Modeling regular, process-structured networks*

by BRUCE W. ARDEN and HIKYU LEE
Princeton University
Princeton, New Jersey

INTRODUCTION

An interesting strategy to exploit micro-technology is to interconnect microcomputers (i.e., microprocessor with local memory) in a regular dataflow network of low degree. The network is regular so that each computer is a similar "building block" with regard to the number of connecting data buses. The number of buses is small not only because microcomputers are inherently limited in their bus capacity but also because many incident buses lead to switching and "memory port" complexity, which is difficult to handle by micro-circuits. Such complexity is one of the reasons why shared memory "mainframe" computers are relatively expensive. In essence, the low-degree, regular network approach replaces the hard-wired switching with programmed message-passing. Since the processor node will not be fully utilized by productive computing, some of the capacity can profitably be used for such message-handling.

This paper is concerned with an illustrative comparison of performance of two such data flow networks. The first is not regular and it models a particular process structure in which all messages are passed to the intended destination within a single step. For convenience, we call this network a process network. In the second, the same process structure is mapped onto a regular network (of degree 3) with the result that a number of multi-step message paths are introduced. The comparison shows the effects on system utilization and response time of the additional message handling overhead introduced in the regular network.

PROCESS STRUCTURE

Current multiprogrammed systems are usually described as process-structured. The processes that are enabled for execution are assigned system resources on the basis of some scheduling procedure. An alternative architecture, which arises from the development and low cost of microcomputers, is to assign each system and user process to a microcomputer (which may have attached I/O devices) for the entire life of the process. The use of a process becomes a matter of sending data to the appropriate process node, and ultimately, receiving the response. This is a data flow network in the sense that the messages enable the resident processes to carry out their specific computation. It is not unreasonable to think about networks of hundreds of microcomputers but, of course, the geometry of a regular network and the mapping of the process structure onto the regular network become important considerations.

In a system of processes, some are autonomous and some are subordinate. That is, some spontaneously generate a message and others are passive in the sense that they only react to the receipt of a message and are otherwise idle. Clearly, user processes are in the first category and most, but not all, of system processes are in the second. For example, system processes such as archiving, and spooling routines can act autonomously.

Each autonomous process and the subordinate processes with which it communicates and for which it provides a workload comprise a chain with one circulating message. It is assumed that chains, even if they are indistinguishable with respect to their workload demands, cannot be combined in the process system, since they cannot be correspondingly combined in the regular system due to the distinct locations of each autonomous processes resulting from the 1-1 mapping of the process structure onto the regular network. Hence, the number of chains is equal to the number of autonomous processes and each chain contains one circulating message. This situation is quite different from the models of conventional, tightly-coupled systems where the number of processes is generally larger than the number of processors.

In the language of queueing theory, the circulating entities are usually called customers and a specific customer class is identified by the distinguishing workload it presents at each service center. To preserve the data flow orientation, this concept is here called message class. Circulating messages are processed at each node in the process system. In the regular system, they are, in addition, simply retransmitted at some nodes without being processed locally.

UNDERLYING MODEL FOR PROCESS SYSTEM

As the preceding suggests, the underlying model for the process system is a closed network of queues model with multiple chains and different classes of messages. There is

---

* This work has been supported by NSF Grant DCR 74-18655.
a finite number of processors indexed 1, 2, ..., $M$ and a finite number of circulating messages numbered 1, 2, ..., $R$, where $R$ is equal to the number of autonomous processes in the system. Obviously, we have $N < M$. Since each autonomous process defines a chain, we assume that there is a finite number of different classes of messages 1, 2, ..., $R$, where $R$ is equal to the number of autonomous processes, i.e., $R = N$. Let $N_i$ denote the number of messages in chain $i$ (i = 1, 2, ..., $R$), which is equal to one for each chain $i$. Assuming that the messages in chain $i$ are of class $i$, $N_i$ also denotes the number of class $i$ messages in the system.

The behavior of circulating messages is determined by its service time distribution at each service center and transition pattern through the network. The service time distribution of messages is assumed to be general with rational Laplace transformation and the queueing discipline at each service center is assumed to be LCFS-PR (Last Come First Served-Preemptive Resume). The transition of circulating messages is described by the transition probability matrix $Q = \{ q_{r\beta} \}$, where $q_{r\beta}$ denotes the probability that a class $r$ message completing its service at service center $i$ will require its next service at service center $j$ in message class $\beta$. Since class $r$ messages are confined in chain $r$, we have $q_{r\beta} = 0$ if $r \neq \beta$ ($r = 1, 2, ..., R$).

