ABSTRACT -- This paper extends the modeling and analysis of packet switched multistage cube interconnection networks to include the use of multiple-packet message formats. A multiple-packet message format is needed within an interconnection network when the message to be transmitted exceeds the network's packet size. The packet offset time is introduced and used to gauge the transmission rate differential between the network interchange boxes and the system PEs. Based on assumed network and system operating characteristics, optimum network performance offset values are given. A methodology for evaluating the relative performance of circuit and packet switching transmission modes is developed. For an assumed set of network and system parameters, results indicate that, in contrast to inter-computer networks, the performance of packet switching networks is better than that of circuit switching networks for long message lengths.

1. INTRODUCTION

An integral part of any parallel processing system is its interconnection network, used to link the processors and, possibly, memory modules together. One class of networks suitable for use in a parallel processing system is the multistage cube networks [24]. This class includes topologically equivalent networks such as the baseline [26], the indirect binary n-cube [18], the generalized cube [22], the omega [14], the flip [1], and the SW-banyan (S-F=2) [11]. Multistage cube networks have been used in the STARAN [1] and BBN Butterfly [4] and have been proposed for use in many future systems such as: the IBM RP3 [19], PASM [23], Ultracomputer [10], and database machines [7].

Multistage cube networks can be designed to function using circuit or packet switching methodologies for data transmission within the networks. A general framework is developed in this paper that quantizes system and network parameters and thereby facilitates the comparison of circuit and packet switched networks. A specific set of network and system parameters are used to demonstrate the utility of the framework. The general framework can be used by network designers to select which network switching methodology best meets their system requirements. Networks can also be designed that incorporate both circuit and packet switching modes [20]. In networks where both modes are available, this analysis framework can be used to identify which transmission mode should be used for a given set of operating conditions.

In a packet switched network, when the amount of data to be transmitted exceeds the network's packet size, multiple data packets must be generated by the network source and then transmitted to the appropriate destination. The performance analysis of networks operating in a multiple-packet environment is considered in this paper. Packet switching network models are extended, through the use of simulations, to include the transmission of multiple-packet messages. A tradeoff analysis, based on assumed network and system operating characteristics, is made to compare the performance of this format to that of circuit switching. This research has been motivated by the design of the PASM parallel processing system at Purdue University [23, 16].

Section 2 defines the system and network operating environments. The performance analysis of multiple-packet message transmissions is then presented in Section 3. Performance comparisons between circuit and packet switching network operations is discussed in Section 4. Conclusions are presented in Section 5.

2. NETWORK MODEL AND SYSTEM OPERATION

In this paper, the generalized cube network will be used as a model of all of the networks within the multistage cube class. The network model will be the same as the models developed in previous analyses for omega networks [2, 3] and baseline networks [15]. Assume the network has N inputs (sources) and N outputs (destinations), where N = 2^a. The network consists of n stages with each stage being composed of N/2 2-by-2 interchange boxes. Interchange boxes in stage i pair I/O lines with labels that differ only in the i-th bit position. The same labeling is used for both the input and the output lines connected to an interchange box. A generalized cube network is shown in Figure 1, with N=8. The network's sources and destinations are assumed to be processing elements (PEs), processor-memory pairs; i.e., PE i is connected to network input port i and network output port i, 0 ≤ i < N. This configuration is generally referred to as a PE-to-PE system architecture. Distributed routing control is assumed, with the settings of the individual interchange boxes (straight or exchange — broadcasting will not be considered in the model) determined by routing information contained within the routing tag of each message [14, 22]. Prior to the transmission of data between a source-destination pair in a circuit switched network implementation, a complete path linking the pair must be established and then held until the completion of the data transfer. Using packet switching, a message consists of one or more data packets. Each packet makes its way from stage to stage releasing links and interchange boxes immediately after using them.

The network is assumed to be operating under the following assumptions [8]. Each source PE generates its messages independently from all other sources. Each source generates messages for each destination PE with equal probability. Messages will be generated based on a predetermined, fixed loading factor. The destination PE can receive data from the network faster than the network can transmit the data (this simplification implies that an output device will not act to bottleneck the operation of the network itself). Finally, a message, or a message packet, consists of two parts -- the routing tag and one or more data words.

