AN ANALYSIS APPROACH TO MULTI LEVEL NETWORKING

Ioannis Stavrakakis
Department of Electrical Engineering and Computer Science
University of Vermont, Burlington, VT 05405

Abstract

The major problem in the analysis of large packet communication networks is related to the difficulty in characterizing the involved packet traffic. In this paper, a general description of the packet processes generated by network components is presented. Based on this description, an analysis approach to multi level networking is developed. The packet traffic generated by a lower level (which is the input traffic to an upper level) is described as a generalized Bernoulli process whose intensity depends on the state of that level. This characterization is shown to be exact, or a well performing approximate one, for many practical systems. The general approach is illustrated through the analysis of a specific multi level packet communication network.

I. Introduction

In this paper, a network component is defined as a network subsystem which generates packets. Such components can be a local area network, a network switch or a repeater, a statistical multiplexer, a link that carries traffic from another large network, or any line carrying packetized information of virtually any kind. We define the First Stage Component (FSC) to be a network component whose packet input process is determined by simple first-born traffic. The simple first-born traffic is defined as a traffic which is not modulated by any network component. This type of traffic is modeled as an independent process.

In a large packet communication network (defined here as a network with more than one components), there is always a mechanism which provides for the backbone network that supports the interconnection of the involved components. Clearly, (some of) the packet processes in this system are the output processes from the supported components. We define the Second Stage Component (SSC) to be a system whose input process is (at least partially) determined by the output processes of other network components.

The performance evaluation of a large network requires the analysis of all FSCs and SSCs. The analysis of FSCs has received much attention in the past and many analytical techniques have been developed for that purpose. The nature of the models for the simple first-born traffic has played an important role in the development of these techniques. The analytical tractability provided by the Bernoulli / Poisson models for the simple first-born traffic has tempted many researchers to adopt these models for the description of the input processes to SSCs as well, [1]-[3]. Such models are usually inaccurate since, although the input process to a FSC might be considered as memoryless, the FSC introduces dependencies to its output process.

In this paper, an alternative to the i.i.d. (independent and identically distributed) characterization of the packet (output) processes generated by the components of a communication network, is presented. The proposed model is exact and it is presented in the next section. The output process is described as a generalized Bernoulli process whose intensity depends on the state of an underlying Markov chain describing the operation of the component. The cardinality of the state space of this Markov chain may be almost arbitrarily large. To provide for a numerically tractable solution in the later case, the exact model is appropriately modified. For this purpose, a meaningful approximate model on the output process, based on the dominant states of the underlying Markov chain, is introduced. The approximate model is described by introducing the concept of the $\alpha\%$-exact model.

II. Models for the output processes.

II.1 The exact model.

The description of the dependent packet process generated by a network component (FSC or SSC) is essential to the analysis of a SSC since the input processes to the latter are determined by the packet
processes generated by the feeding network components. The proposed exact model on the output (or packet departure) process of a network component is described through the following definitions. The discussion in this paper is confined to discrete time network components, as formulated by slotted packet communication systems.

**Definition 1** The packet generation (output) process of a network component is defined to be the discrete time process of the departing packets, \( \{a_j\}_{j=0}^\infty \), where \( 0 \leq r < \sigma \), if \( r \) packets leave the component at the \( j^{th} \) time instant.

**Definition 2** Assume that the network component satisfies the following:

(a) There exists an ergodic Markov chain \( \{z_j\}_{j=0}^\infty \) associated with the description of the state of the component; let \( \mathcal{S} = \{x_1, x_2, \ldots, x_M\} \), \( M < \infty \), be the state space of \( \{z_j\}_{j=0}^\infty \) and \( p(x_k, x_j) \), \( \pi(x_j) \), \( x_k, x_j \in \mathcal{S} \), be the corresponding state transition and steady state probabilities.