We define the state of the model for the process system as the number of messages in each class at each service center. Thus, the state $S$ is written as

$$ S = (y_1, y_2, \ldots, y_M), $$

where

$$ y_i = (n_{i1}, n_{i2}, \ldots, n_{iM}). $$

Then a feasible state of the model satisfies

$$ \hat{N} = (N_1, N_2, \ldots, N_R) = \sum_{i=1}^{M} y_i, $$

where

$$ N_j = \sum_{i=1}^{M} n_{ij} = 1 \quad (j = 1, 2, \ldots, M) $$

and

$$ N = \sum_{i=1}^{R} N_i = \text{constant}. $$

The $n_{ir}$ denotes the number of class $r$ messages at service center $i$.

**UNDERLYING MODEL FOR REGULAR SYSTEM**

The regular system is obtained from the process system by mapping the process structure onto a regular network. Hence, as in the process system, there are $M$ processors and $N$ circulating messages in the system. We assume that a specific network geometry and the mapping for the regular system are given.

Since the adjacency relationship between processes in the process system is not, in general, preserved due to the mapping, transition of class $r$ messages from service center $i$ to an adjacent service center $j$ in the process system might take a multi-step path in the corresponding regular system. We add $R$ new message classes $1, 2, \ldots, R$ to our model for the regular system to distinguish messages being retransmitted, i.e., if a class $r$ message at service center $i$ makes a multi-step transition to service center $j$ for the next service request, then we assume that there exists class $r$ message arrival at each intermediate service centers along the path. Hence, each service center in the regular system is presumed to execute a local, system or user process (which may involve the support of a connected I/O device) as well as simple message-passing task for inter-node communications.

We assume that the shortest path between service center $i$ and service center $j$ in the regular system is used for the inter-node communication. Furthermore, if there exists more than one such path, then each shortest path is used with equal probability. For convenience, we call this routing scheme as SPEP (Shortest Path with Equal Probability) routing scheme.

The service time distribution of messages is assumed to be general with rational Laplace transformation and the queueing discipline at each service center is assumed to be LCFS-PR (Last Come First Served-Preemptive Resume).

The transition probability matrix $P = \{ p_{r\beta} \}$ ($\alpha = 1, 2, \ldots, R; \beta = 1, 2, \ldots, R$) describes transition of messages and can be derived for the regular system from the transition probabilities given in the process system as follows. Let $\lambda_{i\alpha}$ denote the relative rate of class $\alpha$ message arrival at node $j$ from node $i$ and class $\alpha$. $\lambda_{i\alpha}$, corresponds to the average number of times a class $\alpha$ message at node $i$ visits link $(i, j)$ to become a class $\beta$ message at node $j$. Let $e_{ir}$ denote the relative arrival rate of class $r$ messages at service center $i$ in the process system, where $i = 1, 2, \ldots, M$ and $r = 1, 2, \ldots, R$. The $e_{ir}$ are obtained up to multiplicative constant from the following set of linear equations:

$$ e_{ir} = \sum_{\beta=1}^{R} q_{r\beta} e_{i\beta}, $$

where $j = 1, 2, \ldots, M$ and $s = 1, 2, \ldots, R$. Then, for each node pair $(x, y)$ in the process system such that $e_{xy} > 0$ and $q_{rxy} > 0$ for some $r = 1, 2, \ldots, R$, the total relative arrival of messages at node $y$ from node $x$ in the regular system is $e_{xy} q_{rxy}$ and the fraction which passes through an edge $(i, j)$ in the set of edges of the shortest paths from node $x$ to node $y$ is given by

$$ e_{xy} q_{rxy} P(x, i; x, y) P(i, j; x, y), $$

where $P(x, i; x, y)$ denotes the probability that a message, which has completed a local service at node $x$ and is making a multi-step transition to node $y$ for the next local service request, will pass through node $i$ and $P(i, j; x, y)$ denotes the probability that the message which is passing through node $i$ will make a transition to an adjacent node $j$ under SPEP routing scheme. Note that $P(x, i; x, y)$ and $P(i, j; x, y)$.
y) are given by
\[ P(x, i; x, y) = \text{# of shortest paths from } n_x \text{ to } n_y \text{ which include } n_i \]
and
\[ P(i, j; x, y) = \text{# of shortest paths from } n_j \text{ to } n_y \text{ which include } n_i \]

For the special case of fixed routing, where a specific single path is used for the inter-node communication between node x and node y, we have
\[ P(x, i; x, y) = P(i, j; x, y) \]
if \( n_i \) and \( n_j \) are on the path.

