ABSTRACT

We consider connecting networks in which we allow existing calls to be rearranged (rerouted) to accommodate a new call. This capability exists, for example, in the network of the recently proposed automatic main distribution frame [2,3,4] (where the need for switching live calls does not arise). For practical reasons, we are particularly interested in networks in which we allow a fixed small number, say \( t \), of calls to be rearranged. The probability that a (random) new call is still blocked even allowing \( t \)-arrangements of its own is defined as the \( t \)-trial-blocking probability. When \( t \) is unlimited, then the network is said to allow total rearranging. If a network can accommodate any new call under total rearranging, it is said to be rearrangeable.

In this paper, we propose a model for computing the \( t \)-trial-blocking probability recursively for the multi-stage Clos network. The computational complexity of this model is discussed and approximations suggested. We also give various modifications of this model to fit particular networks and possible extensions to more general networks. Finally, some simulation results and computer implementations of this model are mentioned.

1. INTRODUCTION

We consider connecting networks which allow existing calls to be rearranged (rerouted) to accommodate a new call. For practical reasons, we are particularly interested in networks which allow a fixed small number, say \( t \), of calls to be rearranged. The probability that a (random) new call is still blocked even allowing \( t \)-rearrangements is defined as the \( t \)-trial-blocking probability. When \( t \) is unlimited, then the network is said to allow total rearranging. If a network can accommodate any new call under total rearranging, it is said to be rearrangeable.

Since most of the existing results on the problem under discussion deal with a class of networks with simple structure, called Clos networks, we introduce them in the following.

A three-stage Clos network can be described by these properties:

1. There are three ordered stages. The \( i \)-th stage, \( i = 1,2,3 \), consists of \( r_1 \) copies of a rectangular switch \( V_1 \).
2. There is exactly one link between every pair \( (V_1, V_2) \) and every pair \( (V_2, V_3) \).
3. Each \( V_1 \) (called an input switch) is connected to \( r_1 \) input terminals of the network; each \( V_3 \) (called an output switch) is connected to \( r_3 \) output terminals of the network.

When the \( V_1 \) themselves can be networks, then the overall network is called a generalized Clos network.

The problem of computing blocking probabilities allowing rearrangements has been considered by Beneš [1], Ershov and Igashko [7,8], and Bergeron [2-4]. Beneš considered connecting networks where total rearranging is allowed and a standard traffic model is assumed. He gave explicit analytical formulas for the equilibrium probability of a given assignment of input terminals to output terminals. Blocking probabilities can then be computed by using the equilibrium probability. However, due to the large number of possible assignments, significant numerical computations are usually required.

Ershov and Igashko [8] considered rearrangeable three-stage Clos networks (with given traffic) where a fixed number of rearrangements is allowed. The rearrangements are to be done by first selecting a random pair of second-stage switches which can accommodate the new call if total rearranging is allowed, and then rearranging only the allowed number of calls. In order to facilitate their analysis, they also made the assumption that every second-stage switch carries the same load. Their results depend on knowing the sizes of many sets which have to be enumerated explicitly.

A special case of the generalized Clos network is the \((2s+1)\)-stage Clos network which can be derived recursively from a three-stage Clos network by specifying that \( V_1 \) and \( V_3 \) are rectangular switches, but \( V_2 \) is a \((2s-1)\)-stage Clos network. In general, these will be called multi-stage Clos networks.

Bergeron [2] considered a multi-stage Clos network (with given link-occupancies) where every stage has rectangular switches. He considered those paths which have only one busy link and computed the probability that the call using that link can be rearranged. The second-trial-blocking probability is the probability that none of these calls can be rearranged. He found that for a fixed small number, say \( t \), of calls to be rearranged even allowing \( t \)-rearrangements is defined as the \( t \)-trial-blocking probability. When \( t \) is unlimited, then the network is said to allow total rearranging. If a network can accommodate any new call under total rearranging, it is said to be rearrangeable.

Since most of the existing results on the problem under discussion deal with a class of networks with simple structure, called Clos networks, we introduce them in the following.

A three-stage Clos network can be described by these properties:

1. There are three ordered stages. The \( i \)-th stage, \( i = 1,2,3 \), consists of \( r_1 \) copies of a rectangular switch \( V_1 \).
2. There is exactly one link between every pair \( (V_1, V_2) \) and every pair \( (V_2, V_3) \).
3. Each \( V_1 \) (called an input switch) is connected to \( r_1 \) input terminals of the network; each \( V_3 \) (called an output switch) is connected to \( r_3 \) output terminals of the network.

