Metrics for the Evaluation of Multicast Communications

Philip M. Irey IV and David T. Marlow

System Research and Technology Department, Combat Systems Branch
Naval Surface Warfare Center, Dahlgren Division, Dahlgren, Virginia 22448-5100
{ireypm, marlowdt}@nswc.navy.mil

Abstract. *In the distributed shipboard environment of interest to the United States Navy, there is an increasing interest in the use of multicast communications to reduce bandwidth consumption and to reduce latencies. Standardized metrics and tools are needed to evaluate the performance of multicast systems in this environment. This paper proposes a number of metrics and data collection and analysis techniques for assessing multicast communications performance. Of particular significance is a metric that correlates message reception among receivers and shows promise in analyzing topology related problems. The MCAST Tool Set (MTS), which uses the metrics and data collection techniques presented, is used to analyze several multicast application scenarios.*

1 Introduction

In the distributed shipboard environment, there are a number of data streams which must be received by multiple hosts. These data streams may be large in volume and sent often (e.g. 100 Hz gyro data). Multicast transfer techniques are expected to reduce aggregate bandwidth utilization and to reduce transfer latencies. As the number of recipients of these data streams grows in the shipboard environment, the need for utilizing multicast transfer instead of sequential data transfer to each recipient becomes obvious. Testing must be done to ensure that the performance of multicast implementations lives up to its expectations. A large body of work has been published on unicast metrics [1] and [2] which can provide a foundation for defining multicast metrics; however, multicast transfer breaks some of the fundamental assumptions of unicast metrics. The IETF has initiated efforts in defining metrics targeted at multicast exchanges [3] but little work has been done in this area.

2 Multicast Performance Metrics

This paper defines two types of metrics critical for assessing multicast performance: local metrics measured at a single sender or receiver and group metrics which represent an aggregate performance for all receivers in a multicast group. Local metrics are applicable to both unicast and multicast data transfer. Group metrics only apply to multicast transfer. All metrics defined are measured at the application to communications subsystem interface (e.g. the sockets API in a Unix environment) so delays introduced by transmission on the media, queueing delays, etc. are included in measurements based upon them.

2.1 Metric Notation

Metric names are defined with capital letters (e.g. *XYZ*). The subscript *i* is used to select a particular measurement from a set of measurements. The subscript *j* is used to select a particular host from a set of hosts. To represent a particular measurement on a particular host, a double subscript notation is used. For example, $LIAT_{j_i}$ represents *LIAT* measurement number *i* on host number *j*. Statistics can also be applied to a set of measurements. The subscripts avg, max, and min represent average, maximum, and minimum values. The subscript *sdev* represents standard deviation. To denote a statistical measurement from a particular host in group metrics, a subscript is applied to the statistic subscript. For example, XYZ_{avg_j} represents the average value of the *XYZ* metric on receiver *j*. In the equations below, *m* represents the number of messages sent on a data stream and *n* represents the number of multicast receivers in a group.

```
while (done==FALSE) {
    ts1=gettimeofday(ts1)
    send(A,++seq,ts1,data)
    ts2=gettimeofday(ts2)
    LIST[++LMS]=timediff(ts2,ts1)
    local_usleep(sleep_time)
}
```

