Metrics, Methodologies, and Tools for Analyzing Network Fault Recovery Performance in Real-Time Distributed Systems

P. M. Irey IV, B. L. Chappell, R. W. Hott, D. T. Marlow,
K. F. O'Donoghue, and T. R. Plunkett

System Research and Technology Department
Combat Systems Branch
Naval Surface Warfare Center, Dahlgren Division
Dahlgren, Virginia 22448-5100, U.S.A

Abstract. The highly distributed computing plants planned for deployment aboard modern naval ships will serve real-time, mission-critical applications. The high-availability requirements of the distributed applications targeted for these platforms require mechanisms to rapidly recover from system faults (e.g., battle damage). Providing network fault recovery mechanisms that support rapid recovery in the network infrastructure used to compose these systems is critical. This paper examines how metrics used for general network performance analysis can be used to analyze fault recovery performance. A methodology for applying these metrics is presented. A testing toolset that implements the metrics and complies with the testing methodology is presented. Finally, test data collected using the toolset is presented to show the utility of the metrics and testing methodology for evaluating network fault recovery performance.

1 Introduction

Distributed computing systems composed of hundreds of processors interconnected by a network infrastructure are planned for deployment aboard modern naval ships. Applications required to enable the ship to carry out its missions and to keep the ship at sea (e.g., navigation, steering, sonar processing, command and control, etc.) are assigned to processors within the distributed computing system. A wide range of performance requirements exist for these applications. The real-time mission critical nature of the processing performed in the distributed computing system requires a fault-tolerant architecture as a key design element. An important aspect of this architecture is that the performance requirements of the system must be maintained even during system faults.

This paper focuses on a key performance requirement of the fault-tolerant distributed computing architecture: network fault recovery performance. Fault tolerance is an important concern in the design of the network infrastructure since it interconnects all of the processors in the distributed system. We use the term network fault recovery performance to refer to the time required to detect and recover from a network fault (e.g., a broken cable or connector, a failed interface adapter, a failed network switch, etc). If the maximum time required to perform a network fault recovery can be bounded and the system performance requirements are still met at

this bound, then that network fault recovery mechanism can be used as the foundation in the hierarchical fault-tolerant architecture of the distributed computing system. Naval anti-air warfare systems have network fault recovery requirements on the order of hundreds of milliseconds. This requirement is several orders of magnitude more stressing on the network design than requirements typically found in today's commercial systems.

In this paper, we use metrics defined previously [IREY97, IREY98] for measuring network fault recovery performance in a generalized networking environment. Next, the paper defines a testing methodology appropriate for the real-time distributed computing environment based on these metrics. The paper concludes by providing the results of measurements made by a toolset developed by the Naval Surface Warfare Center, Dahlgren Division, for evaluating network fault recovery performance using the metrics and testing methodology defined.

2 Network Fault Recovery Technologies

In the initial efforts to use Commercial off the Shelf (COTS) networking products aboard Navy ships, the standard COTS Fiber Distributed Data Interface (FDDI) [FDDI] technology was used. Complex FDDI-based architectures were developed which supported the loss of multiple interconnects. To evaluate the fault recovery time provided by a highly survivable FDDI architecture, [HILES95] used the FDDI Station Management (SMT) [SMT] standard to define a set of network fault recovery time metrics to compute the theoretical FDDI minimum fault recovery time. The fault recovery times measured varied widely from just under 100 milliseconds to over 500 milliseconds, with typical values in the 200 millisecond range. The paper alerted the FDDI community to the fact that a large variance in fault recovery performance was to be expected in COTS implementations and that the performance varied by product.

As the Navy moves from FDDI towards follow-on COTS solutions, new network fault recovery approaches will be used. Each approach considered must be rigorously tested to evaluate its fault recovery performance. FDDI is the only COTS networking technology that incorporated dual-attachment functionality within its defining standards. Vendors are supplying proprietary versions of fault recovery capabilities in other technologies (e.g., Ethernet); however additional testing is needed to ensure that the fault recovery behavior of these mechanisms is understood. The metrics used in this paper for evaluating network fault recovery performance are defined in general terms and thus may be applied to any type of networking technology.