(b) There exists a stationary probabilistic mapping \( a(z_j) : \mathcal{S} \to \mathbb{Z}_0 \) where \( \mathbb{Z}_0 \) is the set of nonnegative finite integers, which describes the number of packets departing at the end of the \( j^{th} \) time interval (slot). Let \( a(z_j) = \rho, \quad 0 \leq \rho \leq \sigma, \quad z_j \in \mathcal{S} \), with probability \( \phi_\rho(z_j) \).

Then, the output process of the component is given by

\[
\{a_j\}_{j=0}^\infty = \{a_j(z_j)\}_{j=0}^\infty
\]

i.e., it is described as a Markov modulated generalized Bernoulli process. Notice that the process \( \{a_j\}_{j=0}^\infty \), as given by (1), describes exactly the output process of a network component, provided that the conditions in Definition 2 are satisfied.

### II.2. The \( \alpha \%-exact \) model.

To define the \( \alpha \%-exact \) model on the dependent process generated by a network component consider two processes \( \{z_j\}_{j=0}^\infty \) and \( \{a_j(z_j)\}_{j=0}^\infty \), as described in Definition 2, associated with the component. We define the new Markov chain \( \{x_j\}_{j=0}^\infty \) with state space \( \mathcal{S} = \{x_1, \cdots, x_M\} \) for some \( M < \infty \). The states \( x_1, \cdots, x_{M-1} \) of the new Markov chain are identical to the \( M-1 \) dominant states of the original Markov chain; \( x_M \) is the union of the remaining states of the original Markov chain. Dominant states are the states with the largest probability mass. The steady state and state transition probabilities of the new Markov chain are obtained by appropriate averaging of those of the original Markov chain. We define \( \{x_j\}_{j=0}^\infty \) to be a process similar to \( \{z_j\}_{j=0}^\infty \) with a corresponding underlying Markov chain \( \{z_j\}_{j=0}^\infty \) and probabilistic mapping \( a(z_j) \): \( \mathcal{S} \to \mathbb{Z}_0 \) obtained by appropriate averaging of that of the original output process.

Clearly, the process \( \{x_j\}_{j=0}^\infty \) is an approximation on the true packet output process of the network component described by \( \{a_j\}_{j=0}^\infty \). The approximation is introduced by the reduction of the state space of the original underlying Markov process \( \{z_j\}_{j=0}^\infty \), which describes completely the operation of the component. In view of the construction of the approximate process \( \{a_j\} \), it is reasonable to expect that as \( M \) increases, \( \{a_j\}_{j=0}^\infty \) approaches \( \{x_j\}_{j=0}^\infty \); that is, the approximate model approaches the exact one. When \( M = \infty \), the approximate model coincides with the exact one. It is also reasonable to expect that the larger the total probability mass of the (unchanged) \( M-1 \) states the better the approximation achieved by the reduced state space process. In view of the previous observations, we provide the following definition-measure of the approximation on the exact process \( \{a_j\}_{j=0}^\infty \).

**Definition 3** The process \( \{a_j\}_{j=0}^\infty \) defined as before is called \( \alpha \%-exact \) if the cardinality \( M \) of the corresponding approximate underlying Markov chain \( \{z_j\}_{j=0}^\infty \) (as defined before) is such that,

\[
\sum_{1 \leq j \leq M-1} \pi(z_j) \geq \alpha \quad \text{and} \quad \sum_{1 \leq j \leq M-2} \pi(z_j) < \alpha
\]

That is, the total probability mass of the unchanged states of the original Markov chain \( \{a_j\}_{j=0}^\infty \) is at least \( \alpha \) and the total probability mass of the \( M-2 \) dominant states is less than \( \alpha \).