Considering all the node pairs \( (x, y) \) in the process system such that \( e_{rr} > 0 \) and \( q_{rr} > 0 \) for \( r = 1, 2, \ldots, R \), we get \( \lambda_{io}(i, j, 1, 2, \ldots, M) \) for the regular system. Then, the transition probability matrix \( P = \{p_{ij}^{\alpha, \beta}\} \) is obtained from:
\[ P_{\alpha, \beta} = \lambda_{io}(i, j, 1, 2, \ldots, M) / \sum_{k=1}^{M} (\lambda_{io} + \lambda_{io}) \]
where \( \alpha, \beta = r \) or \( \tilde{r} \), and \( r = 1, 2, \ldots, R \).

The state of the model for the regular system is defined as
\[ S = (y_1, y_2, \ldots, y_M) \]
where \( y_i = (n_{i1}, n_{i2}, \ldots, n_{iR}, n_{i1}, n_{i2}, \ldots, n_{iR}) \). Then the feasible state of the model satisfies
\[ \tilde{N} = (N_1, N_2, \ldots, N_R) = \sum_{i=1}^{M} \tilde{y}_i \]
where
\[ N_j = N_{j1} + N_{j2} = \sum_{i=1}^{M} (n_{ij1} + n_{ij2}) = 1 \quad (j = 1, 2, \ldots, R), \]
\[ N = \sum_{i=1}^{R} N_i = \text{constant} \]
and
\[ \tilde{y}_i = (n_{i1}, n_{i2}, n_{i3}, n_{i4}, \ldots, n_{iR} + n_{iR}) \].

**Performance Measures**

The underlying models for both process and regular system are, in essence, special cases of the general queueing network model developed by Baskett et al. where the equilibrium state probabilities are shown to have the following product form:
\[ P(S) = C(\tilde{N}) \prod_{i=1}^{M} f_i(y_i) \]
where
\[ f_i(y_i) = \begin{cases} n_i! \prod_{j=1}^{R} \left( \frac{e_{ij}}{\mu_{ij}} \right)^{n_{ij}} \left( \frac{1}{\mu_{ij}} \right)^{n_{ij}} & \text{for process system} \\ n_i! \prod_{j=1}^{R} \left( \frac{\lambda_{ij}}{\mu_{ij}} \right)^{n_{ij}} \left( \frac{1}{\mu_{ij}} \right)^{n_{ij}} & \text{for regular system} \end{cases} \]

The normalization constant \( C(\tilde{N}) \) is obtained by equating the sum of these products, over all states, to unity, i.e.,
\[ C(\tilde{N}) = \sum_{\text{all feasible states}} \prod_{i=1}^{M} f_i(y_i) \]

The direct approach to compute the normalization constant yields exponential growth in the number of algebraic operations. However, efficient computational techniques have been developed by several authors, which are, in essence, generalizations of Buzen's result.

Let \( P_i(y_i) \) denote the marginal probability that the service center \( i \) is in state \( y_i \) where \( y_i = (n_{i1}, \ldots, n_{iR}) \) for the process system and \( (n_{i1}, \ldots, n_{iR}) \) for the regular system. Then, we have
\[ P_i(y_i) = \sum_{\text{all states } S_i \text{ such that } S_i \text{ contains } i} P(S) \]

The mean number of class \( r \) messages at service center \( i \), \( E(n_m) \) is given by
\[ E(n_m) = \sum_{k=1}^{M} \{ \sum_{\text{all states } S_k \text{ such that } i \in S_k} P(S) \} k \]
where \( \theta = 1, 2, \ldots, R \) for the process system and \( 1, 2, \ldots, R, 1, 2, \ldots, R \) for the regular system.

