PERFORMANCE OF A ROUTING SWITCH UTILIZING TRANSPUTERS

Technical Report CSE-92-08

Christopher Ward
and
Saqib A. Khan

Department of Computer Science and Engineering
Auburn University, AL 36849-5347

Yan-Chieh Chow
and
Richard Newman-Wolfe

Computer and Information Sciences,
University of Florida
Gainesville, FL 32611

Performance of a Routing Switch Utilizing Transputers

Christopher Ward and Saqib A. Khan
Computer Science and Engineering, Auburn University, Auburn, AL 36849

Yuan-Chieh Chow and Richard Newman-Wolfe
Computer and Information Sciences, University of Florida, Gainesville, FL 32611

ABSTRACT

Performance results recorded from a Transputer based packet routing switch are introduced and discussed. The routing switch is used to simulate a node in a satellite network which forms part of a testbed used to study characteristics of low altitude multiple satellite (LAMS) networks. The testbed is composed of five nodes. All nodes are able to directly communicate with one another, and three of the five nodes are equipped with laser transceivers. Results discussed in this paper reflect the design and performance of the Transputer based prototype. Throughput efficiency for the routing switch under a variety of test conditions is provided.

Nodes in the testbed are modeled using a Host computer connected to four RS232 serial boards and a Transputer network. Communications are full-duplex and expected to be between 1-5 Mbits/sec. The code for Layers 7-4 resides in the Host while the code for layers 3-1 resides in both the Host and the Transputer network. The highly mobile nature of the nodes in a LAMS network dictates that each node in the simulation must periodically update its routing tables, reflecting the connection changes on each external link. Since the Transputer network spans layer 1-3 of the ISO model, one of its layer 3 responsibilities is to update its routing tables also. This periodic updating requires dynamic routing table loading from the Host computer (layer 4) to the Transputer network (layer 3). Library functions are provided in layer 4 processes to deal with this service, however, accommodating these updates influences the design of the underlying Transputer network. Allowances must be made not only for expedient routing of messages within the Transputer network but also for rapid and efficient routing table updates *within* the Transputer network.

1 INTRODUCTION

One of the many components that requires development to realize a low altitude multiple satellite (LAMS) network is a retargetable communications laser. Initial studies into laser communications, target acquisition, and routing will be performed using a ground based testbed. One such testbed** forms part of a three year SDI research project to study different characteristics of low altitude multiple satellite (LAMS) networks [2, 1, 8]. The requirements for the testbed are to model a five node LAMS network configuration. Of these five nodes, only three nodes will actually use laser trackers to communicate with each other. The others are "directly connected" to one another through link simulators. These simulators permit a variety of link conditions to be investigated including bust errors, random errors, and link failure. The testbed model is illustrated in Figure 1.

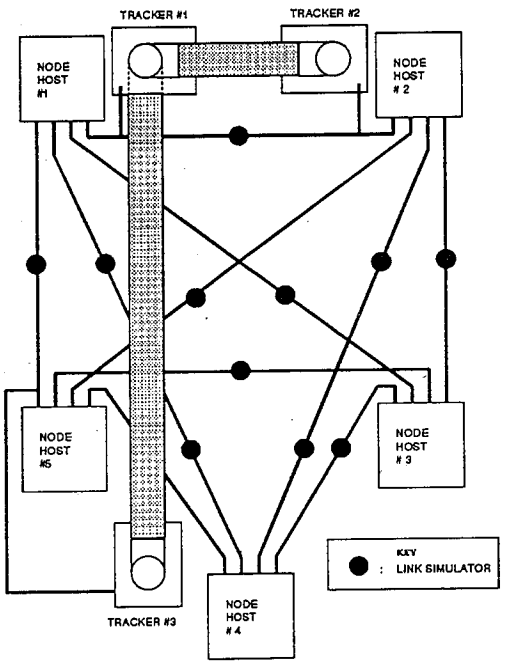

Figure 1 Testbed for five node simulation.

The "directly connected" links require communications to be full-duplex with data rates at least 1Mbits/sec to adequately model a LAMS network node. Since a node

** "A simulation testbed for a satellite computer network with laser cross-links," Strategic Defense Command, Prime Contract DASG60-89-C-0119.

is modelled using a Host PC which cannot alone satisfy the high data rate requirements, a high speed peripheral device with a PC interface is desired. Two candidate systems which would satisfy the data requirements are dedicated hardware (e.g. a board using four digital signal processing chips such as the TMS320C40) or an off-the-shelf system such as a Transputer board. In this prototype a Tranputer system is evaluated, with the remainder of the testbed simulated. Details of the original design are provided in [9, 3]. This paper reviews some material and identifies design modifications where appropriate.

A Transputer is a complete computer on a chip. The Transputer used (a T414) is composed of a 32 bit processor, with a 50 ns processor cycle, 2Kbytes of fast (50 ns cycle) static RAM, a programmable memory interface (which allows up to 4 Gbytes physical memory external to the Transputer) and four serial communications links (external communication links). The Transputer links operate at 10 or 20 Mbits/sec, full duplex. Each link is implemented as an autonomous DMA (Direct Memory Access) engine, and so can perform communications with external devices as background tasks to the processor with negligible performance degradation. The Transputers high external data rates coupled with Transtech's "Genesys" operating system [5, 4, 6] and high level language compliers makes it an ideal choice for the Host PC's high speed data communications needs.

Each node will communicate with the four other nodes using the Transputer network via the transputer links. Control of these links (e.g. laser pointing, acquisition and status) is conveyed by separate low speed (i.e. 9600 bits/sec) *order wires*. For the prototype each of the other nodes will be modeled using a single Transputer. The order wire communications software has been included to gauge the effective overhead but at present is nonfunctional.

The highly mobile nature of the nodes in a LAMS network dictates that each node in the simulation must periodically update its routing tables, reflecting the connection changes on each external link. Since the Transputer network spans layer 3-1 of the ISO model, one of its layer 3 responsibilities is to update its routing tables. This periodic updating requires dynamic loading of the tables from the Host PC (layer 4) to the Transputer network (layer 3). Extra library functions are provided to layer 4 processes to deal with this service, however, accommodating routing table updates influences the design of the underlying Transputer network. Allowances must be made not only for expedient routing of packets within the Transputer network but also for rapid and efficient routing table updates into the Transputer network.

Section 2 introduces the prototype design and indicates the significant changes from previous work. Section 3 discusses the selected test configuration and provides initial throughput statistics. Results for throughput efficiency for the routing switch under a variety of test conditions are then introduced in Section 4. These are used to identify the impact of differing traffic metrics and routing update frequency on the network. Conclusions drawn from these are discussed in Section 5.

2 TRANSPUTER NETWORK DESIGN

The design of the Transputer network is based on two components: the physical configuration of the Transputer network (if more than one TRAM is used) and the accompanying software design. The software design is directly dependent upon the physical configuration chosen for the Transputer network. This is due to the fact that processes which will route messages through the Transputer network cannot be designed unless they are aware of the network topology in advance. This issue becomes more clear in the following sections, where the physical configuration and software design are presented.