When the \( V_1 \) themselves can be networks, then the overall network is called a generalized Clos network.

The problem of computing blocking probabilities allowing rearrangements has been considered by Beneš [1], Ershov and Igashko [7,8], and Bergeron [2-4]. Beneš considered connecting networks where total rearranging is allowed and a standard traffic model is assumed. He gave explicit analytical formulas for the equilibrium probability of a given assignment of input terminals to output terminals. Blocking probabilities can then be computed by using the equilibrium probability. However, due to the large number of possible assignments, significant numerical computations are usually required.

Ershov and Igashko [8] considered rearrangeable three-stage Clos networks (with given traffic) where a fixed number of rearrangements is allowed. The rearrangements are to be done by first selecting a random pair of second-stage switches which can accommodate the new call if total rearranging is allowed, and then rearranging only the allowed number of calls. In order to facilitate their analysis, they also made the assumption that every second-stage switch carries the same load. Their results depend on knowing the sizes of many sets which have to be enumerated explicitly.

A special case of the generalized Clos network is the \((2s+1)\)-stage Clos network which can be derived recursively from a three-stage Clos network by specifying that \( V_1 \) and \( V_3 \) are rectangular switches, but \( V_2 \) is a \((2s-1)\)-stage Clos network. In general, these will be called multi-stage Clos networks.

Bergeron [2] considered a multi-stage Clos network (with given link-occupancies) where every stage has rectangular switches. He considered those paths which have only one busy link and computed the probability that the call using that link can be rearranged. The second-trial-blocking probability is the probability that none of these calls can be rearranged. He found that for \( t \) large, for example, \( t = \binom{n}{k} \), rearranging a single call is almost sufficient to eliminate all blocking even if first-trial-blocking probability is significant, say \( \lesssim 0.1 \). However, many rough approximations were made in his analysis. In [4], Bergeron studied a five-stage network of this type. This network can also be viewed as a three-stage Clos network where \( V_2 \) itself is a three-stage Clos network (each \( V_2 \) is \( r_2 \times r_2 \) and there are \( r_3 \) of them). He suggested that if \( V_2 \) has small first-trial-blocking probability, then we can treat it as nonblocking and reduce the five-stage Clos network to a three-stage one. We will see in Section 6 that this seemingly harmless approximation may lead to a great discrepancy in calculating blocking probabilities.

In this paper, we propose a model to compute the \( t \)-trial-blocking probability recursively for the multi-stage Clos network with given link-occupancies. The computational complexity of this model is discussed and approximations suggested. We also give various modifications of this model to fit particular networks and possible extensions to more general networks. Some simulation results and computer implementations of this model are mentioned at the end.

2. THE CRÉSUS SYSTEM

We use the word system to mean the combined descriptions of the combinatorial structure of the network and the probabilistic assumptions of the traffic load. The basic network we consider is the multi-stage Clos network whose simple structure facilitates probability calculations.

Consider a \((2s+1)\)-stage Clos network (see Fig. 1 for an example). Define a linear graph between two switches \( V_{2i-1} \) and \( V_{2i} \) to be the union of all paths between the two switches. Then it is clear that such a linear graph depends only on \( i \) but not on which particular switches are chosen. Hence, we can denote the linear
We extend our definitions of "request" and "call" for the pair \((V_{s+1-l}, V_{s+1+i})\). A pair \((V_{s+1-l}, V_{s+1+i})\) seeking connection will be termed a request if each of the two switches in it has at least one idle outer link. The pair will be termed a reconnection if one \((V_{s+1-l}, V_{s+1+i})\) (the one the path through which we try to rearrange) is not allowed to connect the request, and a left (right) reconnection if one inner link of the involved \((V_{s+1-l}, V_{s+1+i})\) is not allowed to connect the request. Let \(BR_{t}(i), BR'_{t}(i), B_{t}(i), \) and \(B_{t}(i)\) denote the \(t\)th-trial-blocking probabilities for a request, a reconnection, a left reconnection, and a right reconnection. Furthermore, let the subgraph of \((G)\) which consists of \(a(1)-(1)\) and a pair of links \((V_{s+1-l}, V_{s+1+i})\) incident to it be called a bundle. Then due to the series-parallel structure of \((G)\) and the independence assumption, we can calculate the \(t\)th-trial-blocking probability for each bundle (denoted by \(P_{t}(x,y)\), where \(x,y\) denote the states, idle or occupied, of the \(V_{s+1-l}\) and the \(V_{s+1+i}\)), and then multiply them together.