This research was supported by the Rome Air Development Center, under contract number F30602-83-K-0119.
In an interchange box, if two different message requests (routing tags) attempt to reserve the same output link, or (in the case of circuit switching) if a request is blocked by an already established message path, a conflict is said to have occurred. A conflict resolution algorithm must be invoked to resolve the conflict. In packet switching networks, the algorithm selects one of the requests and permits it to pass through the interchange box. The other request is held in an input buffer and will attempt to traverse the box at a later time. In circuit switching networks, the message that is blocked at the switch is either dropped and then resubmitted to the network (the drop algorithm) or is held in the box until the blockage is removed (the hold algorithm). These two circuit switched algorithms have been investigated in, for example, [15, 6].

3. MULTIPLE-PACKET NETWORK OPERATION

In this section, the effects of multiple-packet messages on the operation of a packet switching generalized cube network are investigated. The system is assumed to be operating in the MIMD mode. The message to be transmitted will consist of a routing tag and one or more data words. If, however, the message size exceeds the size of the network transmission packet, multiple dependent packets must be generated. Each packet will contain the same routing tag and a portion of the data to be transferred. The packets are processed sequentially by the network since only one network path exists between any source-destination pair.

Operational dependencies exist between multiple-packet messages that do not exist when single-packet message formats are used. These dependencies center on the sequential generation of packets (within any given message) all being routed to the same network output port (recall that in single-packet messages, the routing of all of the packets is assumed to be a strictly independent process [8]). As an example, consider two multiple-packet messages, where the packets of one message are being held in one input buffer of an interchange box and the second message's packets are held in the other input buffer. Any conflict (or lack of a conflict) that occurs when the first packets in the two buffers are processed will also occur when the succeeding packets are processed.

The dependencies that exist between packets in multiple-packet message transmissions cause network performance analysis using "standard" Markov chain modeling techniques, as in [8, 3], to be extremely complex. Interconnection network simulators such as PUGS [5] can, however, be used to accurately predict the performance of multiple-packet messages. PUGS is capable of simulating both single- and multiple-packet message formats. When using the multiple-packet format, performance statistics for both individual packets and complete messages (composed of dependent packets) can be obtained. The accuracy of PUGS output data has been validated through comparisons with other researchers' published network simulation and Markov chain performance data, such as [6, 3, 15, 8]. To increase the accuracy of the statistics, the length of the simulation runs was adjusted so as to insure the simulated network had reached a "steady-state" operating condition. This occurred after the first 250-300 simulated packet cycles. The actual simulation lengths were then set to approximately ten times this value -- 2000 simulation cycles. Statistics gathered before steady-state was reached were discarded. Lastly, for each combination of network variables that were evaluated, each simulation was repeated ten time and the results averaged so as to further reduce the statistical variations. Data and statistics obtained from PUGS will be used throughout this paper.

Assume that the data to be transferred is stored in the memory of the source PE. As a result of the slow memory access time within the PE (due to memory device speed limitations and long inter-IC and inter-circuit board transmission delays) and the comparatively fast interchange box logic (small, fast packet buffers and straightforward control logic), the time required by a PE to construct a data packet will be much longer than the time required by an interchange box to process the same size packet. This operational speed differential between the PE and the network (i.e., the interchange boxes) requires that packets be held in a PE's output buffer and not released to the network until the packet is completely formed.

In analyzing multiple-packet message performance, the packet cycle time, defined as the time delay incurred by a packet in traversing a network interchange box, will be used as the basic unit of system/network timing. This delay is the amount of time required to move a packet from the front of one of an interchange box's input buffers to the input buffer of an interchange box in the next stage. As discussed above, this time delay will be much shorter than the time delay associated with the generation of a packet within a PE. A packet offset time, the time (in packet cycles) between packet generation of a multiple-packet message, can be used to quantify the speed differential between the network and the system PEs. If the time to generate a packet equals the time to process a packet in an interchange box, the packet offset would be one. If a message consisted of w packets and the first packet was generated and then submitted to the network at time t, then, in general, a y-cycle packet offset would cause packets to be submitted at times t + y + tw, . . . , t + y(w−1). The total message transmission delay is the time difference between the beginning of the generation of the first packet of the message and the completed transmission time of the last packet of the message. Note that the time from the generation of the first packet to its submission to the network is y. This delay will be y(w)+ (the transmission delay of the last packet), where the first term is the delay incurred by the message before the last packet is available for transmission. Recall that, because of pipeline effects, the delays of the other packets in the message will be reflected in the transmission delay of the last packet.