FIGURE 1. Transmitter Pseudo-Code

2.2 Local Metrics

Local metrics are measured at either the transmitter or a receiver represented in the pseudo-code in Figures 1 and 2. Each message is sent to the address A which can address a unicast receiver or a multicast group of receivers. Each message sent contains a sequence number, seq, a timestamp, ts1, and data. Since the timestamp generated by the transmitter is used by receivers, it is assumed that either the clocks of the sender and all receivers are synchronized (e.g. NTP, GPS, etc.) or clock offsets can be computed [4].

```
bytes_received=LMR=LMRL=LGN=0
expected_seq_num=0
tr1=gettimeofday()
first=TRUE
while (done==FALSE) {
   prev_tr1=tr1
   recv(A,seq,ts1,data)
   tr1=gettimeofday()
   if (first == TRUE) {
      start_time=ts1
      FIRST=FALSE
   }
   process_packet = FALSE
   if (seq == expected_seq_num) {
      process_packet=TRUE
   }
   else if (seq < expected_seq_num) {
      ++LMRL
   }
   else { /* seq > expected_seq_num */
      LGB[++LGN]= (expected_seq_num,seq)
      LGL[LGN]=seq-expected_seq
      process_packet=TRUE
   }
   if (process_packet) {
      LOWL[++LMR]=timediff(tr1,ts1)
      LIAT[LMR]=timediff(tr1,prev_tr1)
      expected_seq_num=seq+1
   }
   bytes_received=bytes_received+sizeof(data)
}
end_time=gettimeofday()
LAT=bytes_received/timediff(end_time,start_time)
```

FIGURE 2. Receiver Pseudo-Code

2.2.1 LMS and LIST Metrics

The *LMS* (Local Messages Sent) and *LIST (*Local Inter-Send Time) metrics are both measured at the transmitter. All other local metrics presented below are measured at a receiver. *LMS* is the count of the number of messages sent to the address A by the transmitter. *LIST* metrics compute the time between successive message sends at the transmitter. Computing *LIST* is important because the measured value may not always correspond to the expected value [5]. *LIST* statistics are computed as shown in Equations 1-4

2.2.2 LMR, LMRL, LMRI, and LPMR Metrics

LMR_j (Local Messages Received) is the count of the number of messages received in sequence and those whose sequence number was greater than the sequence number expected for the next message. Messages received with a sequence number less than the expected one are counted in $LMRL_j$ (Local Messages Received Late). Messages which arrive out of order or are duplicate are classified as “late” using this criteria. $LMRI_j$ (Local Messages Received In-order) is defined to count only the messages which were received with a sequence number equal to the expected sequence number as shown in Equation 5. $LPMR_j$ (Local Percent Messages Received) computes the percentage of the messages sent by the transmitter which were received by a receiver as shown in Equation 6.

$$LIST_{avg} = \sum_{i=1}^{m} \frac{LIST_i}{m} \qquad (1)$$

$$LIST_{max} = \underset{\forall i}{MAX}\{LIST\} \qquad (2)$$

$$LIST_{min} = \underset{\forall i}{MIN}\{LIST\} \qquad (3)$$

$$LIST_{sdev} = \sqrt{\left(\sum_{i=1}^{m} (LIST_i - LIST_{avg})^2\right) / (n-1)} \qquad (4)$$

$$LMRI_j = LMR_j - LGN_j \qquad (5)$$

$$LPMR_j = LMR_j / LMS_j \qquad (6)$$

2.2.3 Local One-Way Latency (LOWL) Metrics

LOWL metrics measure the one-way latency between the transmitter and a receiver as shown in Equations 7-10. These metrics compute the difference between ts1and tr1. Messages which are late or lost do not contribute to these metrics since LMR_j is used.

$$LOWL_{avg_j} = \sum_{i=1}^{LMR_j} \frac{LOWL_{j_i}}{LMR_j} \qquad (7)$$

$$LOWL_{max_j} = \underset{\forall i}{MAX}\{LOWL_{j_i}\} \qquad (8)$$

$$LOWL_{min_j} = \underset{\forall i}{MIN}\{LOWL_{j_i}\} \qquad (9)$$

$$LOWL_{sdev_j} = \sqrt{\left(\sum_{i=1}^{LMR_j} (LOWL_i - LOWL_{avg_j})^2\right) / (LMR_j - 1)} \qquad (10)$$

2.2.4 Local Inter-arrival Time (LIAT) Metrics

LIAT metrics measure the time between the receipt of successive messages at a receiver. Since *LIAT* is computed using timestamps from the local receiver only, synchronized clocks or clock offset conversions are not necessary. Since $LIAT_1$ is always undefined, it is not used in computing

$$LIAT_{avg_j} = \sum_{i=2}^{LMR_j} LIAT_{j_i} / (LMR_j - 1) \qquad (11)$$

$LIAT_{avg}$ as shown in Equation 11. *LIAT* statistics are computed as in Equations 8-10 except that *i* starts at 2 and the $LMR_j - 1$ term in Equation 10 should be replaced with the term $LMR_j - 2$. Messages which are late or lost do not contribute to these metrics since LMR_j is used.