Physical Configuration

The physical configuration of the underlying Transputer network is based primarily around two priorities: efficient routing of data messages and dynamic routing table update capability. Due to the small number of links (4 per TRAM) provided for each TRAM a system configuration must be devised that provides maximum efficiency in dealing with the testbed node's four external links. This is accomplished by dedicating a TRAM to each external link (referred to as the Xlink, external link, TRAMs). The other three remaining links (on the Xlink TRAMs) are used to connect all four TRAMs together, thereby providing a fully connected network.

The connection between the host computer and the underlying Transputer network is established by inserting a "bridge" TRAM between one of the message streams connecting adjacent Transputers. This bridge TRAM acts as a transparent link between the adjacent TRAMs as long as no layer 3-4 communication is needed. If such a need arises, the bridge TRAM either inserts (layer 4 to 3 com-

munication) or accepts and delivers (layer 3 to 4 communication) messages from the passing message stream. Figure 2 illustrates the Transputer network described.

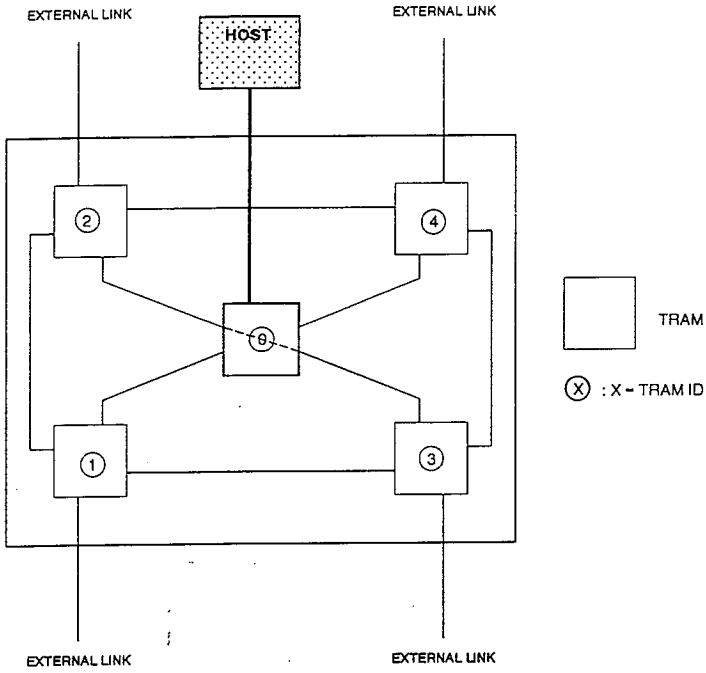

Figure 2 Transputer network configuration for a single testbed node. *Note* dotted line implies wire passing under TRAM 0.

Although an efficient physical configuration has been realized, a new complication arises. Namely, that "fringe" Xlink TRAMs (nos 2 and 3 in Figure 2) must route messages through the TRAMs directly connected (nos 1 or 4) to the "bridge" TRAM to reach the Host, and vice versa. Although we can forego the physical configuration in Figure 2 in favor of direct connection between the bridge TRAM and all link dedicated TRAMs, this cannot be achieved without introducing extra TRAMs which will serve no other purpose than solving connectivity constraints for the network, and introduce additional throughput delays.

The lack of direct connectivity between fringe TRAMs and the bridge is addressed by introducing *internal* routing of messages within the Transputer network, as explained in the next section.

Software Design

Utilizing the physical configuration described in Figure 2 the software design of the Transputer network is proceeds as follows. The Transputer network's external channels receive network layer protocol data units (NPDUs) from other nodes and either accept these messages (i.e. deliver them to the layer above) or route them to the next node (as computed by routing algorithms) via another external channel. In addition to providing the network layer service access point (NSAP) the Transputer network must be designed so that dynamic routing table updates can be performed transparently through the NSAP. I/O between higher layers and the layer 3 processes is accomplished using an interface control information (ICI) message format which allows interface data unit (IDU) destinations to be classified either as "external" (i.e. messages destined for an external link) or "internal" (i.e. messages destined for internal consumption by the Transputer network) bound messages. Processes discern between external and internal bound IDUs and route the messages accordingly to their destination within or outside the Transputer network. The IDU message format used to accomplish transparent I/O through the NSAP is described in detail in [3].

TRAM Process Models All process models make extensive use of concurrent process capabilities and message passing mechanisms supplied in Transtech's "Genesys" operating system. The processes communicate with each other through the use of internal or external message passing requests (which behave essentially in the same manner as queues). All processes wait for message delivery. If a message is received the process is activated and removes the message from its queue, performs the necessary tasks on the message, and once again resumes its wait operation. (*Note*: Processes are identified as circles in process models.)

Two separate process models are used to implement each node's Transputer network. The "Bridge" model and the "Xlink" model. The Bridge processes are resident upon the Bridge Transputer (no. 0 in Figure 2) and the Xlink processes are resident upon the four Xlink TRAMs (nos 1-4 in Figure 2). Both of these models share three elementary processes to perform equivalent duties, these are the "recver" (receiver), the "sender" (sender), and the "rtr" (router) processes. An operational flowchart of these processes is provided in Figure 3.

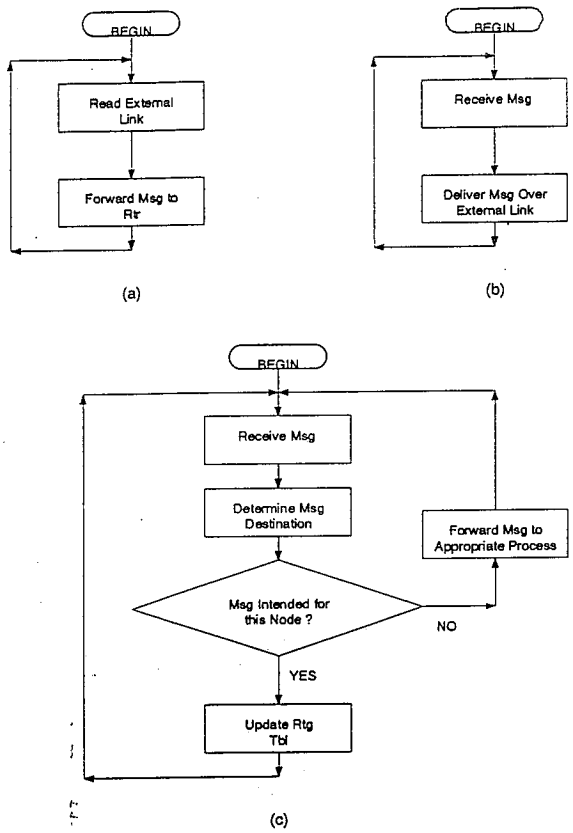

Figure 3 Operational flowcharts of: (a) recver (b) sender (c) rtr.

Each "recver" process is dedicated to a unique external link of the Transputer. The purpose of the "recver" process is to accept incoming messages on its dedicated link and pass them onto the "rtr" process. The "rtr" (router) process accepts messages and routes them according to their ICI's destination (DEST) and control (CTRL) fields. This routing decision is made by looking up 2 separate routing tables: the Internal routing table and the External routing table. The Internal routing table, which is static, reflects the internal topology of the Transputer network. The External routing table reflects the current connections of the external links of the satellite node and may be changed dynamically. If the message is not intended for the "rtr's" TRAM then the message is passed onto an appropriate process (usually a "sender") as determined by the routing tables. Otherwise, the message is parsed to confirm validity (in the present case only routing table updates are sent to TRAM processes) and an appropriate action is taken. In the case of a routing table update the external routing table would be updated and the message discarded.