In [13], Kruskal and Snir have presented an alternate, more constrained, discussion of multiple-packet network operation. They assumed that all of the message packets were generated simultaneously (this corresponds to a packet offset of zero) and that the packets moved through the network as a single "group." Additionally, the inclusion of the packet generation time in the overall transmission time has not been incorporated into previous analyses. In [8], for example, timing delay calculations start when a packet is accepted by a network input port. Time devoted to generating a packet and any time lost waiting for the availability of an network input port is not included in their analysis.
Representative performance analysis results of multiple-packet messages, obtained using PUGS, are shown in Table 1. Results in this table are for a 5-stage network (N=32) and multiple-packet messages consisting of 2, 4, and 8 packets. The packet offset ranges from 1 to 15 packet cycles and the loading factor is 100% in each case. The loading factor is the probability in a given time cycle of a network source generating a new message, given that it is not currently transmitting a message. Buffers at the inputs to each interchange box, capable of holding four packets, are assumed to be in use (buffer size is based on analysis results in [8]). The packet and message delay times can be used to gauge the effects of the packet offset on network performance. Ideally, a packet would require k packet cycles to traverse a k-stage network. An m-packet message would, in turn, require k+m(m-1) packet cycles to completely traverse the network (this time value assumes the packets are transmitted through the network in a pipeline fashion). The normalized packet and message delays can therefore be computed by dividing the packet and message delay times by k and k+m(m-1), respectively.

<table>
<thead>
<tr>
<th>Number of packets per message</th>
<th>Packet offset</th>
<th>Delay</th>
<th>Normalized delay</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Packet</td>
<td>Message</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>17.90</td>
<td>20.94</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>16.35</td>
<td>20.91</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6.75</td>
<td>16.60</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6.57</td>
<td>26.36</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>6.56</td>
<td>36.35</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>21.24</td>
<td>28.18</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>17.66</td>
<td>27.19</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6.40</td>
<td>26.29</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6.30</td>
<td>46.51</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>6.28</td>
<td>66.21</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>26.40</td>
<td>40.83</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>21.53</td>
<td>40.83</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6.29</td>
<td>46.36</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6.18</td>
<td>86.59</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>6.16</td>
<td>126.20</td>
</tr>
</tbody>
</table>

Table 1. Performance results for multiple-packet message transmissions (5-stage network (N=32) and 100% network load).

From Table 1, it can be seen that, for any given message size, as the packet offset is increased, the delay experienced by a packet in the network decreases and approaches the minimum traversal delay (normalized delay of 1). This is to be expected since the apparent network loading decreases as the packet offset is increased, thus reducing the network conflict delays that a packet will experience. The overall delay experienced by the entire message is also seen to decrease as the packet offset begins to increase from its initial value of 1, again a result of decreased network conflicts. However, as the packet offset continues to increase, the message delay time is seen to drop to a minimum level and then begin to increase. The increase is due to the longer delays between packet submissions that unnecessarily increase the message transmission delay. Packet offsets that lead to minimized overall message delays are in the range of 2 to 5. For example, message delays for two and four packet messages are minimized with a packet offset of 5, while an offset of 2 produces minimum delays in an eight-packet message. PUGS was also used to collect statistics for many different combinations of network operating parameters, such as: network size, number of packets per message, network loading, and packet offset values. In all cases, packet offsets in the area of 2-5 have been seen to produce the minimum message delays through the network.

In a typical system development process, the speed of the PEs is determined by the selection of a target microprocessor chip/chip set. The packet offset will then be fixed by the specification of the network interchange boxes and their packet cycle times. The analysis discussed above allows the most cost-effective tradeoff point for the network's hardware design to be predicted. The network's estimated performance over a range of cycle times can be used as a guide in determining the required hardware sophistication of the network design. The most desirable design, in terms of the cost-performance tradeoff, would be the slowest and typically the least expensive network that did not act as a bottleneck to the anticipated inter-PE message flow.

4. PACKET vs. CIRCUIT SWITCHED OPERATIONS

In inter-computer networks, the classic tradeoff between circuit and packet switching is one of message length. Long message formats are best supported by circuit switching whereas short, "bursty" formats are ideally suited for packet switching. The design of these networks tends to neglect propagation delays through the switch points in the network and instead concentrates on the transmission channel (link) delays [21, 25]. The differences in the two switching methods are not as clear-cut in intra-computer networks found in parallel processing systems. Because the processors are generally "close" physically, the interchange box delays dominate over the transmission link delays. System characteristics such as the operational mode (e.g., SIMD or MIMD), the architecture supporting the network, as well as the anticipated message format and system loading, greatly influence the selection of the switching method.