The utilization of service center \( i \) by class \( r \) messages, \( \rho_{i\theta} \) is obtained from:
\[ \rho_{i\theta} = \sum_{S_i} P_i(y_i) \frac{n_m}{n_i} \]

We define the mean response time of class \( r \) messages, \( T_r \) as the time for the class \( r \) message leaving an autonomous node in chain \( r \) after finishing its local service request to revisit the autonomous node in class \( r \) after finishing its service requirements in the rest of the system. Assuming that node \( i \) is the autonomous node in
chain \( r \), the total number of the messages in the rest of the system is \( N_r - E(n_{ir}) \). Applying Little's result to the rest of the system, we have

\[
T_r = \frac{N_r - E(n_{ir})}{v_{ir}}
\]

where \( v_{ir} \) is the mean departure rate of class \( r \) messages at service center \( i \).

EXAMPLE

A 10-node example of process system with necessary parameter values is shown in Figure 1. There are three autonomous nodes, 1, 2 and 3, each of which comprises a chain.

This example, though not based on an actual system, is intended to represent a simple hierarchical operating system structure with different types of task requirements. Chain 1 represents compute-bound batch processing. Node 1 reads a program either in card image format via spool controller or from disk file via file system. Similarly, it produces output to spool controller or for storage in the disk file. Chain 2 and Chain 3 both represent interactive processing of high-level file operations, where Node 2 and Node 3 represent user terminals. It is assumed that Chain 2 represents rather long file operation compared to Chain 3.

The regular system, onto which the process structure in Figure 1 is mapped, is shown in Figure 2. Transition prob-

From the collection of the Computer History Museum (www.computerhistory.org)
abilities obtained by the method explained in the preceding section are also shown for Chain 2 as an illustration. In the network shown, there is a unique shortest path for each pair of nodes. This particular geometry is known as a (3,2) Moore graph, which is also called a Peterson graph. In this case, the mapping is intentionally non-optimal so that appreciable message-passing overhead is introduced. In an optimal assignment, the process network adjacency would be preserved to the greatest extent possible. The mean service time of messages in Classes 1, 2 and 3 is assumed to be equal at each service center and is given by:

\[
\frac{1}{\mu_i} = \frac{1}{\mu_a} = .001 \gamma \text{ sec} \quad \text{and} \quad \frac{1}{\mu_i} = .002 \gamma \text{ sec},
\]

where \( i = 1, 2, \ldots, 10 \) and \( \gamma \) is a multiplication factor. The statistics obtained on performance measures for the process system is shown in Table I. Table II shows corresponding statistics for the regular system with \( \gamma = 1 \). Comparison of the result shows the effects of message-passing overhead on service center utilizations and mean response time of the system. For example, the productive utilization of Service Center 1 is decreased by \( \approx 7 \) percent, while overall utilization is increased by \( \approx 1.5 \) percent, and the mean response time is increased by \( \approx 12 \) to \( \approx 25 \) percent for each chain due to the message-passing overhead.

Figures 3 and 4 show the effects of changing the mean service time for the messages in class 1, 2 and 3 on service center utilization and mean service time of the system.

\[
\frac{1}{\mu_{11}} = \frac{1}{\mu_{13}} = .001 \gamma \text{ sec} \quad \text{and} \quad \frac{1}{\mu_{12}} = .002 \gamma \text{ sec} \quad i = 1, 2, \ldots, 10
\]
### Table I—Model Statistics of Process System

#### a) Overall statistics

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>.389</td>
<td>.814</td>
<td>.718</td>
<td>.099</td>
<td>.062</td>
<td>.173</td>
<td>.156</td>
<td>.156</td>
<td>.104</td>
<td>.254</td>
</tr>
<tr>
<td>$E(n_i)$</td>
<td>.389</td>
<td>.814</td>
<td>.718</td>
<td>.102</td>
<td>.062</td>
<td>.193</td>
<td>.156</td>
<td>.156</td>
<td>.111</td>
<td>.300</td>
</tr>
</tbody>
</table>

#### b) Detailed statistics

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>.389</td>
<td>0</td>
<td>0</td>
<td>.078</td>
<td>.062</td>
<td>.049</td>
<td>.156</td>
<td>.156</td>
<td>.029</td>
<td>.058</td>
</tr>
<tr>
<td>2.</td>
<td>0</td>
<td>.814</td>
<td>0</td>
<td>.007</td>
<td>0</td>
<td>.039</td>
<td>0</td>
<td>0</td>
<td>.023</td>
<td>.093</td>
</tr>
<tr>
<td>3.</td>
<td>0</td>
<td>0</td>
<td>.718</td>
<td>.014</td>
<td>0</td>
<td>.085</td>
<td>0</td>
<td>0</td>
<td>.051</td>
<td>.103</td>
</tr>
</tbody>
</table>

| class |     |     |     |     |     |     |     |     |     |     |
| 1.   | .389 | 0   | 0   | .080 | .062 | .016 | 0   | 0   | .044 | 0   |
| 2.   | 0   | .814 | 0   | .007 | 0   | .044 | 0   | 0   | .025 | .109 |
| 3.   | 0   | 0   | .718 | .016 | 0   | .094 | 0   | 0   | .054 | .119 |