The "sender" processes, like the "recver" processes, are single link dedicated processes. These processes accept messages sent to them and immediately transmit these messages over their dedicated link.

Bridge Process Model The Bridge processes are designed to provide three services: the layer 3/4 interface i.e. accepting information from layer 4 and sending information to layer 4; act as a transparent bridge for internal/external bound traffic; and update personal routing table when required. Layer 4 IDUs sent to the bridge can be composed of data intended for external link delivery or I/O intended for Transputer network consumption as identified by the ICI. I/O information is currently used for routing table updates. NPDUs sent to layer 4 currently takes the form of data intended for this *node*. An operational flowchart of the Bridge is provided in Figure 4 and the Bridge model is provided in Figure 5. This differs from the previous design model in that a single router process is used between the inputs and outputs as opposed to having each input process access an output queue. This improvement removes the requirement for explicit "critical region" code within the process model at the cost of having a potential bottleneck at the router.

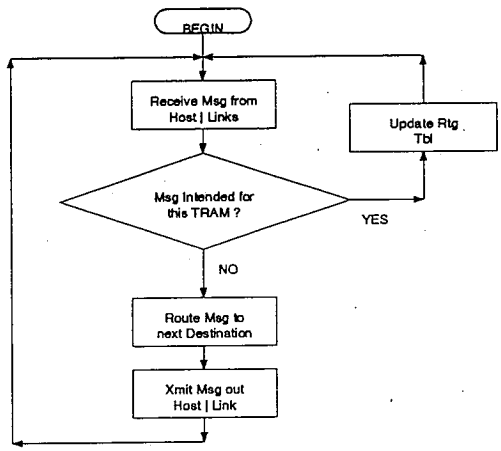

Figure 4 Bridge operational flowchart.

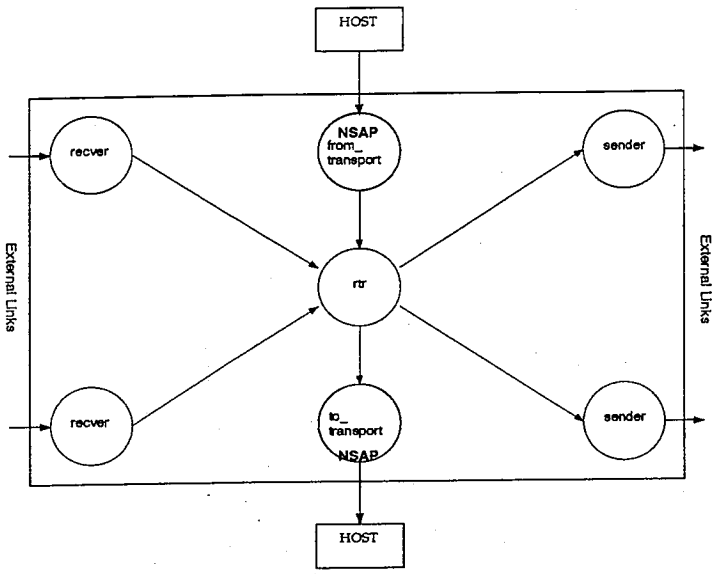

Figure 5 Bridge Process Model

The "recver" processes functions by forwarding incoming messages to the Bridge's "rtr" (router) process. In addition to performing all standard router functions (as described earlier) the Bridge "rtr" is designed to be capable of receiving IDUs from the Transport layer (or host computer) and routing NPDUs to the Transport layer, transparently. This "transparency" is achieved by the Bridge "rtr" communicating with two processes that directly communicate with the Transport layer. These processes are the "to_transport" process and the "from_transport" process. The Bridge "rtr" simply routes all messages according to its routing tables. If a message is intended for this *node* then the routing tables force the bridge to forward the message to the "to_transport" process. Conversely, if the "from_transport" process forwards an IDU message to the Bridge "rtr", the "rtr" simply determines the next process to forward the message to by looking up its routing tables and continues.

The "to_transport" and "from_transport" processes are designed to be the actual NSAP to the Transport layer functions. The "from_transport" process functions by receiving IDU messages directly from layer 4 processes and the "to_transport" process passes messages directly to "installed" layer 4 processes. Operational flowcharts of the "to_transport" and "from_transport" processes are provided in Figure 6.

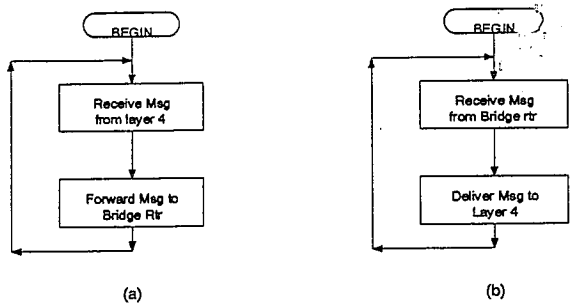

Figure 6 NSAP processes operational flowcharts (a) from_transport (b) to_transport.

Xlink Process Model Each of the four TRAMs connected directly to the four external links use the Xlink processes. These processes again have two main functions: routing NPDU messages and accepting control information. Packet routing is performed by dedicating "recver" (receiver) and "sender" (sender) processes to each physical link of the TRAM and inserting a "rtr" (router) process between these processes. Again, this model differs from the previous design in that a single router process is used to coordinate activity. An operational flowchart of the Xlink processes is provided in Figure 7 and Figure 8 illustrates the process model used in each of the four TRAMs.

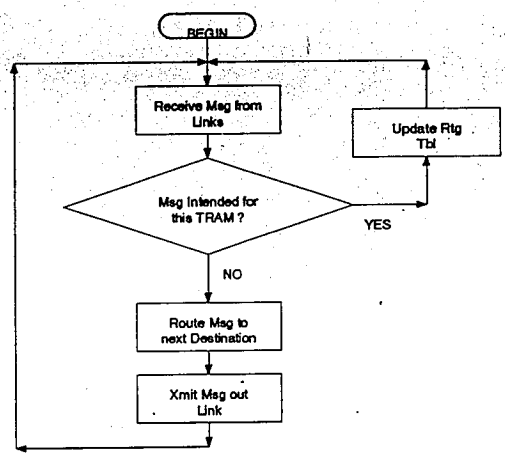

Figure 7 Xlink processes operational flowchart.

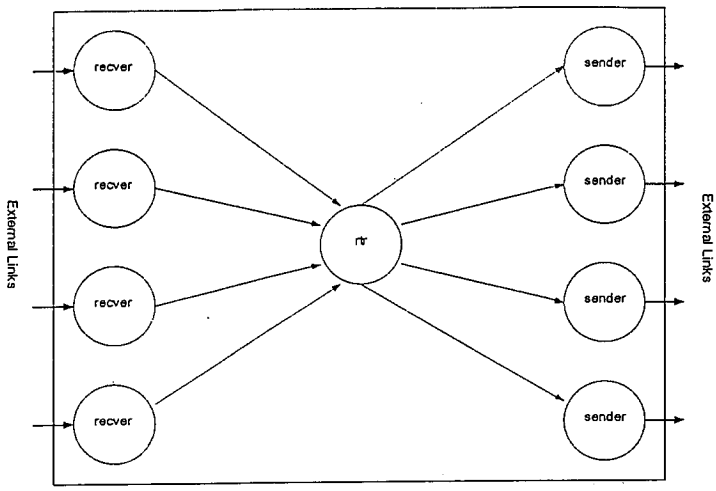

Figure 8 Xlink Process Model.