2.2.5 Local Application-to-Application Throughput (LAT) Metric

The *LAT* metric is used to characterize the end-to-end throughput between the transmitter and a receiver. The start of the time interval begins with the receiver recording the value of ts1 in the first message received on the data stream in start_time. The receiver then records the time when the last message was received in end_time. LAT_j is the number of bytes received divided by the difference between these two timestamps.

2.2.6 Local Gap Boundaries (LGB) Set

LGB_j is the set of ordered pairs of the start and end points of gaps in the sequence space of received messages observed by receiver *j*. LGB_j is a set from which metrics are derived for host *j*. When a message is received with a sequence number which is not equal to expected_seq, one greater than the highest previously received in-order message, one of two scenarios has occurred: 1) the message has a sequence number greater than expected_seq in which case the ordered pair (expected_seq_num,seq-1) denoting the sequence space gap is recorded in LGB_j; or 2) the message has a sequence number less than expected_seq in which case the message is considered late and no action associated with LGB_j is taken and a sequence space gap is not recorded

2.2.7 LGN and LGL Metrics

The *LGN* (Local Gap Number) metric is a count of the sequence space gaps observed (i.e. messages received which had a sequence number greater than expected_seq). The *LGL* (Local Gap Length) metrics measure the size of sequence space gaps observed by a receiver. LGL_{j_k} for sequence space gap LGB_{j_k} is computed by subtracting the ending sequence number for gap *k* from the starting sequence number as shown in Equation 12. (Figure 2 shows (start, end)-tuples being recorded for LGB). LGL_{avg_j} is computed by iterating over *k* as shown in Equation 13. LGL_{max_j}, LGL_{min}, and LGL_{sdev_j} are computed similarly to Equations 2-4 except that they are computed over *k*.

$$LGL_{j_k} = LGB(end)_{j_k} - LGB(start)_{j_k} \quad (12) \qquad LGL_{avg_j} = \left(\sum_{k=1}^{LGC_j} LGL_{j_k}\right) / LGN_j \quad (13)$$

2.3 Group Multicast Metrics

Unlike local metrics which characterize the performance of the transmitter or a receiver, group metrics attempt to simultaneously characterize the performance of all receivers in a multicast group.

2.3.1 GOWL, GIAT, GAT, GGN, and GGL Metrics

GOWL (Group One-way Latency) metrics characterize the one-way latency performance of the group using *LOWL* measurements from each receiver. *GOWL* statistics can be computed as shown in Equations 14-17 which serve as prototypes for computing statistics on sets for *GIAT*,

$$GOWL_{avg} = \sum_{j=1}^{n} \frac{LOWL_{avg_j}}{n} \quad (14) \qquad GOWL_{max} = \underset{\forall j}{MAX}\{LOWL_{max_j}\} \quad (15)$$

$$GOWL_{min} = \underset{\forall j}{MIN}\{LOWL_{min_j}\} \quad (16) \qquad GOWL_{sdev} = \sqrt{\sum_{j=1}^{n} (LOWL_{avg_j} - GOWL_{avg})^2 / (n-1)} \quad (17)$$

GAT, *GGN* and *GGL* defined below. *GIAT* (Group Inter-arrival Time) metrics characterize the message inter-arrival performance of the group using *LIAT* measurements from each multicast receiver. *GAT* (Group Application-to-application Throughput) metrics characterize the application-to-application throughput performance of the group using *LAT* measurements from each multicast receiver. *GGN* (Group Gap Number) metrics characterize the number of message gaps observed by the group using *LGN* measurements from each multicast receiver. *GGL* (Group

Gap Length) metrics characterize the size of message gaps observed by the group using *LGL* measurements from each multicast receiver.