Assuming ergodicity of all processes involved, Definition 3 can be roughly interpreted in the following way: An \( \alpha \%-exact \) output process \( \{a_j\}_{j=0}^\infty \) is based on the true packet generating mechanism (state of the system) \( \alpha \%-\) of the time; \((100-\alpha)\%\) of the time, the output process is based on an average packet generating mechanism. An average packet generating mechanism is the only one assumed present when \( \{a_j\}_{j=0}^\infty \) is approximated by a (generalised) Bernoulli process. The latter case corresponds to merging all states of the true underlying Markov chain into a single one (\( M = 1 \)). An average number of outputs is generated under the latter model throughout the time horizon, independently of the true state of the underlying packet generating mechanism. In view of Definition 3, the exact model on the output process \( \{a_j\}_{j=0}^\infty \) corresponds to \( \alpha = 100 \). \( 100-\alpha \) can also be seen as a measure of the smoothing on the output process introduced by the merging of the states of the true packet generating mechanism.

From the definition of the \( \alpha \%-exact \) process and the construction of \( \{x_j\}_{j=0}^\infty \) turns out that the cardinality of the state space of \( \{x_j\}_{j=0}^\infty \) (i.e. \( M \)) increases with \( \alpha \). Thus, it is reasonable to expect that the
larger the value of $\alpha$ the better the approximation on
the packet output process. Assuming that this is gen-
erally true, there is a trade off between the degree of
the accuracy of the approximating process and the
introduced complexity in its description, as measured
by $M$. The real concern at this point is not about
the complexity in the description of the approximating (or
exact) packet output process itself, but it is on the
tractability of the analysis of other components of a
large network, whose input processes are described by
the proposed models.

III. The output process of some network components

Example 1: Bursty traffic network links.

Consider a link which carries traffic modulated
by various other components of a large network and
by routing decisions. The network component in this
case is the link and its input and output processes are
identical. In [4] it has been found that network
packet traffic is bursty. As a result, a first order Mark-
ov model could be adopted in the description of this
packet process. If $p(0,1)$ and $p(1,1)$ are the condi-
tional probabilities of a packet arrival (departure)
given that 0 or 1 arrivals (departures) occurred in the
previous slot, respectively, then the burstiness
coefficient is defined, [4], as
\[
\gamma = p(1,1) - p(0,1)
\]
This traffic model can be easily described in terms of
the proposed model on the packet process generated
by the link (network component). The parameters of
the packet output process of a network component
which generates bursty traffic can be determined from
the packet rate $\pi(1)$ and the burstiness coefficient $\gamma$.

Example 2: The single message node

Consider now a network node which is capable of
storing and forwarding a single message at a time. It
is assumed that the input process to this component is
Bernoulli with intensity $\mu$ messages per slot. Each
message is assumed to consist of a variable number of
packets; let $\sigma(i) = \text{Pr}\{\text{message consists of } i \text{ packets}\}$.
$1 \leq i \leq K$. The single message buffering assumption
implies that messages which find the component non-
empty are either discarded or served by a (buffered)
low priority link. Without loss of generality, it is
assumed that a new message is also accepted if there
is only one packet (the last of the previous message)
in the node. It is assumed that arrivals occur at the
beginning of a slot. As a result, a new message may
start being served right after the end of the previous
message transmission. The packet output process of
this component is definitely a non-Bernoulli process.
It can be easily described in terms of the processes
$\{x_j\}_{j=0}^{\infty}$ and $\{z_j\}_{j=0}^{\infty}$ defined in the previous section.
If $x_j$ is the number of packets in the node at the end of
the $j^{th}$ slot, then it can be easily shown that $\{x_j\}_{j=0}^{\infty}$ is
a Markov chain with state space $S = \{0,1,2,\ldots,K\}$.
The nonzero transition probabilities are given by
\[
p(0,1) = \pi(1,1) - \mu \sigma(1), \quad 1 \leq i \leq K
\]
\[
p(0,0) = p(0,1) = 1 - \mu, \quad p(k,k-1) = 1 \text{ for } 2 \leq k \leq K
\]
Given $\mu$ and $\sigma(i)$, $1 \leq i \leq K$, the steady state prob-
babilities $\pi(i)$, $0 \leq i \leq K$ can be easily computed. The map-
ing given by (1) is deterministic in this case and has the
following parameters
\[
\phi_1(k) = 1, \quad 1 \leq k \leq K \quad \Rightarrow \phi_0(k) = 0
\]
\[
\phi_0(k) = 1 - \phi_1(k), \quad 0 \leq k \leq K
\]
A Bernoulli approximate model on the output process
of the node would have intensity $1 - \pi(0)$. A better
approximate model on the true packet output process
would be a first order Markov model. If 1 and 0
 denote one or zero packet outputs, respectively,
then the parameters of this Markov model are given by
\[
\pi_m(0) = \pi(0), \quad \pi_m(1) = 1 - \pi_m(0)
\]
\[
p_m(0,0) = 1 - p_m(0,1), \quad p_m(1,0) = 1 - p_m(1,1)
\]
\[
p_m(0,1) = \mu, \quad p_m(1,1) = 1 - p_m(0,1) = \frac{\pi_m(0)}{\pi_m(1)}
\]