Consider such a bundle with an idle \(V_{s+1-l}\) and an idle \(V_{s+1+i}\). Then clearly \(P_{t}(I,I) = BR_{t}(I-I)\).

Next consider a bundle with an idle \(V_{s+1-l}\) but an occupied \(V_{s+1+i}\). Then this bundle will be \(t\)th-trial nonblocking only if the \((G)\) can carry this call using exactly \(t\) rearrangements, \(t - 2 \geq 2\) and \(0\), and the blocking call can be rerouted somewhere else using at most \(t - 2 - t\), rearrangements (not counting the rearrangement of the blocking call).

Note that if the path of a rerouted call differs from its original path in the link \(L_{s+1-i}\), where \(i\) is the largest such \(i\), then its \(L_{s+1+i}\) link will also be different. In fact every link between stage \(s+1\) to stage \(s+1+i\) will be different. Such a rerouting is termed a \(21\)-link rearrangement [2]. Also note that a \(21\)-link rearrangement is exactly like routing a \((V_{s+1-l}, V_{s+1+i})\) request except that each link used in the original path can no longer carry the call.

Define \(\Delta BR_{t}(1)\) to be \(BR_{t}(1) - BR'_{t}(1)\), where \(BR_{t}(1)\) is \(0\) for any \(t < 1\). Then \(\Delta BR_{t}(1)\) is the probability that the request can be connected in exactly \(t + 1\) trials (t rearrangements). By noting that \(21\)-link rearrangements and \(2j\)-link rearrangements are disjoint for \(j \neq j'\), and that the \(V_{s+1+i}\) on the path of the blocking call has an idle terminal since there is a request involving that switch, we have

\[
P_{t}(I,0) = \prod_{j=1}^{s} \left(1 - \sum_{t=1}^{t_{j}} \Delta BR_{t}(1-I)\right)\]

where

\[
BR_{t-1,t}(j) = \begin{cases} BR_{t-1,t}(1) & \text{for } j = 1, \\
B_{t-1,t}(1) & \text{otherwise.}
\end{cases}
\]

\(P_{t}(0,1)\) can be similarly obtained by replacing \(BR_{t-1,t}(1)\) by \(B_{t-1,t}(1)\).

Finally, consider a bundle with both links occupied. When the network is not too small and traffic load not too light, then we can ignore the unlikely event that both busy links are used by the same call. As a good approximation, we assume that there are two blocking calls and we have to reroute both. By an argument similar to that of the previous case, we obtain

---

**Fig. 1. A Clos Network Linear Graph**

or by expanding \(G(1)\) so that each path between \(V_{s+1-l}\) and \(V_{s+1+i}\) has \(2i\) links. \(G(0)\) is just a single switch \(V_{s+1}\). Since the linear graph is of the series-parallel type, the method of Lee [13] applies readily to the calculation of blocking probabilities.

Let \(L_{s+1-i}\) denote any link between the \(i\)th stage and the \((i-1)\)th stage for \(1 \leq s\), or any link between the \((i-1)\)th stage and the \(i\)th stage for \(i \geq 2\). For a given \(V_{s+1-i}\), those links on the side closer to the center stage are called its inner links, whose number is denoted by \(m_{s+1-i}\). Those on the other side are called outer links, whose number is \(n_{s+1-i}\). (For \(V_{s+1+i}\), the distinction is irrelevant.) Note that \(m_{s+1-i} = m_{s+1+i}\) for \(i = 1, ..., s\).

We make the following assumptions on the traffic load of \(L_{s+1-i}\), \(i = 1, ..., 2s+1, i \neq s+1\).

(1) The sure-idle links (links which will be unoccupied with probability one). For \(m_{s+1-i} \geq n_{s+1-i}\), at most \(n_{s+1-i}\) of the \(m_{s+1-i}\) inner links of \(V_{s+1-i}\) can be occupied. Hence at least \(\max(0, m_{s+1-i} - n_{s+1-i})\) of the \(L_{s+1-i}\) and \(\max(0, n_{s+1-i} - m_{s+1-i})\) of the \(V_{s+1+i}\) in \((G)\) are sure-idle links. The remaining \(L_{s+1-i}\) and \(V_{s+1+i}\) are called regular links. The sure-idle links are assumed to be distributed randomly among the links of the switch. They are handled in this paper very much in the same manner employed by Krupp [1] in his work on networks with no reconnections.