Systems that utilize the network solely for inter-PE data transfers (e.g., the PE-to-PE system architecture) can be characterized by lightly loaded, low message conflict networks. In this case, circuit switched networks can give very acceptable performances using circuitry that is much less complex than a comparable packet switched network (e.g., message queues and their control logic are not needed in the interchange boxes). Implementation of a circuit switched network is particularly advantageous in SIMD systems performing data permutations with no network conflicts where, once the path connecting the source-destination PE pair is established, the interchange boxes contribute only minimal gate delays to the transmission time.

In systems where the network is used to connect processors to memory modules (e.g., the processor-to-memory system architecture), inter-processor communication and instruction fetch operations utilize the network. The network loading would be expected to be high due to the preponderance of instruction and data fetch operations that must be supported. Packet switched networks have been proposed for this environment [8, 9]. Note that, through the use of caching techniques, the network loading can be substantially reduced and, in effect, giving a network performance similar to the PE-to-PE architecture's. PE-to-PE architectures will be assumed in this paper.

When packet switching is used for inter-PE transfers, if the message exceeds the packet length, multiple packets must be sent through the network. The processors must absorb the added overhead of message packetization (at the source PE) and packet recombination (at the destination PE). Each packet must independently perform its own routing/path establishment. These delays, coupled with the queuing delays at each switch, can increase the overall message transmission time in a packet switching network.

In this section, performance tradeoffs between packet and circuit switched interconnection networks will be explored. The analysis will be done using the same external system conditions in both switching modes. SIMD and MIMD operations will be considered separately.

4.1 SIMD operations

SIMD systems are generally used to execute data permutations, which are then transferred from the network sources to a set of network destinations using a particular transfer scheme. Cube-type networks have been found to perform most useful SIMD permutations in...
The performance of a circuit-switched network could possibly be improved through the use of data pipelining (with data buffers placed between the network stages) or the use of a message-switched network design [12]. If, on the other hand, the generation time is greater than the transmission delay, the total time will be \( L \) generation delays plus the transmission delay of the last data word. In other words, if the transmission delay of the network is small with respect to time required by the source PEs to generate the data words, the overall data transmission time of a message will be determined by the speed of the PEs -- not the speed of the network. In these instances, circuit-switched network performance enhancements such as data pipelining and message-switching will not be considered in this paper.

For example, the Motorola MC68010 microprocessor, operating at a clock frequency of 10 MHz, is capable of performing one memory-to-memory data transfer in, at best, 800 ns (the network source and destination ports are considered to be memory addresses by the PE) [17]. If the 2-by-2 switching logic of the interchange boxes is implemented using multiplexers or by two levels of tri-state buffers, \( T_b \) can be expected to be approximately 20 ns (worst-case, TTL logic). \( T_c \) can be assumed to be no more than 5 ns. A reasonable value for \( T_{cu} \) is 100 ns. The total delay experienced by a data word would then be (from Equation 2)

\[
T_{message} = T_{setup} + T_{xmit} + T_{packet} = T_{setup} + (T_{xmit} + L) \cdot (PE \text{ word generation rate})
\]  

Clearly, for system and network sizes that are currently implementable, the speed of the PE will be the determining factor in the data transmission time. Using the representative timing values, listed above, the set-up time (from Equation 1) and the data transmission times can be combined into the overall message transmission time, \( T_{message} \):

\[
T_{message} = T_{setup} + T_{xmit} + T_{packet} = T_{setup} + (T_{xmit} + L)(PE \text{ word generation rate})
\]  

Next, consider the operation of a packet switched network. Note that, because there are no conflicts within the network, there will be no queuing delays within the network. The size of the interchange box buffers will not effect the overall network operation. With the exception of the logic to form and control the interchange box buffers, the architecture and speed of a packet switched interchange box will be the same as that of the circuit switched box, discussed above. The values for \( T_{set} \), \( T_{trans} \) and \( T_{r} \) will be the same. Let \( T_b \) be the time needed to access a buffer in an interchange box. Because the buffer will be physically close to the rest of the circuitry for the box, if not on the same VLSI chip, \( T_{b} \) will be substantially less than the time required to perform a PE memory access. Assume that buffer reads and writes can be overlapped. The delay, \( T_{packet} \) that a data packet would experience in traversing a packet switched box would be

\[
T_{packet} = T_{set} + (T_{trans} + T_{b} + T_{r}(P+1))
\]  

The factor \((P+1)\) is the actual packet size including the routing tag (P data words plus one routing word). Time required for
handshaking between adjacent interchange boxes can be overlapped with the other phases of $T_{packet}$.