2.3.2 Group Reception Correlation (GRC) Metric

The *GRC* metric is used to measure the degree of correlation in the messages received by members of a group. To compute *GRC*, reception vectors are created for each group member. A reception vector records which messages on a data stream were received in sequence and which were not by a particular multicast receiver in the group. To compute the reception vector for multicast receiver *j*, $\vec{V}_j$, component *i* of $\vec{V}_j$ is set equal to one if the message with sequence number *i* was received in order and is set equal to zero otherwise. The information needed to populate the reception vectors is recorded in *LGB*.

A set of reception vectors, *S*, can be grouped into a reception matrix, *R* as shown in Equation 18. Each column *j* of *R* contains the reception vector for multicast receiver *j*. Each row *i* of *R* contains a Reception Report (RR) for all receivers for message *i*. The number of rows in *R* is always equal to *m*. The number of columns in *R* is equal to the number of reception vectors in *S*. The reception matrix in Equation 18 shows *R* constructed from S^* (the set which contains reception vectors for all multicast receivers in a group). *R* can be constructed for any subset, *S*, of S^*.

$$R = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ V_{31} & V_{32} & \cdots & V_{3n} \\ \cdots & \cdots & \cdots & \cdots \\ V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix} \begin{array}{l} \leftarrow \text{Reception Report for Message \#1} \\ \\ \leftarrow \text{Reception Vector for Host \#2} \end{array} \qquad (18)$$

GRC is computed on a reception matrix *R* as shown in Equation 21 which is derived from Equations 19 and 20. Equation 19 computes the Message Reception Correlation (*MRC*) for message *i*. In this equation, the sum of R_{ij} is the number of multicast receivers which received message *i* (R_{ij}=1) and *n* minus this sum is the number which did not (R_{ij}=0). The absolute value of the difference of these two quantities is then divided by *n* (the number of multicast receivers) which yields a value between 0 and 1. This value represents the degree of correlation among the multicast receivers in receiving message *i*. A degree of correlation equal to 1 indicates that all receivers in the group whose reception vectors are contained in *R* received that message or all of them did not receive that message. A degree of correlation equal to 0 indicates that half of the receivers received the message and the other half did not. The sum of the *MRC* values is computed and divided by *m* (the total number of messages which could have been received) to give the average degree of correlation for all messages, or *GRC*, as shown in Equation 20. The values of *GRC* range from 0 to 1.

$$MRC_i = \left| \left(\sum_{j=1}^{n} R_{ij} \right) - \left(n - \sum_{j=1}^{n} R_{ij} \right) \right| / n \qquad (19)$$

$$GRC = \sum_{i=1}^{m} \frac{MRC_i}{m} \qquad (20)$$

3 Testing Considerations

The utility of the metrics defined in Sections 2.2 and 2.3 are dependent on the test environment. In the environment of interest in this paper, IP multicast data transmission is used which has unreliable semantics which can affect any measurements collected.

3.1 Correlation and Cross Checking of Group Receivers

$$GRC = \left(\sum_{i=1}^{m} \left| 2 \sum_{j=1}^{n} R_{ij} - n \right| \right) / (n \times m) \qquad (21)$$