3 Network Fault Recovery Performance

Network fault recovery performance is used to evaluate the ability of a given networking technology to detect and recover from network faults such as a broken cable or a failed network switch. Network fault recovery performance is a critical design parameter for systems such as the mission critical distributed computing systems of interest to the United States Navy.

Previous characterizations of FDDI fault recovery performance were obtained through the use of either theoretical analysis or hardware specific measurement techniques [HILES95] [RALPH]. While these approaches are valid for a technology such as FDDI, they are not general enough to evaluate technologies such as Fast

Ethernet where no standardized fault recovery mechanisms exist. Although work is ongoing to standardize some of these fault recovery mechanisms [802.1D], all implementations today are proprietary and may or may not interoperate. [HUANG] defines fault recovery performance metrics for a fault-tolerant Ethernet architecture based on specific fault-tolerant middleware.

We propose that the network fault recovery performance of a component, system, or architecture should be characterized by a set of metrics rather than as a single number. The metrics we propose are independent of specific hardware capabilities or network architectures. They allow network fault recovery performance to be evaluated from an application point of view. While these metrics have been used before in other contexts [IREY97] [IREY98], their application to the evaluation of network fault recovery performance and the measurement methodology required to obtain accurate measurements for this testing domain is unique. Little has been published on the evaluation of network fault recovery performance which is independent of any particular networking technology or concept.

3.1 Testing Model

The network fault recovery performance metrics used in this paper are based on a constant stream of test data to a receiving host through a network infrastructure. In a fault-tolerant environment, it is likely that the transmitting and receiving hosts are interconnected using redundant network connections and paths. These details, however, are transparent to the metrics we use to evaluate network fault recovery performance.

To measure network fault recovery performance, the messages in the test data stream are transmitted at a constant rate. The offset in time between when a message is received and when it was transmitted reflects the latency incurred in delivering the message. At a controlled point during the test, one or more faults are injected into network infrastructure connecting the transmitter and receiver. The faults cause perturbations in the normally regular data stream as observed at the receiving host (e.g., lost messages, increased message latency, etc.).

3.2 Fault Recovery Performance Metrics

There are several metrics used to describe the performance of a fault recovery mechanism in a network. These metrics are inter-send time, inter-arrival time, one-way latency, percent data received, and number of duplicates. Inter-send time and inter-arrival time are measures of the time between the sending of two consecutive packets at the transmitting host or the receiving of two consecutive packets at the receiving host respectively. One-way latency is the measure of the time period between sending a packet at the transmitting host and receiving that packet at the receiving host. Percent data received is the number of received packets divided by the number of transmitted packets. Finally, the number of duplicate packets is the number of packets received that are in excess of the number sent. All of these metrics can be combined to give a comprehensive picture of the fault-recovery performance.

4 Testing Methodology

A testing methodology must be defined specific to this problem domain. The testing methodology includes a calibration test procedure for tuning test runs for

1) Select a test message size, S.
2) Minimize Is_i so that $PDR = 100\%$.
3) Maximize Th so that $M(Ev) - E(Ev)$ approaches or equals zero for all metrics of interest.
4) Select N so that $(Is_i * N) >> F$.
5) Run test measurement with computed Is_i, Th, N, and S.

Figure 1: Calibration Test Procedure

maximum accuracy and a high-fidelity testing procedure. At first glance, it may appear that a test tool as simple as Ping [KESSLER] would suffice to measure the network fault-recovery performance. Tools such as this are not adequate, however, as they lack the fidelity to make accurate measurements for the real-time distributed computing environment of interest. Also, many of these tools use transaction based measurements rather than one-way measurements which again reduces the fidelity of the measurements as shown in [IREY98].

Figure 1 defines a calibration test procedure for tuning the measurement process to maximize the accuracy of network fault-recovery performance measurements. Several new terms are used in the calibration test procedure: S specifies the size in bytes of each message sent on the test data stream during a test; Th specifies a measurement threshold used to filter out non-event related measurements; N specifies the number of messages transmitted on the data stream during a test; and F specifies the time elapsed between the injection of the first failure into the network and when the network has recovered completely from all injected failures. It is expected that multiple test runs will be conducted to select values for the parameters in each of the steps of the calibration test proceedure. When a value is selected for a given step, additional test runs should be conducted to ensure that the new value selected didn't unexpectedly impact values previously selected for other steps.