Using dual-ported memories for the packet buffers, $T_q$ would be at least 100 ns. Substituting this and the previously assumed values, Equation 6 then reduces to

$$T_{packet} = 125(P+1) + 100 \text{ ns}.$$  \hspace{1cm} (7)$$

The time required by a PE to generate a packet would be $(P+1)\text{PE word generation time}$. Continuing with the MC68010 example, this would be, at best, 800$(P+1)$ ns. The ratio of the packet generation time and $T_{packet}$, the packet offset, will be approximately 6. This leads to an alternate view of the packet offset time for SIMD operations.

When more than one packet is generated per message, if the number of stages in the network is less than or equal to the packet offset, a packet will completely traverse the network before the next packet in the message can be formed by the source PE and submitted to the network. The pipelining of message packets will not occur—effectively eliminating one of the general advantages of packet switched networks. An L-word message, divided into $[L/P]$ data packets will have a transmission time of

$$T_{message} = \left[\frac{L}{P}\right](\text{Packet generation time})$$

$$+ n \cdot T_{packet}.$$ \hspace{1cm} (8)

The first term represents the delay between the generation of the first word in the message (contained in the first packet) and the time the last packet is submitted to the network. The second term is the time required by the last packet to traverse the network.

Table 2 lists the message transmission times for various sized messages using both packet and circuit switching modes. The message transfer times were calculated using the representative timing values for the interchange box components and PEs, discussed above. It is clear that, for SIMD operations, the circuit switched mode provides lower message transmission delays since the transmission of individual data words can be overlapped with the PEs' generation of the next data words to be transmitted. The overlap or pipeline performance normally associated with circuit switched operations is precluded by the processing speed differential between the PEs and the packet switched network. From Table 2, it is also observed that packet sizes of two or four words yielded generally lower message transmission times than the other packet sizes. Small packet sizes incur an additional generation overhead of having to write the routing tag to many different packets (e.g., for a 4-word message, four 1-word packets with routing tags must be generated compared to only two 2-word packets and one 4-word packet). As the packet size increases, the packet delay time through an interchange box, $T_{packet}$, increases, as does the packet generation time. The combination of these two effects causes an increase in the total transmission delay through the network of each packet and negates the relative advantage of having to transmit fewer total packets for a given message size. The performance of the 2- and 4-word packets were a compromise between the two extremes.

4.2 MIMD operations

The performance analysis and comparison of circuit and packet switched networks presented in the last subsection was based on the assumption that network transfers would take place without conflicts. In MIMD operations, this assumption is not valid. Here, the performance of a network will be affected by conflicts within the network and the associated queuing delays.

For MIMD networks, the circuit switched message transmission process can still be decomposed into the set-up and the data transfer phases, as before. The conflict and blocking effects on the network's MIMD performance can be predicted using the Markov analysis of [15] or by simulation. The analysis of packet switched networks must also include the effects of queueing delays within the buffers associated with each interchange box. The simulation methodology described in Section 3 for multiple-packet messages can also be used here.

Any attempt to directly compare the performance of circuit and packet switched networks must use the same operational environment for both networks. External (to the network) factors such as the system size, processing speed of the PEs, and the network loading factor must be the same for both networks. Additionally, the fundamental design of the two networks must also be the same (e.g., data path width, the technology used in the interchange box implementations, and the PE-network interface methodology). When these internal and external factors are fixed, a comparison between the two switching methods can be made.

Assume the networks are constructed such that the timing values used in Section 4.1 remain valid. Specifically, $T_{cu} = 100\text{ns}$, $T_b = 20\text{ns}$, $T_i = 5\text{ns}$, and $T_q = 100\text{ns}$. Furthermore, assume that the PEs are implemented using a clock frequency of 10 Mhz, as in an MC68010 microprocessor (the microprocessor requires 800ns to write one word from memory to the network). Under these assumed conditions, Tables 3 and 4 list simulation results for the message transmission times and throughput performances of circuit and packet switched networks for a range of loading values and message lengths. Comparing the data in these tables to that of Table 2, it can be seen that the MIMD performance of both network types approaches the SIMD performance values as the network loading is decreased (as would be expected). The performance relationship between the circuit switched conflict resolution algorithms is as predicted by [15]—the drop algorithm provides generally lower transmission times and higher message throughputs than does the hold algorithm.

