# A Performance Evaluation of the SAFENET Lightweight System

Bert J. Dempsey, John C. Fenton, Jeffrey R. Michel,
Alexander S.Waterman and Alfred Weaver
Computer Networks Laboratory
University of Virginia

## 1. Introduction

This document describes the performance of the SAFENET Lightweight System. The performance of the system is measured at three levels—the user Ada program, the transport protocol (XTP), and the FDDI MAC interface. For each level we present measures of its throughput and latency. At the user level, such measurements are presented for both multicast and unicast transfers.

## 2. The User Level

The following measurements are for user-level data transfers performed by two Ada user programs running on two separate DTC-2 hosts. The data transfers were performed using the `SEND_MESSAGE` and `GET_MESSAGE` primitives of the Ada binding using `ASYNCHRONOUS` mode. All values are based on a time average over several individual transfers of a given message size. The message sizes used span the complete range of possible values.

### 2.1. Unicast Performance

For each unicast test, burst and rate control were used. `BURST` was set to 10,000 bytes/burst, and `RATE` was set to 1,500,000 bytes/second.

***Throughput***

Figure 1 shows unicast throughput. In this experiment several data transfers of a fixed message size were repeated in order to utilize the maximum possible bandwidth of the communications architecture. Here, throughput measures the total number of bytes of user data transferred from one host to another divided by the amount of time between the execution of the first and last `SEND_MESSAGE`, divided by the number of repetitions made. In this case there were 1000 repetitions. The graph has an asymptote at approximately 12 megabits per second, the maximum user throughput.

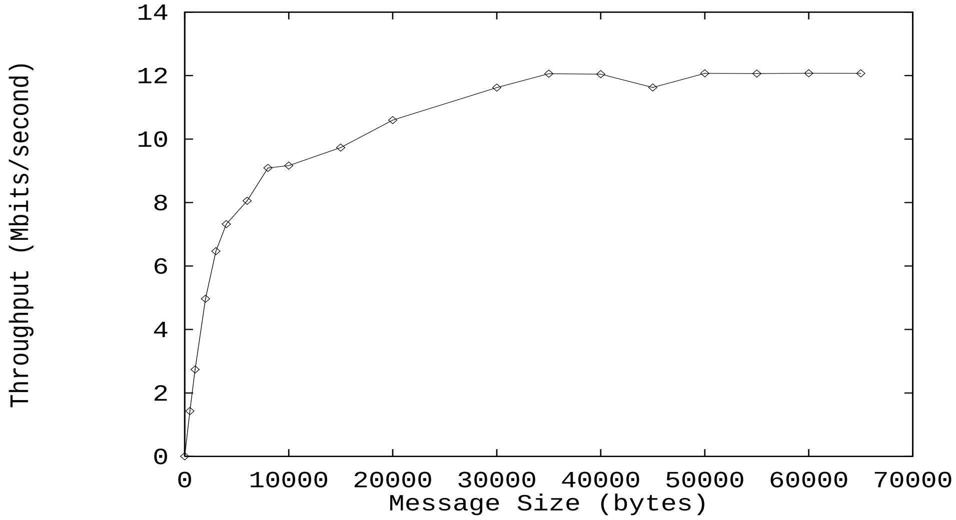


**Figure 1 — Unicast Message Size vs. Throughput**

***Latency***

Figure 2 shows unicast latency. Here, latency measures the amount of time required to asynchronously send a message of a fixed size to the receiver and then synchronously receive an echo of equal size in response, divided by two. The curve in the graph is approximated by the equation $y = 0.001299x + 5.42$, where $y$ is latency in milliseconds and $x$ is message size in bytes.

