime_(0.0)
{
```

```
   bind_time("burst_time_", Ontime_.avgp());
   bind_time("idle_time_", Offtime_.avgp());
   bind_bw("rate_", &rate_);
   bind("packet_size_", &size_);
}

void EXPOO_Traffic::init()
{
   /* compute inter-packet interval during bursts based on
   /* packet size and burst rate.  then compute average number
    * of packets in a burst.
    */
   interval_ = (double)(size_ << 3)/(double)rate_;
   rem_ = 0;
   state_ = 0;
   if (agent_)
      agent_->set_pkttype(PT_EXP);
}

double EXPOO_Traffic::next_interval(int& size)
{
   double t;
   double on_time;

   /* if coming from the off state */
   if (state_ == 0) {
      /* generate the size of the on period */
      on_time = Ontime_.value();
      /* calculate the number of packets in this period */
      rem_ = int(on_time / interval_);
      /* calculate the residual time between the last packet */
      /* and the start of the off period */
      time_rem_ = on_time - (rem_ * interval_);
      /* set the state to on if there are packets to send */
      if (rem_ > 0) {
         state_ = 1;
         t = interval_;
      } else
         /* if this is a very short on period, calculate the */
         /* start of the next on period */
         t = time_rem_ + Offtime_.value();
      }

      /* if coming from the on state */
   else {
      /* if this is not the last packet just decrease the number */
      if (rem_ > 1) {
         t = interval_;
         rem_--;
      } else {
         /* last packet. calculate the start of the next on period */
```

```
            state_ = 0;
            t = time_rem_ + Offtime_.value();
         }
      }

      size = size_;
      /* return the time of the next call to this function */
      return(t);
}

void EXPOO_Traffic::timeout()
{
   if (! running_)
      return;

   /* send a packet */
   // The test tcl/ex/test-rcvr.tcl relies on the "NEW_BURST" flag being
   // set at the start of any exponential burst ("talkspurt").
   if (nextPkttime_ != interval_ || nextPkttime_ == -1)
      agent_->sendmsg(size_, "NEW_BURST");
   else
      agent_->sendmsg(size_);
   /* figure out when to send the next one */
   nextPkttime_ = next_interval(size_);
   /* schedule it */
   if (nextPkttime_ > 0)
      timer_.resched(nextPkttime_);
}
```

References

[1] J. N. Daigle and J. D. Langford. Models for analysis of packet-voice communication systems. *IEEE Journal on Selected Areas in Communications (JSAC)*, pages 847–855, Sept. 1986.

[2] The Network Simulator Man Pages URL http://www mash.cs.berkeley.edu/ns/ns-man.html. Lawrence Berkeley National Laboratory - University of California, Berkeley.

[3] Steven McCanne and Sally Floyd. The LBNL network simulator URL http://www-nrg.ee.lbl.gov/ns. Lawrence Berkeley National Laboratory - University of California, Berkeley.

[4] Mischa Schwartz. *Broadband Intergrated Networks.* Prentice Hall, 1996.