Compared to the Bridge processes the Xlink processes perform a less complex task. The Xlink "recver's" clear their respective links and forward the messages to the Xlink "rtr". The Xlink "rtr's" main concern lies simply in performing the duties identified earlier in the chapter, i.e. it simply routes NPDU messages as determined by the routing tables or ingests external routing table update NPDUs as identified by their ICI. Therefore when dynamic routing table updates are issued the entire Transputer network updates its external routing tables.

The "sender" processes simply accept messages from the "rtr" (when addressed by the "rtr") and transmit these messages over their designated links.

3 TRANSPUTER LINK THROUGHPUT

The Transputer links function at 10 or 20Mbits/s [7]. Their throughput was calculated using the Genesys supplied datalink layer functions so that the throughput results obtained for the routing switch could be meaningfully compared to the maximum data rate. The maximum data rate for the Transputer links was computed by sending messages from one TRAM to an adjacent TRAM over a specified link. The received messages were timed and the throughput was calculated. Figure 9 shows the configuration used.

Figure 9 Transputer link throughput test configuration.

Packets Generated	Packet Size	Packet Xmission Delay	Packet Destination
5000	Uniform (1002 bytes)	Uniform	Constant

Table 1 Link throughput test source configuration.

The source used a uniform packet size and a uniform packet delay (Table 1). The maximum throughput was calculated by decreasing the packet transmitting delay on successive runs of the test. Figure 10 shows a plot of the link throughput versus the packet transmission delay. The maximum throughput recorded from the test was:

$$\lambda_{max} = 0.7068 * 10^6 bits/s$$

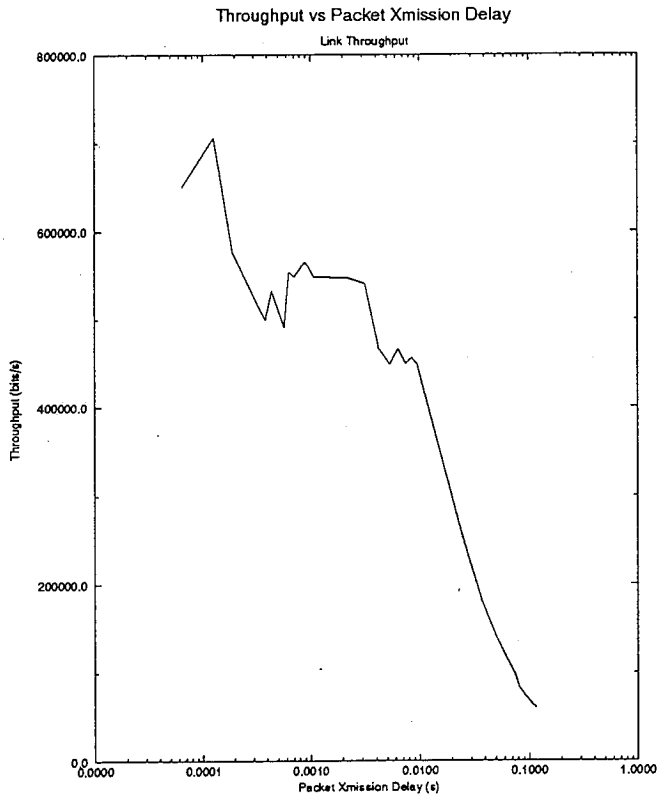

Figure 10 TRAM Link throughput plot.

4 SIMULATION RESULTS

Although the maximum data rate of the Transputer links is easy to calculate, the accumulation of meaningful performance results for the Transputer based router is

more complicated. This difficulty stems from the fact that there are numerous modes of operation in which these results could be accumulated, in particular the following cases are of interest:

- I: Single source to single destination (through bridge TRAM).
- II: Multiple source to single destination (through bridge TRAM).
- III: Standard performance, multiple source to multiple destinations.

Since some of these modes of operation yield redundant information, tests were performed on strategically identified configurations. These cases were chosen because they would yield valuable information in determining the routing switch's performance in its original testbed environment. The configurations selected were:

- Test 1: uniform packet size, uniform packet generation interval, constant destination, message traffic mode I.
- Test 2: uniform packet size, uniform packet generation interval, constant destination, message traffic mode II.
- Test 3: uniform packet size, uniform packet generation interval, constant destination, message traffic mode III.
- Test 4: normally distributed packet size, normally distributed packet generation interval, random destinations, message traffic mode III, host excluded.
- Test 5: normally distributed packet size, normally distributed packet generation interval, random destinations, message traffic mode III, host included.

The following sections describe each of the tests performed, their respective configurations, a justification for choosing these tests, and the results accumulated.

Maximum Switch Throughput, Test 1

The first test involves recording the throughput of the packet routing switch under ideal conditions. The purpose of this test is to compute the maximum possible data rate at which the switch can operate. These "ideal conditions" are defined as when the message traffic is one-way along a link and takes the shortest path through the switch from a source (src) to a sink (snk). Figure 11 illustrates the traffic pattern used.

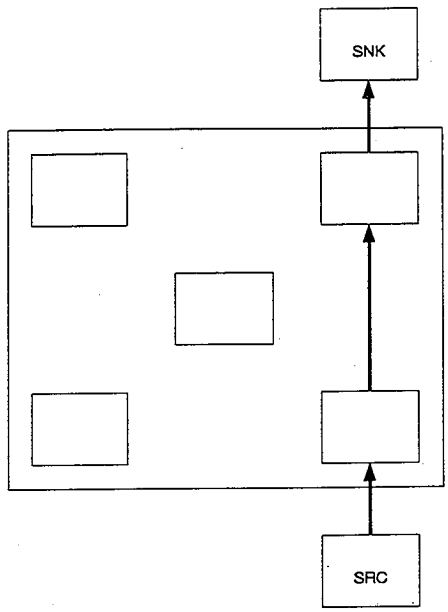

Figure 11 Traffic pattern for Test 1.

Packets Generated	Packet Size	Packet Xmission Delay	Packet Destination
5000	Uniform (1002 bytes)	Uniform	Constant

Table 2 Test 1 source configuration.