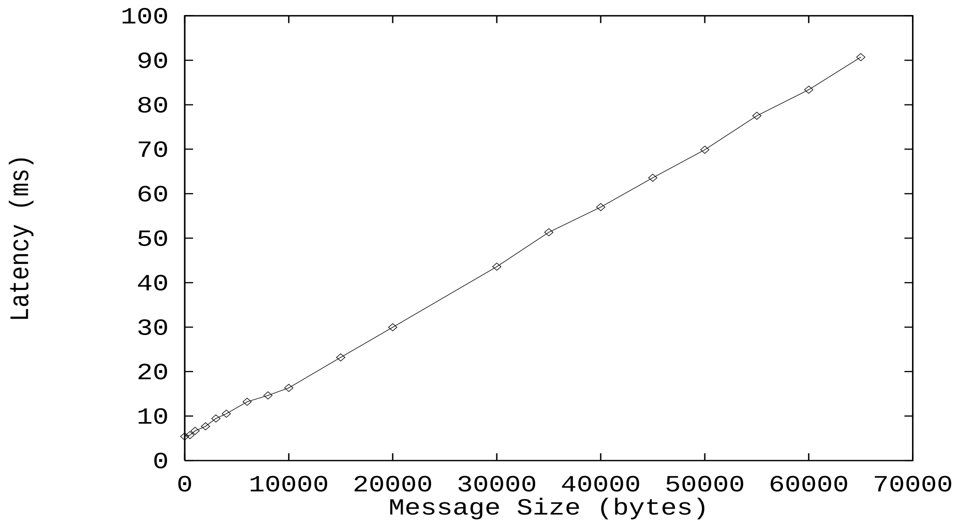


**Figure 2 — Unicast Message Size vs. Latency**

## 2.2. Multicast Performance

### *Throughput*

The multicast throughput measurements were taken using one DTC-2 sender multicasting to two DTC-2 receivers. Figure 3 shows multicast throughput. In this experiment several data transfers of a fixed message size were repeated in order to utilize the maximum possible bandwidth of the communications architecture. Here, throughput measures the total number of bytes of user data transferred from one host to the multicast receivers divided by the amount of time between the execution of the first and last SEND_MESSAGE, divided by the number of repetitions made. In this case there were 1,000 repetitions. The graph has an asymptote at approximately 12 megabits per second, the maximum user throughput.

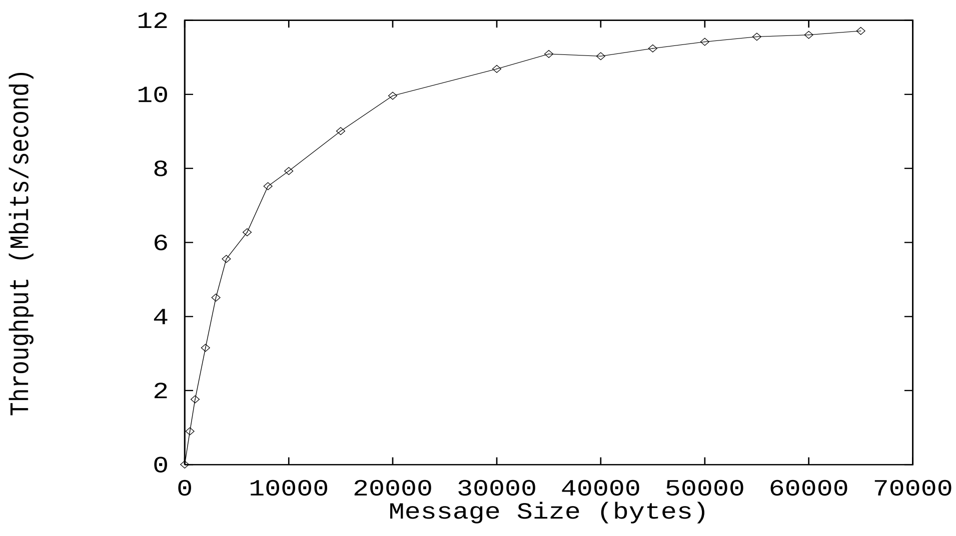


**Figure 3 — Multicast Message Size vs. Throughput**

***Latency***

The multicast latency measurements were taken using one DTC-2 sender multicasting to one DTC-2 receiver. Figure 4 shows multicast latency. Here, latency measures the amount of time required to asynchronously multicast a message of a given size to the multicast receiver and then synchronously receive a multicast response of the same size, divided by two. The curve in the graph is approximated by the equation $y = 0.001312x + 6.645$, where $y$ is latency in milliseconds and $x$ is message size in bytes.