(II) Each regular \(L_{s+i}\) is occupied independently with probability \(l_{s+i}\) and idle with probability \(1 - l_{s+i}\).

Note that these assumptions are approximately true if a random-routing strategy is employed, i.e., the choices of a route does not depend on the labeling of the links.

We will refer to this system as CRIBS where \(C\) stands for multi-stage Clos network, \(R\) and \(B\) stand for Random distribution of the sure-idle links, \(I\) and \(B\) stand for the Independent Binomial probability distribution of each regular \(L_{s+i}\) being occupied.

### 3. The Basic Formula for Blocking Probabilities of CRIBS

Define a request to be an idle pair of (input terminal, output terminal) seeking connection. A request becomes a call after it is connected. Since rectangular switches do not block, a request can be viewed as a pair of \((V_{s+1-l}, V_{s+1+i})\), with the understanding that each of the \(V_{s+1-i}\) has at least one idle terminal.

By our assumptions on the traffic load on the \(L_{s+i}\), the probability that a request is blocked depends solely on its linear graph. As the linear graphs are defined recursively, let us consider \(G(1), i > 0\), shown in Fig. 1.
where the appropriate quantities $B_r$, $B_l$, and $B_{r'}$ should substitute for $B_{r*}$.

We can now write $B_{r+}(i)$, $B_{l+}(i)$, $B_{r'}+$(i), and $B_{l+}(i)$ in terms of these conditional probabilities.

Suppose $G(i)$ has "a" sure-idle $L_{s+1-i}$, "b" sure-idle $L_{s+1+1}$ and $k$ available $G(i-1)$. Let $v$ be the number of $G(i-1)$ with a sure-idle $L_{s+1-i}$ and a sure-idle $L_{s+1+1}$ and a regular $L_{s+1+1}$, (b-v) $G(i-1)$ with a sure-idle $L_{s+1-i}$ and a regular $L_{s+1+1}$ and a sure-idle $L_{s+1+1}$ and (k-a+b-v) $G(i-1)$ with a regular $L_{s+1-i}$ and a sure-idle $L_{s+1+1}$. Let $S-R$ denote a $G(i-1)$ with a sure-idle $L_{s+1-i}$ and a regular $L_{s+1+1}$ and and similar definitions for $S-S$, $R-S$, and $R-R$. Then the $t$th-trial-blocking probability for a bundle will be

$$P_t(i, I), \quad \text{if } (G(i-1) \text{ is } S-S),$$

$$(1-\lambda_{s+1-i})P_t(I, I) + \lambda_{s+1-i}P_t(0, I), \quad \text{if } R-S,$$

$$(1-\lambda_{s+1-i})P_t(I, I) + \lambda_{s+1-i}P_t(I, 0), \quad \text{if } S-R,$$

$$(1-\lambda_{s+1-i})(1-\lambda_{s+1+1})P_t(I, I) + (1-\lambda_{s+1+1})\lambda_{s+1+1}P_t(I, 0)$$

$$+ \lambda_{s+1-i}(1-\lambda_{s+1+1})P_t(0, I) + \lambda_{s+1-i}\lambda_{s+1+1}P_t(O, O) \quad \text{if } R-R.$$

Furthermore the probability of having $v$ S-S's is

$$\frac{(k-a)}{b-v} \left(\frac{a}{b}\right)^v \left(\frac{k}{b}\right).$$

Combining all these, then the $t$th-trial-blocking probability for the $G(i)$ is

$$B_t(i; a, b, k) = \sum_{v=a+b-k}^{\text{range of } v} \left(\frac{a}{b-v}\right)^v \left(\frac{k}{b}\right)$$

where $\binom{n}{y}$ is 0 for $y < 0$. By setting $a = \max(0, 1-s_{i-1}+t_{i-1}+1)$, $b = \max(0, m_{s+1-i}+t_{s+1-i}+1)$, $k = m_{s+1-i}$, we obtain $BR_t(r)$. $BR_t(r)$, $BR_t(l)$, and $BR_t(l)$ can be obtained from $BR_t(r)$ by reducing $a$, $b$, and $k$ by 1 when necessary.

Finally, since $v_{r+}$ is nonblocking, $BR_t(r) = BR_t(l) = 0$ for all $t$.