The *GRC* computed for the group provides a measure of the consistency of data reception among the set of multicast receivers. If there is a low degree of correlation among the group members, it is unlikely that the receivers are making their measurements on the same set of samples. As an extreme example, suppose Host 1 receives all messages with odd sequence numbers and Host 2 receives all messages with even sequence numbers which were sent on the same datastream to a multicast group. In this case, Host 1 and Host 2 are making measurements on what can be viewed as two different sets of data. In this case, *GRC* is equal to zero. Non-gap group metrics should be viewed with caution in this case. Cross-checking of the data measurements can clarify what appears at first glance to be inconsistent results. For example, it is possible to measure a low value for *LOWL* and a high value for *LIAT* on a data stream. This seems inconsistent since the measured *LOWL* implies good network performance yet the *LIAT* implies poor network performance. Examining *LGC* might show a large number of gaps which would contribute to the large value for *LIAT*. *LOWL* is still low since the time interval used for that metric only uses timestamps associated with that message. The *LIAT* metric, on the other hand, uses a timestamp taken when a previous message was received on the data stream.

3.2 GRC Partitioning Algorithms

Two partitioning algorithms are defined to partition hosts into groups based on the *GRC* metric computed for those groups. First, a threshold value is specified. Next, groups of hosts whose *GRC* is greater than or equal to the threshold value are created. Loose Partitioning creates a set Π to partition the set of receiving hosts into subsets based on their RR's such that the GRC for each subset is greater than or equal to a threshold correlation, τ as shown in Equation 22 and 23. Strict partitioning creates a set $\widehat{\Pi}$ to partition the set of receiving hosts into subsets based on their RR's such that the GRC for each subset and all subsets of that subset is greater than or equal to a threshold correlation, τ as shown in Equations 24 and 25. It should be noted that there are 2^n subsets which can be generated from R (where $n = |R|$) so generating $\widehat{\Pi}$ can be an expensive operation.

$$P = \{s \subseteq R | GRC(s) \geq \tau\} \qquad (22)$$

$$\Pi = \left\{ \begin{matrix} A \subseteq P | A \text{ is not a proper subset of} \\ \text{any other element of } P \end{matrix} \right\} \qquad (23)$$

$$P = \{s \subseteq R | \forall T, T \subseteq s, GRC(T) \geq \tau\} \qquad (24)$$

$$\widehat{\Pi} = \left\{ \begin{matrix} A \subseteq P | A \text{ is not a proper subset of} \\ \text{any other element of } P \end{matrix} \right\} \qquad (25)$$

4 The MCAST Tool Set (MTS)

MTS was developed at the Naval Surface Warfare Center - Dahlgren Division to enable the performance analysis of UDP/IP multicast systems using the metrics defined in this paper. It used in Section 5 for instrumentation and analysis. There are four types of components in MTS: an instrumentation tool called MCAST (Multicast Communications Analysis and Simulation Tool), a test coordination tool, a data extraction tool, and several data analysis tools. Although these tools can be used individually, they are generally used together. (Contact the authors for information on obtaining MTS).

5 IP Multicast Performance Tests

Several tests were conducted to evaluate the performance of UDP over IP multicast using the metrics developed in Section 2. All testing was done using MTS on the testbed shown in Figure 3. The hosts are interconnected with 10/100/1000 Mbps Ethernet switches from Extreme Networks. It is important to note that multicast routing is not being used in the testbed. In this environment, all multicast traffic is forwarded across all collision domains.

5.1 Unicast Versus Multicast

As described in Section 2, local metrics are applicable to measuring both unicast and multicast performance. MCAST allows both unicast and multicast measurements to be made using these metrics. In these tests, Tide is the transmitter and Cheer is the receiver. First a unicast data stream is measured between the hosts. Next, a multicast data stream is measured between the hosts. The unicast and multicast *LET* and *LOWL* results are then compared.

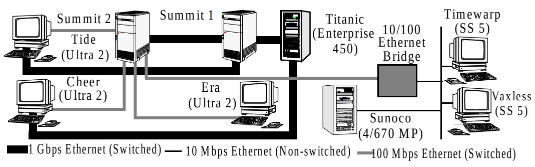

FIGURE 3. Multicast Testbed

5.1.1 100 Mbps Switched Ethernet

As can be seen in Figures 4 and 5, there is little if any difference between unicast and multicast transmission using 100 Mbps switched Ethernet with Tide as the transmitter and Cheer as the receiver.