Tests of network performance by the authors (e.g., [IREY98]) in the real-time distributed environment of interest to the Navy has shown that the testing methodology used must support high-fidelity measurements. To satisfy this requirement, the network fault recovery performance tests must: 1) gather a large number of test samples to determine the range of performance; 2) examine the maximum and minimum (e.g., worst case) values rather than only mean values with standard deviations; and 3) provide visualization tools which allow the large data sets to be analyzed and iteratively reduced.

5 Network Fault Recovery Performance Measurement Toolset

A toolset was developed to measure network fault recovery performance using the metrics and testing methodology defined here. The tools in the toolset fall into three classes: test orchestration tools, data collection tools, and analysis/visualization tools. The relationships among these tools are shown in Figure 2.

5.1 Test Orchestration Tools

The main test orchestration tool in the toolset is called *nettest*. The main functions of *nettest* are: 1) to activate the data collection tools (e.g., transmitters and

receivers) on specified nodes; 2) to initiate the test data stream at a specified time; 3) to inject faults in the network at specified times; 4) to gather results from the experiment; and 5) to repeat the process from step 1 until a specified number of iterations have been completed. The *nettest* program is generally run on a control host which is usually a system other than the sending and receiving hosts involved in an experiment. The complete set of results gathered by *nettest* is passed to the analysis/visualization tools.

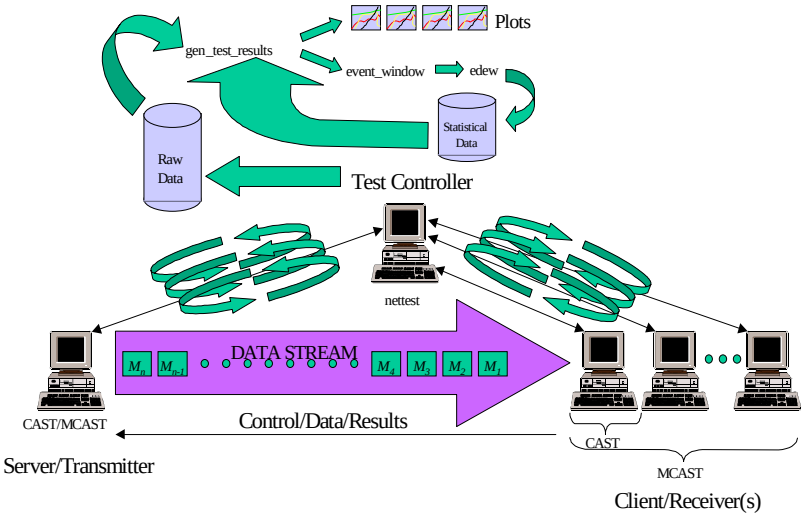

Figure 2: Network Fault Recovery Performance Toolset

One of the unique features of *nettest* specifically related to the measurement of network fault recovery performance is its fault injection capability. A scripting language was developed for *nettest* which allows the user to invoke actions on network components and to specify faults to be injected to specific network components (e.g., cables, switches, etc.) at specified times. When the power is removed from a media converter, it appears to the network components or hosts connected by the media converter that a cable has been broken. This provides one mechanism that enables automated testing needed for performing numerous tests which enables the high-fidelity testing previously described. A large library of reusable test scripts has been developed for testing failure scenarios for a variety for fault-tolerant network architectures.

For each test iteration, nettest first runs a user provided reset script to force the components under test into a known state. Next, the test data stream is started in parallel with a fault injection script which injects faults into specified network components at specified times. We have found that very complex reset scripts may be needed for particular architectures to be tested. It is best to assume that the network infrastructure is in an unknown state and to reset the state to a known state before a test run is conducted. Events external to a particular test procedure (e.g., a previous test) may have left the network infrastructure in an undesirable state.