4. COMPUTATIONS AND APPROXIMATIONS OF THE BASIC FORMULA

For $t = 1$, then $P_1(I, O) = P_1(0, I) = P_1(O, O) = 1$, and $P_1(I, I) = BR_1(1-i)$. Thus

$$B_1(i; a, b, k) = \sum_{v=a+b-k}^{\text{range of } v} \left(\frac{a}{b-v}\right)^v \left(\frac{k}{b}\right)$$

$$\left(1-\lambda_{s+1-i}\right)BR_1(1-i)\lambda_{s+1-i}^{b-v}$$

and

$$B_2(O; a, b, k) = 0.$$

This result has been obtained by Krupp [11] in dealing with networks allowing no rearrangement. The recursion equation can be solved in time proportional to $\sum_{h=2}^{\text{range of } v_h} v_h$ where $v_h$ is the $v$ in $B_1(i; a, b, k) = 0$.

This is of course the famous Clos theorem [5] for nonblocking networks.

For $t = 2$, then $P_2(I, 0) = BR_2(1-i)$, and $P_2(0, I) = \prod_{j=1}^{s} \left(1 - 1 - BR_1(1-i)\right) \left(1 - BR_1(j)\right)$, and $P_2(0, 0) = 1$, and $P_2(0, I) = \prod_{j=1}^{s} \left(1 - 1 - BR_2(1-i)\right) \left(1 - BR_2(j)\right)$, and $P_2(0, 0) = 1$, and $P_2(0, 0) = 1$, and $P_2(0, 0) = 1$, and $P_2(0, 0) = 1$, and $P_2(0, 0) = 1$, and $P_2(0, 0) = 1$.
The basic formula may have to be modified in various ways to suit individual networks with their own special features. Here we just mention a few possibilities which are particularly relevant to the AMDF networks discussed in [2], [3], and [4].

(i) Since there is a risk of disconnection, however slight, in actually rerouting a call, there are terminals whose calls should not be candidates for rearranging. We estimate the number of such terminals and define a parameter $\theta$ to be the probability that a random terminal is one of these. Then whenever we use the probability that a call can be re-routed in the formula, we shall multiply it by $\theta$.

(ii) In [3], it is reported that occupancy of adjacent links by different types of calls can cause excessive interference. Define a neighborhood of a link to be those adjacent links which are not allowed to carry incompatible calls. Then it is suggested in [3], whenever the probability of a link being idle is used, we should multiply it by the probability that there are no incompatible calls in its neighborhood. Note that the latter probability varies according to which type of call the idle link is expected to carry, since each type has different frequency. This approach can be readily adopted for our model.

(iii) Sometimes one is not so much concerned with a fixed number of rearrangements, but rather with a restricted class of rearrangements. For example, one such class (suggested by J. Dunn [6]) for application to the AMDF network) includes all rearrangements which rearrange the two calls occupying the two links connecting to the same terminal, and define $\delta$ the risk of disconnection, however slight, in actually rerouting a call. Note that the latter probability varies according to which type of call the idle link is expected to carry, since each type has different frequency. This approach can be readily adopted for our model.

The basic formula may also be extended to cover the following more general networks.

(i) A multi-stage connecting network has the same structure as a multi-stage Clos network, i.e., sequential stages, identical switches in a stage, links existing only between adjacent stages, except that the links do not have to be of the pattern as prescribed for the Clos network and the number of stages $s$ does not have to be odd. If the linear graph of $(v_1, v_{s+1})$ in a multi-stage connecting network depends only on $i$ and is of the series-parallel type, its trial-blocking probability can be computed similarly as in this paper (under the same kind of traffic-load assumptions). However, since the set of $l_i (l_{s+1})$ in $G(i)$ is only a subset of the $m_i$ inner links of $v_i (v_{s+1})$, it is difficult to assign the numbers of sure-idle links. One way to handle this is to treat every link in $G(i)$ as regular. Another way is to compute blocking probabilities for every possible value of the number of sure-idle links and then take an average.

(ii) In some networks [6], terminals are partitioned into groups where two terminals of the same group have more connecting paths between them than two terminals of different groups. Here we have two different linear graphs for two types

$$B_i(1;i_1, a, b, k)$$

can be approximated by the first term in the sum, i.e., the term corresponding to $v = 0$. The computation time then becomes proportional to $t \sum_{h=2}^{t} (s-h)^2$.

When the two approximations are used together, the computation time becomes proportional to $t$.
of requests. We can also consider cases where there are more than two types. We compute blocking probabilities for each type and then take the average probabilities combined over the various types. When we rearrange a call, we use the average blocking probability, or average conditional on any information we have about the call to be rearranged, in the formula.