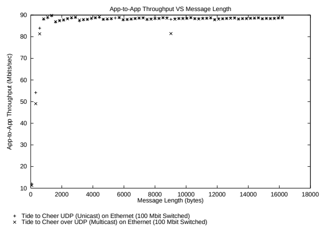

FIGURE 4. Unicast Versus Multicast LAT over 100 Mbit Switched Ethernet

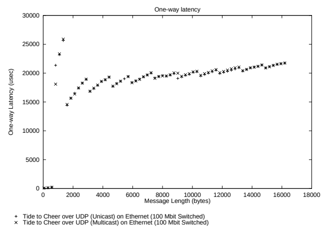

FIGURE 5. Unicast Versus Multicast LOWL over 100 Mbit Switched Ethernet

5.1.2 One Gigabit/Second (Gbps) Switched Ethernet

As shown in Figures 6 and 7, there is a noticeable performance difference in the *LAT* and *LOWL* metrics measured with Tide as the transmitter and Cheer as the receiver over Gigabit Ethernet. No satisfactory explanation for the performance differences observed between unicast and multicast transmission in the one Gbps tests has been found yet. Since no differences were observed in the 100 Mbps tests and differences were observed in the one Gbps tests, one might conclude that the multicast forwarding capabilities of the switch do not scale in the same way as the unicast forwarding at higher transmission rates or that architectural differences between the Summit 1 and Summit 2 switches used in the tests contribute to the performance differences observed. Further work in this area will attempt to test these theories and attempt to identify the source of the performance differences.

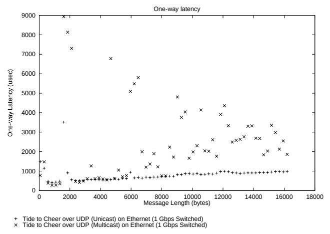

FIGURE 6. Unicast Versus Multicast LOWL over Gigabit Ethernet

5.2 Group Multicast Performance

When analyzing the behavior of a multicast group, the *GRC* metric is often a key in the analysis process. Figures 8, 9, and 10 show experiments, E1, E2, and E3, in which sunoco transmitted 10,000 messages to era, tide, timewarp, and vaxless. As shown in Figure 3, the transmitter (sunoco) uses a 10 Mbps shared Ethernet connection to communicate with the receiving hosts and thus can not exceed a transmission rate of 10 Mbps.

Figure 8 shows the results of E1 where sunoco sent messages of length 16384 bytes to the multicast group with a sleep_time equal to 0. As can be seen visually, few messages were considered lost by any of the receivers. Table 1 shows a high degree of correlation (.9991) among the receiver set and that a high percentage of the messages were received (99.87%).

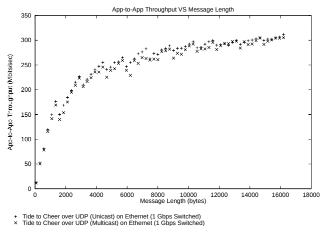

FIGURE 7. Unicast Versus Multicast LAT over Gigabit Ethernet

Figure 9 shows the results of E2 where sunoco sent messages of length 16384 bytes to the multicast group with a sleep_time equal to 0. In this scenario, unlike E1 and E3, a unicast background load of 6 Mbps (between two hosts not involved on the multicast test) was introduced on the 10 Mbps Ethernet. As can be seen, in Figure 9, a large number of messages were dropped. Table 1 shows a high degree of correlation among the receiver set (0.9989). It also shows that nearly all the messages which were considered lost were considered lost by all receivers (RR=0). In this environment, a message considered lost by all hosts probably indicates that the message was lost at the sender. It is likely the background load on the shared media to which the transmitter is attached caused large numbers of packet collisions which resulted in queue overflows on the transmitter.