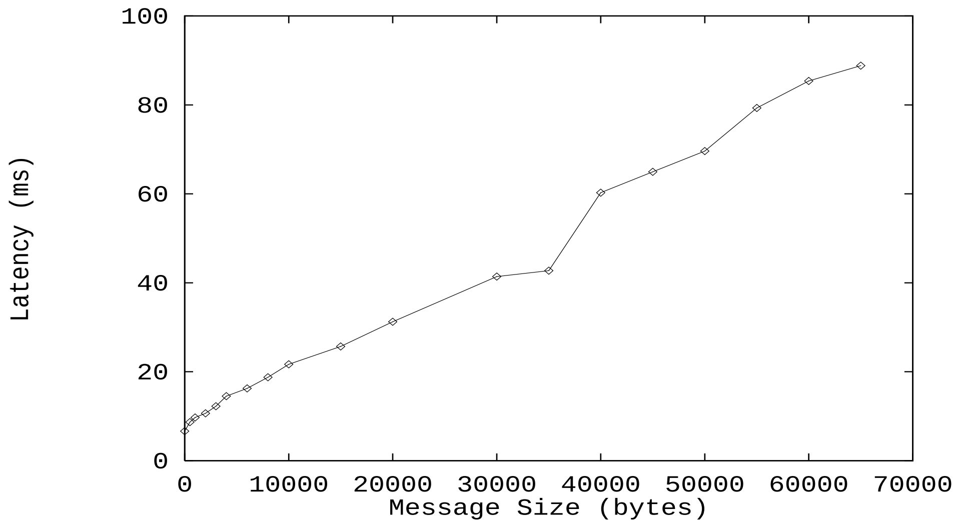


**Figure 4 — Multicast Message Size vs. Latency**

## 3. The Transport Level

All measurements were made by timing transfers from the top level interface of XTP. This is the interface on the off-board protocol processor used by the DTC kernel to communicate with the off-board processor.

### *Throughput*

Figure 5 shows the average throughput of a user at this interface. The message sizes on the *x*-axis represent user data only. The graph has an asymptote at approximately 21 megabits per second, the maximum transport-layer throughput.

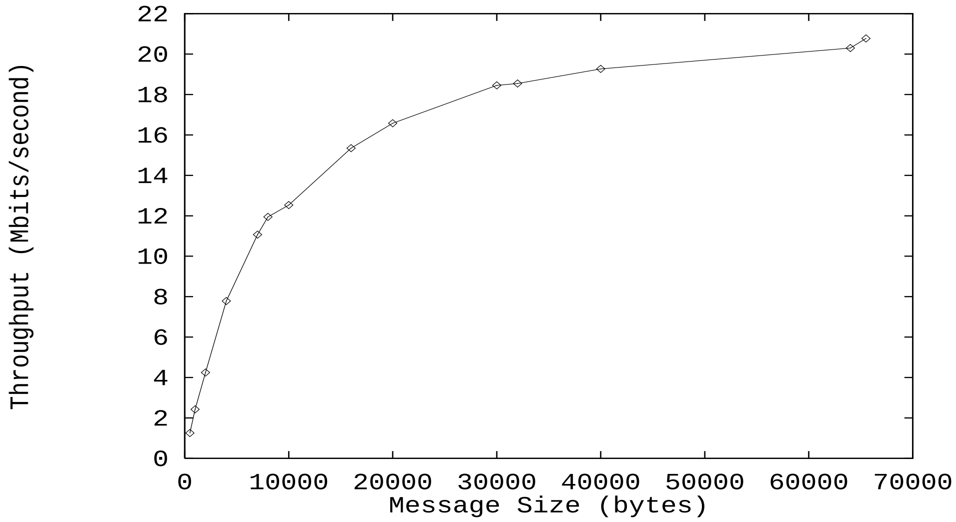


**Figure 5 — Message Size vs. Throughput**

***Latency***

Figure 6 shows the average message latency for a user at this interface. The message sizes on the *x*-axis represent user data only. The latency measures were calculated from an average roundtrip time for each message size. The average roundtrip times were produced by measuring the time to ping pong 1,000 messages between users. The curve is approximated by the equation $y = 0.000423x + 3.15$, where $y$ is latency in milliseconds and $x$ is message size in bytes.

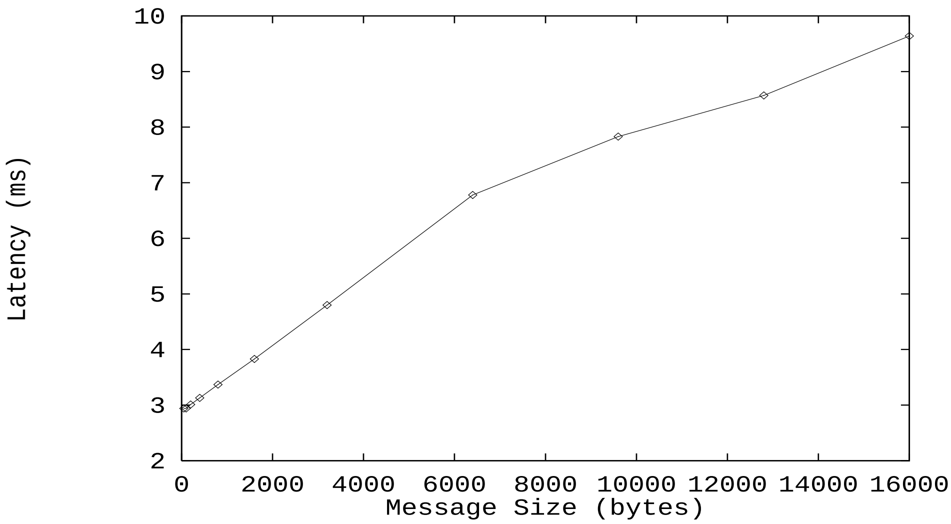


**Figure 6 — Message Size vs. Latency**

## 4. The MAC Level

All measurements were performed with no operating system running on the Motorola MVME 167 and each node housed in a stand-alone VME chassis. This is the interface used by XTP for communication with other hosts.