The generator used a uniform packet size and a uniform packet delay (see Table 2). The maximum throughput was calculated by decreasing the packet transmitting delay on successive runs of the test. Figure 12 shows a plot of the throughput versus packet transmission delay. The maximum throughput recorded from the test and the throughput efficiency (relative to the maximum link throughput) was:

$$\lambda_{max} = 0.5305 * 10^{6} bits/s$$
$$\lambda_{eff} = 75.06\%$$

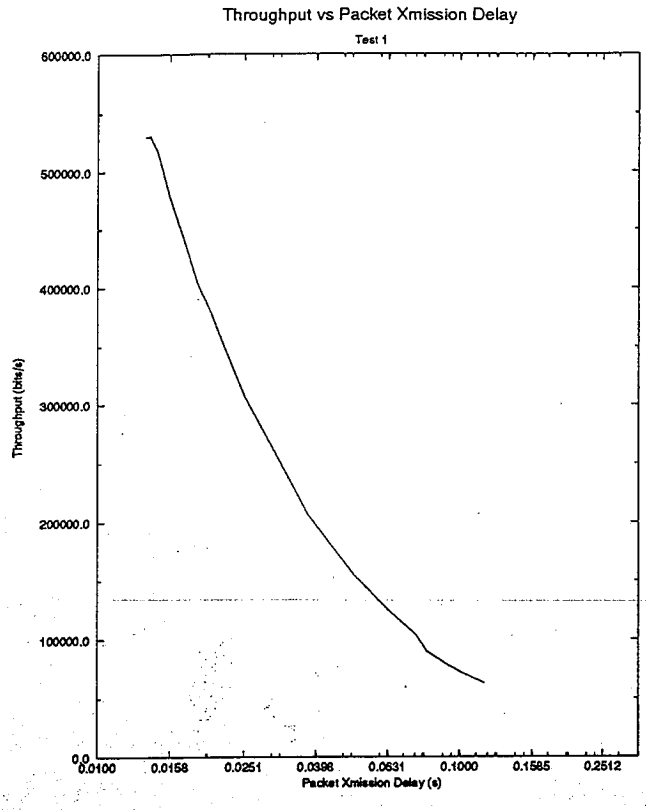

Figure 12 Maximum switch throughput plot.

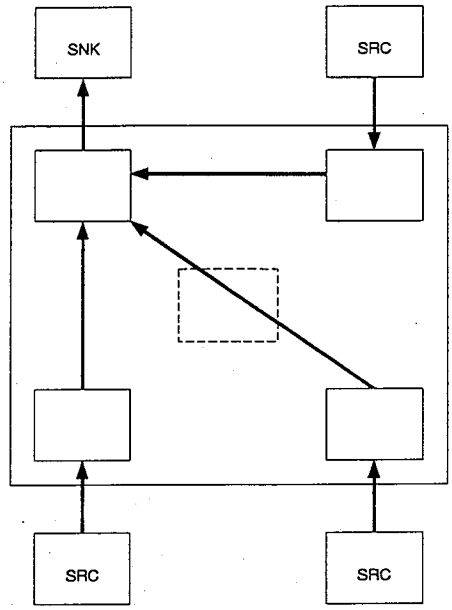

Figure 13 Test 2 Traffic pattern.

Packets Generated	Packet Size	Packet Xmission Delay	Packet Destination
5000	Uniform (1002 bytes)	Uniform	Constant

Table 3 Test 2 source configuration.

Congested External Link, Test 2

The worst possible performance of the packet routing switch is conceived as when all messages entering the switch are bound for the same external link. Congestion will result at the external link TRAM forcing the whole Transputer network to slow down. This condition is illustrated in Figure 13.

It should be noted that incoming traffic could be induced on the outgoing link (connected to the sink). This case was not explored because an assumption is made regarding the routing tables. Namely, that routing tables are assumed to be correct (i.e. not cyclic) and up to date at all times. Therefore, any incoming traffic on the external link would not require to be bounced back out the same external link.

The generator used a uniform packet size and a uniform packet delay (Table 3). The maximum throughput was calculated by decreasing the packet transmitting delay on successive runs of the test. Figure 14 shows a plot of the throughput versus the packet transmission delay. The maximum throughput recorded from the test and the throughput efficiency (relative to the maximum link throughput) was:

$$\lambda_{max} = 0.5678 * 10^{6} bits/s$$
$$\lambda_{eff} = 80.33\%$$

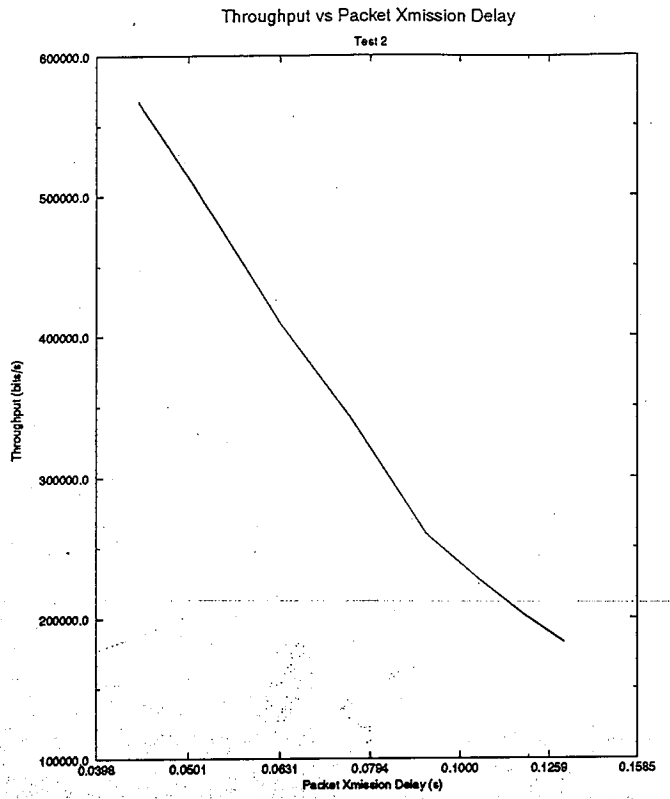

Figure 14 Congested external link throughput plot.

Impact of Dynamic Routing Table Updates, Test 3

The most interesting feature of the packet routing switch is its dynamic routing table update capability. The affects of dynamically loading routing table updates into the Transputer network during standard operation are of particular concern. Since the throughput will be directly effected by these updates this test was performed to measure the throughput as the update frequency was increased. For the sake of clarity the traffic patterns was kept constant in the switch while the throughput was recorded (see Figure 15).

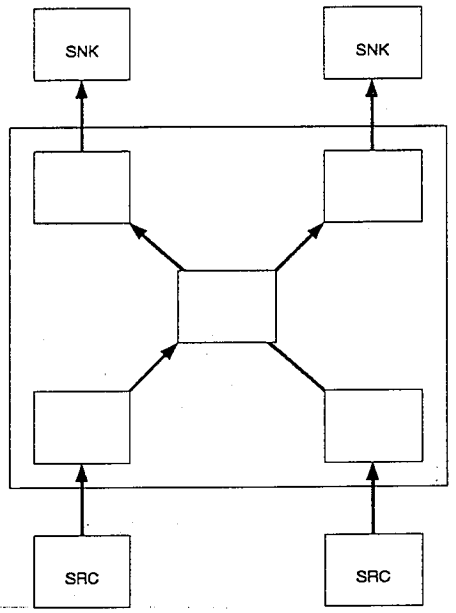

Figure 15 Test 3 traffic pattern.