(iii) A network is called a one-sided network (see [10]) if two terminals on the same side (both input or both output) can also generate a request. It is called input-mixed or output-mixed if every request must contain at least one terminal of the designated type. If we can assume that the linear graphs of all requests can be partitioned into a few types, say input-type, input-output type, output-output type, then a situation mathematically similar to case (ii) arises.

6. IMPLEMENTATION AND SIMULATION

Some simulation results have been obtained by W. F. Hoyt [9] for a five-stage Clos network where each stage has two rectangular witches. \(v_1\) and \(v_6\) each has \(n\) outer links and \(r\) inner links, \(n \leq r\), \(v_2\), \(v_3\) and \(v_4\) each has \(r\) outer links and \(r\) inner links. It is assumed that \(\lambda_1 = \lambda_2 = \lambda\), and \(\lambda_5 = \frac{R}{\lambda}\). A summary of the results is given in the following table.

<table>
<thead>
<tr>
<th>(r)</th>
<th>(n)</th>
<th>(\lambda)</th>
<th>(1^{st})</th>
<th>(2^{nd})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4)</td>
<td>(4)</td>
<td>(0.8)</td>
<td>(0.44)</td>
<td>(0.67)</td>
</tr>
<tr>
<td>(16)</td>
<td>(15)</td>
<td>(0.9)</td>
<td>(0.44)</td>
<td>(0.67)</td>
</tr>
<tr>
<td>(64)</td>
<td>(60)</td>
<td>(1.2)</td>
<td>(0.22)</td>
<td>not done</td>
</tr>
</tbody>
</table>

Table 1

Model 1 is the simple model proposed in [3] which assumes that blocking occurs only due to the nonavailability of \(L_1\) and \(L_2\), while the three inner stages (G(1)) can not block. Model 2 is the model proposed in this paper.

The above data suggests two surprising conclusions:

(i) The closeness of the probability values between Model 2 and the simulation for \(n = 4\) and \(n = 16\) lends evidence to Bergeron's suggestion [2] that rearranging a single call is almost sufficient to eliminate blocking in a large network, even if \(1^{st}\)-trial-blocking probability is nontrivial.

(ii) Model 2 seems to result in a significant improvement over Model 1. Their differences diverge increasingly as \(r\) gets large.

The unavailability of simulating results for \(r = 64\) at the present stage is due to the large cost involved in enumerating all possible pairs of idle terminals and the huge number of possible rearrangements which must be explored to determine the blocking situation. Work is presently underway to overcome these difficulties.

A computer program to calculate the \(1^{st}\)-trial-blocking probability and the \(2^{nd}\)-trial-blocking probability has been written by C. T. Chen and J. Dunn.

The author wishes to acknowledge the many helpful suggestions and critical comments on this work provided by V. E. Beneš, R. F. Bergeron, J. C. Dunn, and R. L. Graham.

REFERENCES

1. Beneš, V. E., "Traffic in connecting networks when existing calls are rearranged", Bell System Tech. J.
ERRATA

Paper No 543 - F.K. HWANG

Blocking Probabilities for Connecting Networks Allowing Rearrangements

<table>
<thead>
<tr>
<th>P 1, Col 1, line 9 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>v_1</td>
<td>v_1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P 2, Col 2, line 13 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{j=1}^{s} )</td>
<td>( \sum_{j=i}^{s} )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P 3, Col 1, line 8 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_r (i;a,b,k) )</td>
<td>( B_t (i;a,b,k) )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P 3, Col 1, line 7 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \left( \prod_{i=1}^{V} P_t (I,I) \right)^V )</td>
<td>( \left( \prod_{i=1}^{V} P_t (I,I) \right)^V )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P 3, Col 2, line 11 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>in ( B_1 (i;a,b,k) = 0 )</td>
<td>in ( B_1 (h;a,b,k) ),</td>
<td></td>
</tr>
</tbody>
</table>

Note that \( B_1 (i;a,b,k) = 0 \)

<table>
<thead>
<tr>
<th>P 4, Col 1, line 22 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t \sum_{h=2}^{i} )</td>
<td>( t \sum_{i=2}^{s} \sum_{h=2}^{i} )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P 4, Col 1, line 7 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t \sum_{h=2}^{i} )</td>
<td>( t \sum_{i=2}^{s} \sum_{h=2}^{i} )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P 4, Col 1, line 6 from bottom</th>
<th>Existing</th>
<th>Should be</th>
</tr>
</thead>
<tbody>
<tr>
<td>to the end of the paragraph)</td>
<td>Strike out</td>
<td></td>
</tr>
</tbody>
</table>