Figure 10 shows the results of E3 where sunoco sent messages of length 256 bytes to the multicast group with a sleep_time equal to 0. For E3, Table 1 shows a much lower degree of correlation (.04498) than E1. This is also reflected in the Reception Reports (RR) in the table. Computing Π for E3 with $\tau = 0.5$ yields the set $\{\{R_1, R_2\}, \{R_3, R_4\}\}$ where R_1, R_2, R_3, R_4 are the reception reports for Tide, Era, Vaxless, and Timewarp respectively. Consequently, the high performance machines, Tide and Era, correlate to the specified degree as do the low performance machines, Vaxless and Timewarp.

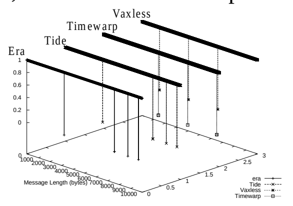

FIGURE 8. E1: Reception Vectors for Message 16384 Bytes in length (no background load)

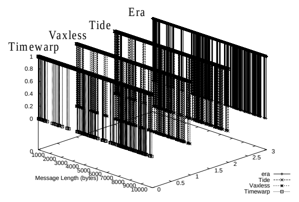

FIGURE 9. E2: Reception Vectors for Messages 16384 bytes in length (6 Mbps background load)

6 Conclusions

The metrics and analysis techniques developed in this paper as well as their realization in MTS proved to be useful in analyzing the performance of multicast systems. The performance of complex multicast applications can be simulated and evaluated on the target hardware environment without the complexity of running the actual applications. Multicast performance variations among the set of receivers related to network topology, network utilization, and host performance were observed with MTS. In particular, the GRC metric proved to be extremely useful for analysis. Without the metrics defined here and their implementation in MTS, analyzing performance in a distributed multicast application would be very difficult.

TABLE 1. GRC Measurements

	GRC	*RR=0*	*RR=N*	*0 < RR < N*
E1	0.9991	0%	99.87%	.13%
E2	0.9989	1.81%	97.99%	0.2%
E3	0.4498	0%	21.06%	78.94%

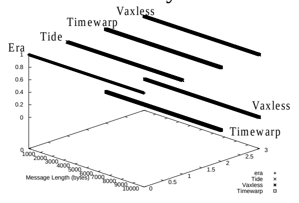

FIGURE 10. E3: Reception Vectors for Messages 256 bytes in length (no load)

7 Future Work

The work to date has focused on instrumenting IP Multicast in a non-routed environment. Future work in this area will examine a routed multicast environment and a switched VLAN environment. Further investigation is planned to investigate the Ethernet performance differences observed. Investigating applying multicast techniques and using metrics defined here in a network resource monitor which overcomes the scalability problems cited in [6] is planned.

References

1. Irey IV, Philip M., Harrison, Robert D., Marlow, David T., *Techniques for LAN Performance Analysis in a Real-Time Environment*, Real-Time Systems - International Journal of Time Critical Computing Systems, Volume 14, Number 1, pp. 21-44, January 1998.†

2. Paxon, V., Almes, G., Mahdavi, J., Mathis, M., *Internet Draft: Framework for IP Performance Metrics*, November 1997.

3. Dubray, K., *Internet-Draft: Terminology for IP Multicast Benchmarking*, July 1997.

4. Mills, D., *RFC-1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis*, March 1992.

5. Irey IV, Philip M., Marlow, David T., Harrison, Robert D., *Distributing Time Sensitive Data in a COTS Shared Media Environment*, Joint Workshop on Parallel and Distributed Real-Time Systems, pp. 53-62, April 1997.†

6. Irey IV, Philip M., Hott, Robert W., Marlow, David T., *An Architecture for Network Resource Monitoring in a Distributed Environment*, Lecture Notes in Computer Science 1388, pages 1153-1163, Springer-Verlag, 1998†.

†Documents available on http://www.nswc.navy.mil/ITT/documents.