#### c) Mean response time (sec)

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.031</td>
<td>1.143</td>
<td>0.787</td>
</tr>
</tbody>
</table>

### Table II—Model Statistics of Regular System

#### a) Overall statistics

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_i$</td>
<td>.395</td>
<td>.778</td>
<td>.704</td>
<td>.092</td>
<td>.059</td>
<td>.163</td>
<td>.169</td>
<td>.145</td>
<td>.098</td>
<td>.269</td>
</tr>
<tr>
<td>$E(n_i)$</td>
<td>.414</td>
<td>.778</td>
<td>.725</td>
<td>.096</td>
<td>.059</td>
<td>.181</td>
<td>.175</td>
<td>.145</td>
<td>.104</td>
<td>.323</td>
</tr>
</tbody>
</table>

#### b) Detailed statistics

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>.362</td>
<td>0</td>
<td>0</td>
<td>.072</td>
<td>.058</td>
<td>.045</td>
<td>.145</td>
<td>.145</td>
<td>.027</td>
<td>.054</td>
</tr>
<tr>
<td>2.</td>
<td>0</td>
<td>.778</td>
<td>0</td>
<td>.006</td>
<td>0</td>
<td>.037</td>
<td>0</td>
<td>0</td>
<td>.022</td>
<td>.089</td>
</tr>
<tr>
<td>3.</td>
<td>0</td>
<td>0</td>
<td>.680</td>
<td>.014</td>
<td>0</td>
<td>.081</td>
<td>0</td>
<td>0</td>
<td>.049</td>
<td>.097</td>
</tr>
</tbody>
</table>

| class |     |     |     |     |     |     |     |     |     |     |
| 1.   | .007 | 0   | .005 | 0   | 0   | 0   | .005 | 0   | 0   | .029 |
| 2.   | .012 | 0   | .009 | 0   | 0   | 0   | .009 | 0   | 0   | 0   |
| 3.   | .013 | 0   | .010 | 0   | .001 | 0   | .010 | 0   | 0   | 0   |

| class |     |     |     |     |     |     |     |     |     |     |
| 1.   | .372 | 0   | 0   | .074 | .058 | .051 | .148 | .145 | .029 | .066 |
| 2.   | 0   | .778 | 0   | .007 | 0   | .042 | 0   | 0   | .024 | .107 |
| 3.   | 0   | 0   | .690 | .015 | 0   | .088 | 0   | 0   | .051 | .116 |

| class |     |     |     |     |     |     |     |     |     |     |
| 1.   | .007 | 0   | .009 | 0   | 0   | 0   | .006 | 0   | 0   | .035 |
| 2.   | .017 | 0   | .015 | 0   | 0   | 0   | .010 | 0   | 0   | 0   |
| 3.   | .018 | 0   | .010 | 0   | .001 | 0   | .011 | 0   | 0   | 0   |

#### c) Mean response time (sec)

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.035</td>
<td>1.433</td>
<td>0.911</td>
</tr>
</tbody>
</table>
1.0 OVERALL UTILIZATION

0.8
0.6
0.4
0.2
0.0

PRODUCIVE UTILIZATION

MESSAGE PASSING UTILIZATION

Figure 3—Service center utilizations vs. \( \gamma \).

CONCLUSION

A network of queues model for process-structured systems is discussed and analyzed by using solution techniques recently developed. The comparison of process system and regular system shows the effects on system performance measures of the message-passing overhead introduced when the process system is mapped onto the regular system.

Although the issues concerning the mapping of a process system to a regular system and specific geometry for a regular network of low degree are not covered in this paper, they are vital considerations in considering the effects on system performance of the message-passing overhead.

The analysis technique presented could be used to design networks to meet specific performance requirements. Given a particular geometry for the regular system, for instance, the analysis technique can be used as a tool to determine a mapping which gives preference to a certain set of autonomous processes in terms of the mean response time or other performance measures.

REFERENCES

3. Baskett, F., K. Chandy, R. Muntz and F. Palacios, "Open, Closed and