5.2 Data Collection Tools

The main data collection tool used in the toolset is called *CAST* (Communications Analysis and Simulation Tool). *CAST* was developed by the authors to support end-to-end network performance analysis for unicast communications. It gathers a large number of performance metrics of which the network fault recovery performance metrics presented in Section 3 are a subset. *CAST* performs preliminary filtering on the data using Th defined in Section 3.

5.3 Analysis/Visualization Tools

To decrease the time required for a user to analyze various *CAST* test runs, a set of analysis and visualization tools were developed. The *nettest* tool generates large data files containing the results of iterative *CAST* test runs. In most cases, the data is best interpreted through visual plots. A tool was developed called *gen_test_results* that extracts the data of interest from the raw data files and then plots the extracted data.

To aid in reducing the data gathered during a network fault recovery test (i.e., analysis of events), two other tools were developed, *edew* and *eventwindow*. *Edew* computes statistical functions on the set of recorded events, such as the mean and standard deviation. *Eventwindow* filters the data from a network fault recovery test to obtain the event data. During the analysis of a network fault recovery test, these two tools are used to process and format the data. The output of these tools is provided as input to the *gen_test_results* tool.

6 Applying the Metrics, Tools, and Testing Methodology

The previous sections described the metrics, the tools that are used to measure those metrics, and some methods of how those tools can be used to perform an experiment. This section demonstrates the utility of the metrics in evaluating the performance of a survivable network by examining experimental data collected.

6.1 General Test Setup

This testing methodology is applied to both an FDDI based network and a survivable Fast Ethernet based network. The FDDI network consists of two host machines and a FDDI concentrator. One of the hosts is dual-homed to the FDDI concentrator. This is achieved by connecting both ports of a dual-attached FDDI NIC, which is installed on the host, to two ports on the FDDI concentrator. The second host uses only one of the ports on its installed FDDI NIC. The dual-homed machine serves as the transmitting host during the test, while the single-homed machine performs the function of the receiving host.

In the Fast Ethernet test, two host machines are used in conjunction with two Fast Ethernet switches. Each network host is dual-homed to each of the two Ethernet switches. The two dual-homed interfaces on the hosts may be on the same Network Interface Card (NIC) or may be on multiple NICs depending on the type of network fault recovery scheme being used. In either case, both network ports appear as a single network interface to the applications running on the network host. The two Ethernet switches are directly interconnected by one or more Fast Ethernet links.

6.2 Example FDDI Test Results

The results of performing a network fault recovery performance test on FDDI are located in the plots in Figure 3. The plot on the left shows the distribution of inter-arrival times measured throughout the length of the test. Notice that a majority of the inter-arrival times occur around the expected value of 200 milliseconds, which agrees with the results found by other measurement techniques [HILES95]. All other values of inter-arrival times appear to be well below this range. It is difficult to determine the significance of these values until the plot located on the right in Figure 3 is examined. In this plot, the inter-arrival times are examined with respect to the sequence numbers of the packets being received. This shows at what point in the test that the events occurred. Notice that this plot shows there are two distinct sets of events occurring during the test. This result agrees with the expectation that two events occur during each iteration of the test. These expected events are 1) the primary port on the transmitting host machine is brought down, and 2) the primary port is subsequently brought back up again.

The second event is expected since FDDI has one port that is the default when both ports are active. Other fault tolerant solutions including some of the fault tolerant Fast Ethernet solutions do not exhibit this behavior because they do not have