To analyze the effects of the routing table update injection the maximum throughput of the switch was first recorded prior to introducing the updates. The source used a uniform packet size and a uniform packet delay (Table 5). The maximum throughput was calculated by decreasing the packet transmitting delay on successive runs of the test. Figure 16 shows a plot of the throughput versus packet transmission delay. The maximum throughput recorded from the test and the throughput efficiency (relative to the maximum link throughput) for traffic through the Bridge was:

$$\lambda_{max} = 0.4057 * 10^6 bits/s \text{ (through Bridge)}$$
$$\lambda_{eff} = 57.40\% \text{(through Bridge)}$$

The maximum throughput and the throughput efficiency (relative to the maximum link throughput) for traffic not passing through the Bridge was:

$$\lambda_{max} = 0.4067 * bits/s \text{ (Not through Bridge)}$$
$$\lambda_{eff} = 57.74\% \text{ (Not through Bridge)}$$

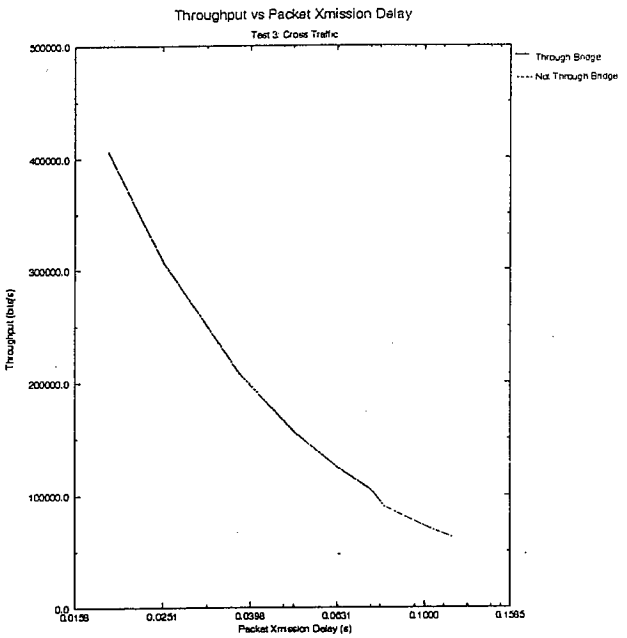

Figure 16 Test 3 Normal link throughput plot for cross-traffic.

Updates Generated	Update IDU Size	Inter-update Xmission Delay
Infinite (duration of test)	Uniform (12 bytes)	Uniform

Table 4 Test 3 Routing table update source configuration.

Once the maximum operating throughput of the switch was recorded, the switch was driven at this constant throughput and routing table updates were introduced. The routing table updates used did not reflect changes of external link connected nodes, rather they re-initialized the external routing tables to their original value, so that the throughput deterioration obtained could be compared to the maximum throughput obtained without the routing table updates. This deterioration demonstrates the overhead required to update external routing tables throughout the switch.

The update source used in the test was similar to the packet generators in capability. The only difference was that it distributed routing table update IDUs from the host. The update source sent five routing table update IDUs, each destined for a separate TRAM in the network. These update IDUs were delivered one after another without using any inter-departure delays. Once all five updates were injected into the Transputer network the update source

Packets Generated	Packet Size	Packet Xmission Delay	Packet Destination
5000	Uniform (1002 bytes)	Uniform	Constant

Table 5 Test 3 source configuration.

waited for the inter-transmission delay and then repeated the same operation. More details are provided in Table 4.

The inducement of routing table updates on the routing switch in addition to the cross traffic produces the traffic pattern depicted in Figure 17. The same source characteristics identified in Figure 5 were used to drive the switch again, except both sources were driven at the maximum allowable throughputs (as recorded earlier). Figure 18 depicts the effects upon the throughput due to an increase in routing table update frequency. Figure 19 shows the effects upon the end-end delay for the first packet delivered as the routing table update frequency is increased.

In both cases the effects are regarded as minimal since the variance in the throughput and the end-end delay are not significant. The results are reported below:

λ = Same as w/out updates (through Bridge)

λ_{det} = Negligible (through Bridge)

λ = Same as w/out updates (Not through Bridge)

λ_{det} = Negligible (Not through Bridge)

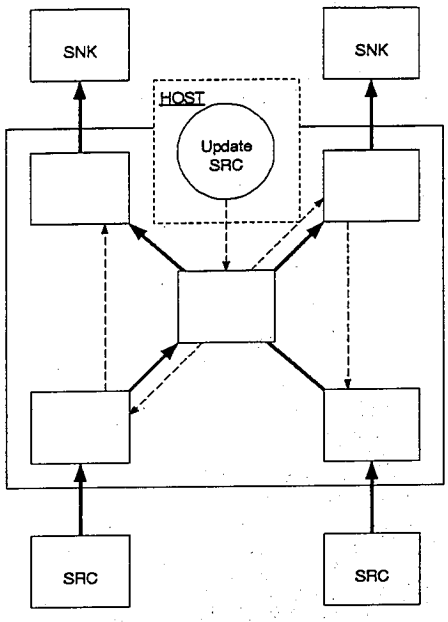

Figure 17 Test 3 Traffic pattern with routing table updates. *Note* dotted lines depict update paths.

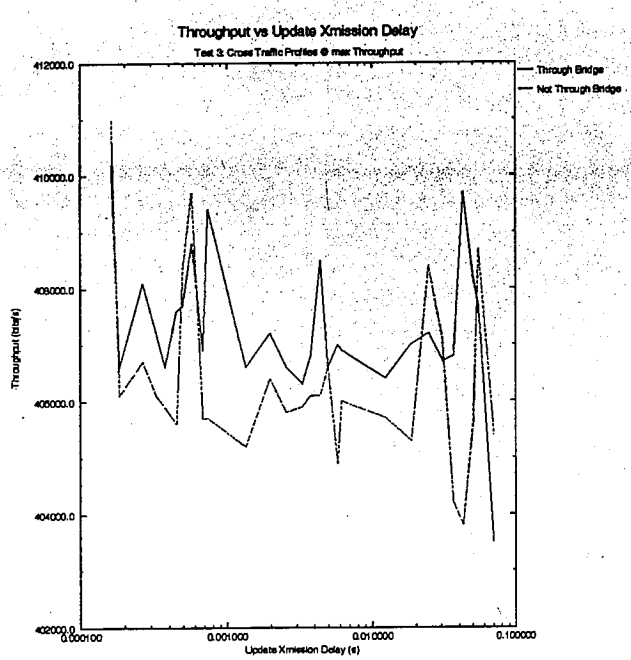

Figure 18 Throughput w/ Increasing Routing Table Update Frequency.

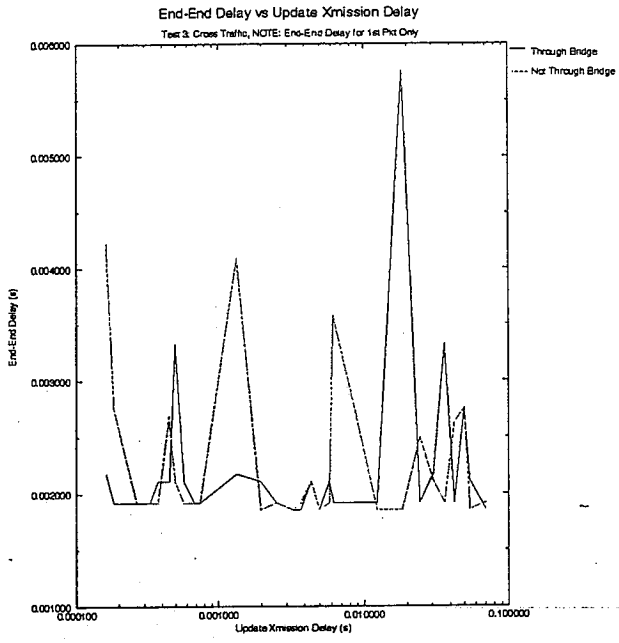

Figure 19 End-End Delay (1st pkt only) w/ Increasing Routing Table Update Frequency.