Example 3: A node with arbitrarily large buffer.

In this case it is assumed that all messages which
would fit into the buffer of size $M<<\infty$ are received;
no message is partially received. Let $g(k)$, $0 \leq k \leq K$,
be the probability that a message with k packets
arrives over a slot; $k=0$ corresponds to no message
arrival.

Similarly to the previous example, the output
process of the finite buffer node can be easily
described in terms of the processes $\{x_j\}_{j=0}^{\infty}$ and
$\{z_j\}_{j=0}^{\infty}$. If $x_j$ is the number of packets in the node at
the end of the $j^{th}$ slot, then $\{x_j\}_{j=0}^{\infty}$ is a Markov chain
with state space $S = \{0,1,2,\ldots,M\}$. The transition
probabilities are given by (assume $g(k)=0$ for $k>M$)
\[
p(0,j) = g(0), \quad 0 \leq j \leq M
\]
\[
p(k,j) = g(j-k+1), \quad 1 \leq k \leq M, \quad k-1 \leq j \leq M
\]
and the probabilistic mapping is given as in Example
2. The Bernoulli and the Markov approximations on
the resulting packet output traffic can be determined
as in the previous example.

IV. Performance analysis of a large network

In this section, the multi-component network
Let $C_1$ be the network component described in Example 1 with parameters $\pi(1) = .1$ and $\gamma = .3$. Let $C_2$ be the network component described in Example 2 with parameters $K=5$, $\sigma(0) = 0$, $\sigma(1) = .1$, $\sigma(2) = .3$, $\sigma(3) = .3$, $\sigma(4) = .2$, $\sigma(5) = .1$ and $\mu = .1$, which result in a packet output rate of .244 packets per slot. Finally, let $C_3$ be the network component described in Example 3 with parameters $M=50$ (buffer size), $r=g(0) = .8$ (probability of no message arrival in a slot), $g(1) = .1(1-r)$, $g(2) = .3(1-r)$, $g(3) = .3(1-r)$, $g(4) = .2(1-r)$ and $g(5) = .1(1-r)$, where $g(k)$, $k=1,...,5$, is the probability that a message consists of $k$ packets. The output processes of components $C_1$ and $C_2$ are exactly described as mentioned in section III. For the description of the output process of component $C_3$, both, the exact (which involves an underlying Markov chain with a state space of cardinality 51) and the a%-exact approximate (which involves an underlying Markov chain of cardinality less than 51) models, are adopted (see Section II).