### *Throughput*

Figure 7 shows the average throughput of a user at this interface. The frame sizes on the *x*-axis represent user data only. The graph has an asymptote at 57 megabits per second, the maximum MAC-layer throughput.

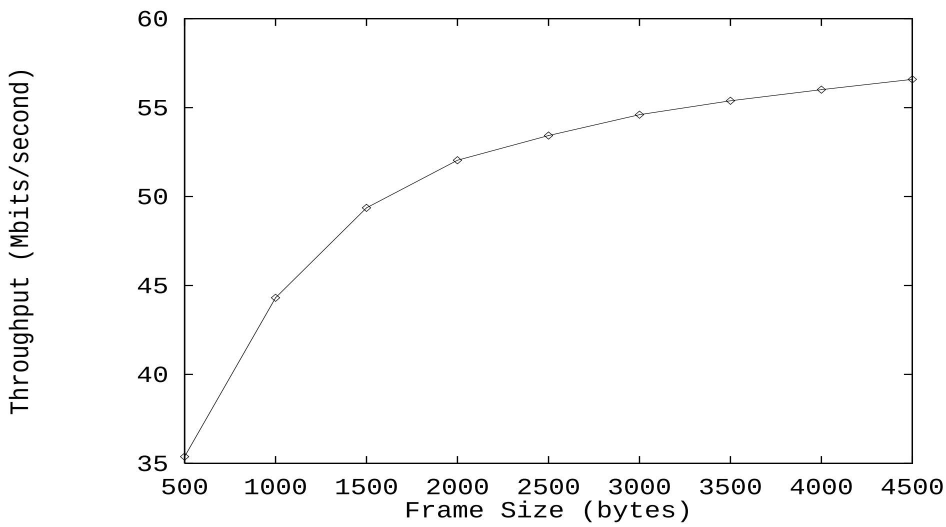


**Figure 7 — Frame Size vs. Throughput**

***Latency***

Figure 8 shows latency. The latency measures were calculated from an average roundtrip time for each frame size. The average roundtrip times were produced by measuring the time to ping pong 10,000 frames between MAC users. The curve is approximated by the equation $y = 0.000341x + 0.0915$, where *y* is latency in milliseconds and *x* is frame size in bytes.

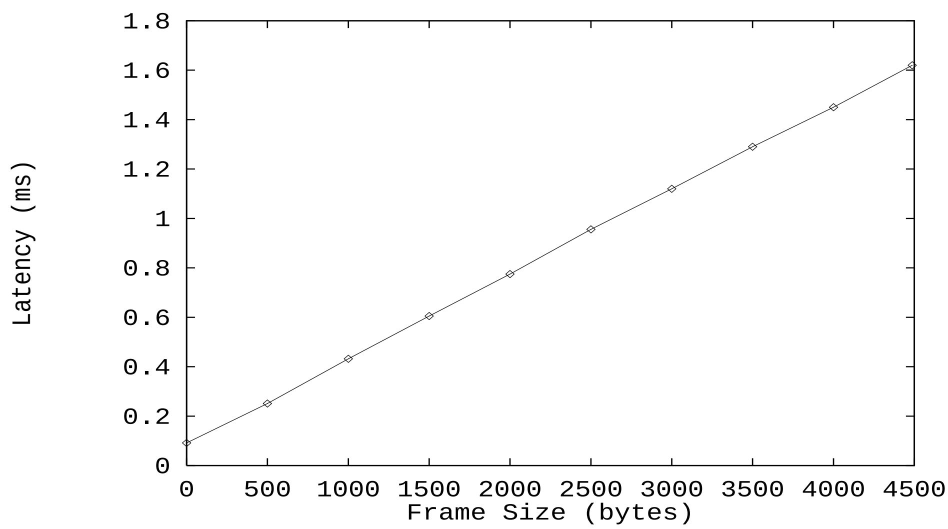


**Figure 8 — Frame Size vs. Latency**