General Switch Operation, Test 4 and Test 5

The last two tests were conducted to generate data which would reflect the packet routing switch's general operational characteristics. In these cases, random destination traffic would be passing through the switch with varying packet sizes and varying inter-arrival times. Since we are interested in the switch's operation in the project testbed, packet sizes and inter-arrival times were matched to the expected traffic patterns.

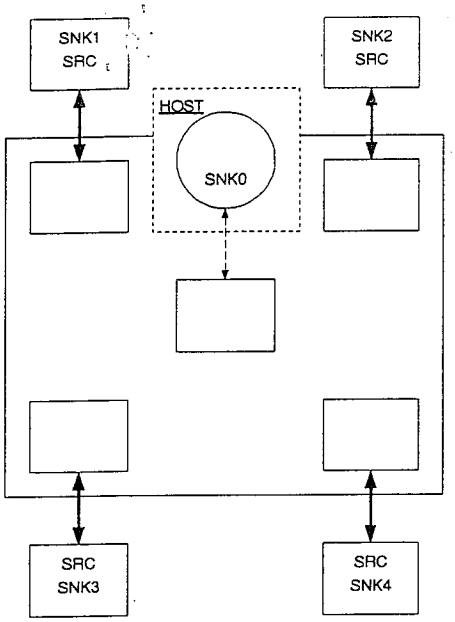

Figure 20 Test 4 traffic pattern.

Test 4 was designed to analyze throughput of the switch with random destination packets passing through the switch. Sources and sinks were connected to all the external links. A sink was also connected to the NSAP to absorb packets with destinations set for the packet routing switch's host node. It should be noted that no source was connected to the NSAP (depicting traffic orginating from the host node), this variation is explored in Test 5. Figure 20 depicts the sources and sinks present in Test 4 but does not show traffic flow, since it is random. Table 6 describes the source characteristics used to generate the throughput profiles for both Test 4 and Test 5.

Packets Generated	Packet Size	Packet Xmission Delay	Packet Destination
5000	Normal Distribution (Mean = 1002, Std. Deviation = 100)	Normal Distribution (Standard Deviation = 1.152e-03)	Random (Uniformally Distributed)

Table 6 Test 4 and Test 5 source configuration.

The generator used a normally distributed packet size and a normally distributed packet delay with a uniformly distributed random packet destination field (Table 6). The maximum aggregate throughput was calculated for each sink by decreasing the packet transmitting delay on successive runs of the test. Figure 21 shows a plot of the throughput versus the packet transmission delay. The maximum throughputs recorded from the test were:

$$\text{SINK } 0\ \lambda_{max} = 121.1 * 10^3 bits/s$$
$$\text{SINK } 1\ \lambda_{max} = 116.0 * 10^3 bits/s$$
$$\text{SINK } 2\ \lambda_{max} = 109.8 * 10^3 bits/s$$
$$\text{SINK } 3\ \lambda_{max} = 118.4 * 10^3 bits/s$$
$$\text{SINK } 4\ \lambda_{max} = 116.0 * 10^3 bits/s$$

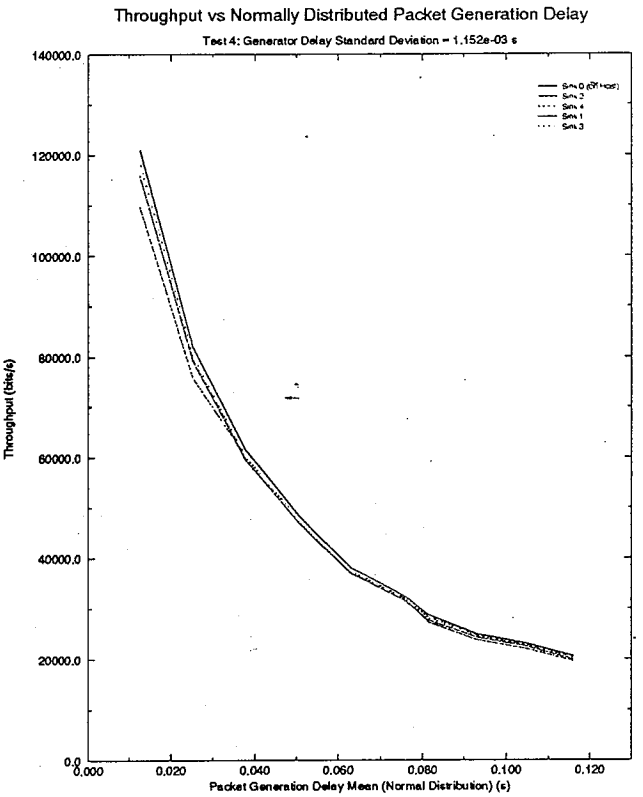

Figure 21 Test 4: Switch general operation throughput plot.

To analyze the general operation of the switch random traffic patterns were used in Test 5. The configuration used for Test 5 was identical to Test 4 except an additional source was connected to the NSAP, which was resident upon the host (depicting traffic originating from the host node.) Table 6 describes the source characteristics used to generate the throughput profiles.

The generator used a normally distributed packet size and a normally distributed packet delay with a uniformly distributed random packet destination field (see Table 6.)

Figure 22 shows a plot of the throughput versus the packet transmission delay. The maximum throughputs recorded from the test were:

$$\text{SINK 0 } \lambda_{max} = 67.78 * 10^3 bits/s$$
$$\text{SINK 1 } \lambda_{max} = 62.87 * 10^3 bits/s$$
$$\text{SINK 2 } \lambda_{max} = 66.14 * 10^3 bits/s$$
$$\text{SINK 3 } \lambda_{max} = 64.27 * 10^3 bits/s$$
$$\text{SINK 4 } \lambda_{max} = 66.38 * 10^3 bits/s$$

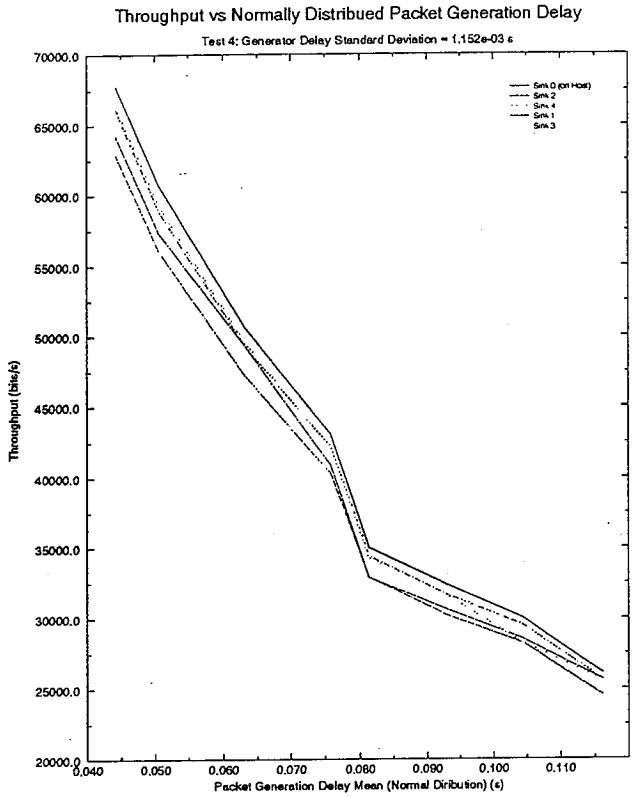

Figure 22 Test 5: Switch general operation throughput plot.