For the network described above, the mean packet delay induced by the SSC $C_0$ is shown in Table I. Exact results, obtained by incorporating the exact models on the input processes to $C_0$, are shown. Results for various degrees of approximation, $\alpha$, of the output process of $C_3$ are shown, as well. For a certain value of $\alpha$, the range of the dominant states is also shown. It can be easily seen that, e.g., the 97%-exact model (resulting in a Markov chain of cardinality 10) gives mean delay results in $C_0$ very close to the accurate ones obtained by solving 5 times more linear equations. This observation indicates that a significant reduction in the numerical complexity of the problem can be achieved by incorporating an approximate model based on the dominant states of a component, at the expense of an insignificant deviation from the accurate results. Other possible approximations on the true output process, which also simplify the analysis, are the Bernoulli and the Markov ones. Results under these approximations are shown in Table I. As it can be easily concluded the Bernoulli model completely fails to approximate the accurate result, while the Markov model is clearly inferior to the $\alpha%$-exact model for sufficiently large $\alpha$. Some more conclusions about the performance of the $\alpha%$-exact model are drawn in the next case.

Case 2:

Let $C_3$ be as in Case 1 and $C_1$ and $C_2$ be network components as described in Example 1. Let $\pi(1) = .17$ and $\gamma = .3$ for each of the components $C_1$ and $C_2$. The delay results in $C_0$ under various models on the output processes of $C_1$, $C_2$, and $C_3$ are given in Table II, for $\gamma = .3$, and in Table III for $\gamma = .0$; the latter value of $\gamma$ corresponds to Bernoulli traffic.

For a very small value of $\alpha$ only the most significant state will be considered, while the rest of them will be merged into a single state. This situation is reflected for $\alpha = 1$ where the state 0 (the most significant one) is selected. Let $1^*$ denote the new
<table>
<thead>
<tr>
<th>Model</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%(0-50)</td>
<td>14.655</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>4.174</td>
</tr>
<tr>
<td>Markov</td>
<td>13.427</td>
</tr>
<tr>
<td>85%(0-5)</td>
<td>13.720</td>
</tr>
<tr>
<td>90%(0-6)</td>
<td>13.959</td>
</tr>
<tr>
<td>95%(0-8)</td>
<td>14.295</td>
</tr>
<tr>
<td>97%(0-9)</td>
<td>14.388</td>
</tr>
<tr>
<td>99%(0-12)</td>
<td>14.515</td>
</tr>
</tbody>
</table>

Table I

<table>
<thead>
<tr>
<th>Model</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%(0-50)</td>
<td>12.884</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>4.060</td>
</tr>
<tr>
<td>Markov</td>
<td>10.604</td>
</tr>
<tr>
<td>1%(0-0)</td>
<td>10.604</td>
</tr>
<tr>
<td>50%(0-1)</td>
<td>10.171</td>
</tr>
<tr>
<td>80%(0-4)</td>
<td>11.595</td>
</tr>
<tr>
<td>90%(0-6)</td>
<td>12.387</td>
</tr>
<tr>
<td>95%(0-8)</td>
<td>12.705</td>
</tr>
<tr>
<td>97%(0-9)</td>
<td>12.793</td>
</tr>
<tr>
<td>98%(0-10)</td>
<td>12.850</td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th>Model</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%(0-50)</td>
<td>11.404</td>
</tr>
<tr>
<td>Bernoulli</td>
<td>4.060</td>
</tr>
<tr>
<td>Markov</td>
<td>9.140</td>
</tr>
<tr>
<td>1%(0-0)</td>
<td>9.140</td>
</tr>
<tr>
<td>50%(0-1)</td>
<td>8.689</td>
</tr>
<tr>
<td>80%(0-4)</td>
<td>10.113</td>
</tr>
<tr>
<td>90%(0-6)</td>
<td>10.905</td>
</tr>
<tr>
<td>95%(0-8)</td>
<td>11.224</td>
</tr>
<tr>
<td>97%(0-9)</td>
<td>11.311</td>
</tr>
<tr>
<td>98%(0-10)</td>
<td>11.368</td>
</tr>
</tbody>
</table>

Table III

concentrates less than 20% of the probability mass.

References