Over the range of message lengths considered in Tables 3 and 4, minimum message transmission times for a given data transfer size (message length) can be obtained in the packet switched network when the entire message is contained in only one or two packets. As in SIMD operations, the performance trade-off is lighter network loading (number of packets) with larger packet sizes versus shorter packet transmission times of the individual packets with smaller packet sizes.

A representative plot of the message transmission times for both circuit and packet switched networks is shown in Figure 2. For each plot, the data used was taken from the conflict resolution algorithm (for circuit switching) or the packet size (for packet switching) that provided the lowest transmission times. The circuit switched network provided superior network performance for smaller message sizes while packet switching performed best with larger message sizes. This can be attributed to, in circuit switch-
Table 3. MIMD message transmission times for circuit and packet switched networks (4-stage network (N=16), delays given in microseconds).

<table>
<thead>
<tr>
<th>Switching mode</th>
<th>Packet size</th>
<th>Loading factor</th>
<th>Data transfer size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit (hold algorithm)</td>
<td>100</td>
<td>50</td>
<td>2.21</td>
</tr>
<tr>
<td>Circuit (drop algorithm)</td>
<td>100</td>
<td>5</td>
<td>7.25</td>
</tr>
<tr>
<td>Packet 1</td>
<td>100</td>
<td>50</td>
<td>8.89</td>
</tr>
<tr>
<td>Packet 2</td>
<td>100</td>
<td>5</td>
<td>13.55</td>
</tr>
<tr>
<td>Packet 4</td>
<td>100</td>
<td>50</td>
<td>21.76</td>
</tr>
<tr>
<td>Packet 8</td>
<td>100</td>
<td>5</td>
<td>39.45</td>
</tr>
<tr>
<td>Packet 16</td>
<td>100</td>
<td>5</td>
<td>64.59</td>
</tr>
</tbody>
</table>

Table 4. MIMD message throughputs for circuit and packet switched networks (4-stage network (N=16), throughputs expressed as the number of messages processed per microsecond).

<table>
<thead>
<tr>
<th>Switching mode</th>
<th>Packet size</th>
<th>Loading factor</th>
<th>Data transfer size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit (hold algorithm)</td>
<td>100</td>
<td>50</td>
<td>2.21</td>
</tr>
<tr>
<td>Circuit (drop algorithm)</td>
<td>100</td>
<td>5</td>
<td>7.25</td>
</tr>
<tr>
<td>Packet 1</td>
<td>100</td>
<td>50</td>
<td>8.89</td>
</tr>
<tr>
<td>Packet 2</td>
<td>100</td>
<td>5</td>
<td>13.55</td>
</tr>
<tr>
<td>Packet 4</td>
<td>100</td>
<td>50</td>
<td>21.76</td>
</tr>
<tr>
<td>Packet 8</td>
<td>100</td>
<td>5</td>
<td>39.45</td>
</tr>
<tr>
<td>Packet 16</td>
<td>100</td>
<td>5</td>
<td>64.59</td>
</tr>
</tbody>
</table>

Figure 2. Comparison of message transmission delays for both circuit and packet switched networks (4-stage network (N=16), 100% network loading, times in microseconds).

This paper has centered on the modeling and analysis of multiple-packet message formats and their relative performance advantages when compared to circuit switched networks. A multiple-packet message format is needed within an interconnection network when the message to be transmitted exceeds the network's packet size. In order to quantize network performance, the operation of interchange boxes, networks, and external systems were expressed as time functions. General equations relating these time functions to the performance of a network were developed. By using these equations and the PUGS simulator, the effects of any set of internal and external environment assumptions could be evaluated and network performance -- either packet or circuit switched -- could be predicted.

During the analysis of packet switched networks, the effects of various packet sizes was examined. The packet offset figure of merit was introduced and used as a performance gauge of the transmission rate differential between the network and the PEs. It was observed that the optimum network performance (in terms of low transmission delays and reasonable network component costs) could be obtained with a packet offset in the range of 2 to 5.

Comparing the performance of circuit and packet switched networks for a specific set of time function values, circuit switching was shown to provide minimum transmission delays in all
SIMD operations and in short message transfers in MIMD operations. Packet switched networks functioned better than circuit switched networks as the message length increased. The dominant element of the network performance was seen to be the processing rate of the PEs themselves.

A general methodology has been developed that quantized system and network parameters, thus allowing the performance of a network design to be evaluated under a set of anticipated operating conditions. By adjusting the assumed values of these parameters, this methodology can be used by other researchers as an aid in the design of their own networks.

6. REFERENCES