5 CONCLUSIONS

In this paper we have introduced the design process required to realize a Transputer based packet routing switch using a network of T414 Transputers and the Transtech "Genesys" operating system. The results obtained from this simulation are summarized in Table 7 which includes the aggregate data rate for the routing swich taken as a whole.

Test	$\lambda_{aggregate}$		λ	$\lambda_{relative}$
Transputer Link Throughput	706.8 Kbits/s		λ_{max} = 706.8 Kbits/s	N/A
1	530.5 Kbits/s		λ = 530.5 Kbits/s	λ_{eff} = 75.06%
2	567.8 Kbits/s		λ = 567.8 Kbits/s	λ_{eff} = 80.33%
3	812.4 Kbits/s	Direct Traffic (no updates)	λ = 406.7 Kbits/s	λ_{eff} = 57.54%

Table 7 Cumulative results (Continued . . .)

Test	$\lambda_{aggregate}$		λ	$\lambda_{relative}$
		Direct Traffic (w/ updates)	λ = 406.7 Kbits/s	$\lambda_{Deterioration}$ = 0%
		Through Bridge (no updates)	λ = 405.7 Kbits/s	λ_{eff} = 57.40%
		Through Bridge (w/ updates)	λ = 405.7 Kbits/s	$\lambda_{Deterioration}$ = 0%
4	581.30 Kbits/s	Sink 0	λ = 121.1 Kbits/s	N/A
		Sink 1	λ = 116.0 Kbits/s	N/A
		Sink 2	λ = 109.8 Kbits/s	N/A
		Sink 3	λ = 118.4 Kbits/s	N/A
		Sink 4	λ = 116.0 Kbits/s	N/A
5	327.44 Kbits/s	Sink 0	λ = 67.78 Kbits/s	N/A
		Sink 1	λ = 62.87 Kbits/s	N/A
		Sink 2	λ = 66.14 Kbits/s	N/A
		Sink 3	λ = 64.27 Kbits/s	N/A
		Sink 4	λ = 66.38 Kbits/s	N/A

Table 7 Cumulative results

Transputer Link Throughput

The maximum Transputer link throughput calculated was 0.7068 Mbits/s. This computed value is far less than the Inmos published link speed of 10 Mbits/s. The measured value was calculated using the Genesys supplied datalink layer message sending/receiving functions using 1002 byte long messages. The difference between the anticipate and recorded throughputs is large enough to require reconsideration of the Transputer system's suitability to the project testbed. This issue will be explored later.

Maximum Switch Throughput

The maximum switch throughput recorded was 0.5305 Mbit/s with a throughput efficiency of 75.06%. The throughput efficiency indicates that the software design functions well. If the Transputer links, with operating system overhead included, were indeed functioning close to 10 Mbits/s, the throughput achieved using the same software would be within the required testbed range.

Congested External Link

One of the most interesting results was gathered during Test 2. In this test the throughput recorded was 0.5678 Mbit/s with a throughput efficiency of 80.33%. This throughput efficiency is greater than that of Test 1, which mimics ideal switch routing conditions. The reason for a higher throughput efficiency in Test 2 is believed to be a better utilization of the services provided by the congested TRAM. That is, when all three internal links of the congested TRAM are loaded with packet streams the software utilizes its message receiving and processing capabilities better than when only one internal link is receiving messages. It is conjectured that the Transtech software processes message arrival interrupts more efficiently under heavy loads than under light loads.

Impact of Dynamic Routing Table Updates

The change in the throughput of the switch traffic in Test 3 was so low that it was undetectable and is considered negligible. This is not a surprising if we consider the fact that a routing message destined for each internal TRAM is only 12 bytes long (including interface

control information). The processing required to update a node's external routing table is also minimal, since the software design employs highly efficient C language macros to update the 10 byte long external routing table.

General Switch Operation

The throughput results obtained after conducting Test 4 are comparable to those obtained from Test 2 (the congested external link test). This is expected since all external links of the packet routing switch are heavily loaded during Test 4, similar to the conditions encountered during Test 2. Test 5, on the other hand produces more disturbing results. The addition of a packet generating source at the NSAP connection decreases the aggregate throughput by 253.86 Kbits/s.

The comparison between the $\lambda_{aggregate}$ (aggregate throughputs) of Test 4 and Test 5 yields several interesting points. Firstly, as mentioned earlier, it is noted that the addition of a packet generating source at the layer 4/3 interface has a significant effect on the switch throughput. This is expected since all the traffic passing through the Bridge will be forced to slow down when multiple packets of substantial length are injected into the Bridge Transputer. Secondly, these tests also established conditions in which all external links (including the NSAP link in Test 5) were fully utilized, i.e. heavy traffic was travelling both ways on the external links as well as the internal links. These conditions accurately reflect the switch in general operation under what would be considered a heavy traffic load. The results obtained show the switch throughput degradation (as compared to the maximum switch throughput) expected when heavy testbed traffic is encountered. Thirdly, and most importantly, the comparison between Test 4 and Test 5 has succeeded in pinpointing the weakest point in the packet routing switch design, the Bridge Transputer. As load is increased on the Bridge Transputer, the throughput of the whole switch degrades considerably.

From the data accumulated the software design of the switch yields efficiency results that clearly would be within the operating range specified for the testbed nodes, if the effective external data rate was indeed close to 10 Mbits/s. Since this is not the case, the suitability of the Transputer system for the project testbed is questionable.

6 REFERENCES

[1] C. H. Choi and C. Ward. The LAMS-DLC protocol. Research Project TR-CSE-91-03, Computer Science and Engineering Department, Auburn University, Auburn, AL 36830, 1991.

[2] Y. C. Chow, C. Mc Lochlin, R. Newmann-Wolfe, C. Ward, and J.N. Wilson. Optimizing the Delay and Reliability of Low Altitude Satellite Network Topologies. In *Milcom'87*, Washington D.C., October 19-22 1987.

[3] S. A. Khan and C. Ward. Design condsiderations for a high speed packet routing switch utilizing transputers. In *Simulation Technology Conference International 1991, Orlando, FL*, October 1991.

[4] Transtech Devices Limited. *Genesys C Reference Manual*, 1990.

[5] Transtech Devices Limited. *Genesys Command Reference Manual*, 1990.

[6] Transtech Devices Limited. *MCP 1000 Software Utilities Manual*, 1990.

[7] Transtech Devices Limited. *MCP 1000 Technical Reference Manual*, 1990.

[8] C. Ward and C. H. Choi. The LAMS-DLC ARQ Protocol. *Computer Communication Review*, 24(4), September 1991.

[9] C. Ward and S. A. Khan. Performance Analysis of a High Speed Transputer based Packet Routing Switch. In *Twenty-Second Annual Pittsburgh Conference on Modeling and Simulation 1991, Pittsburgh, PA*, May 1991.