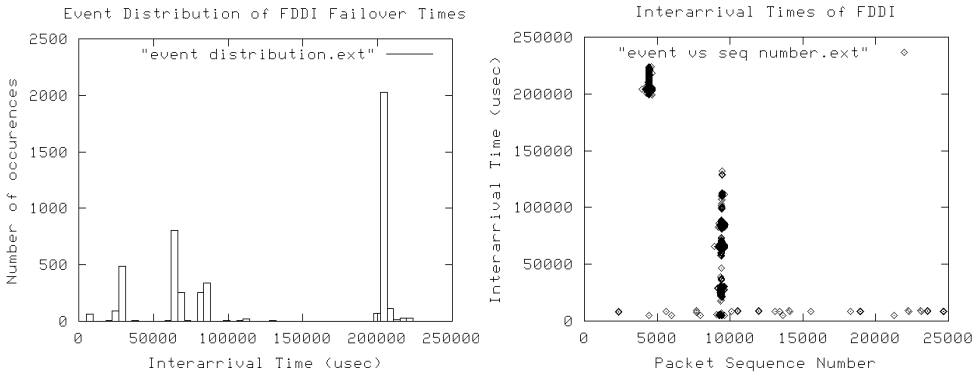

Figure 3: Measured FDDI Interarrival Times

a default port. In this case a failover only occurs if the active port goes down..

6.3 Example Fast Ethernet Test Results

The results of performing network fault recovery performance tests on Fast Ethernet are located in figures 4-6. The first set of data represented by the left hand plot of Figure 4 shows measured inter-send times in a test where the inter-send interval has been set to 30 milliseconds. Notice that most of the values are very near the 30-millisecond setting. The outlying values are useful in determining the threshold for the inter-arrival metrics in the absence of network failures. For instance, if an inter-arrival time above 35 milliseconds is considered a network fault, then it is likely that one network failure would be erroneously detected due to the one inter-send value above 37 milliseconds. Normally the variance in the inter-send values will be taken into account if the calibration test procedure described in Section 4 has been performed properly. However, it is wise to check the inter-send data from a test to

insure that the test results are due to the performance of the network and not a result of an anomaly on the transmitter.

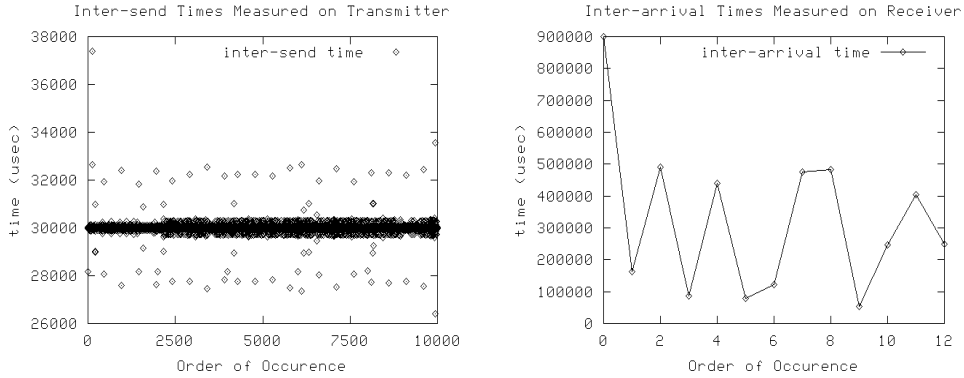

Figure 4: Measured inter-send and inter-arrival times

The next few sets of data shown in the right hand plot of Figure 4 and both plots of Figure 5 are from a test in which one of the two Ethernet switches is powered off and the clients have to fail over to the second Ethernet switch. The right hand plot of Figure 4 shows the inter-arrival times measured during this particular test. The data set indicates that the network fault recovery performance of this survivable Ethernet solution is similar to the fault recovery time of FDDI. All the network fault recovery times are less than one second, and all but one of the measurements are below 500 milliseconds. If this were the only type of measurement made during the test, as was the case with [HILES95], then some very undesirable network behavior would have been undetected.

The left hand plot of Figure 5 shows the percent of the transmitted data which was received by the receiving host. Since there is a period of time in which the network is inaccessible due to the powering down of the Ethernet switch, one would expect that only a fraction of the total messages would make it to the destination. Instead more messages were received than were transmitted (approximately 1.5 times as many). This indicates that messages are being duplicated by the network fault recovery process. The data set shown in right hand plot of Figure 5 confirms this. This data set shows the actual number of duplicate messages that have been received. In this case, conclusions drawn from the data in the left hand plot of Figure 5 lead to a closer examination of the data in the right hand plot of Figure 5. The order of magnitude of the number of duplicate messages is important to the results of this test. Even if each individual receiving host knows to ignore duplicate packets, the health

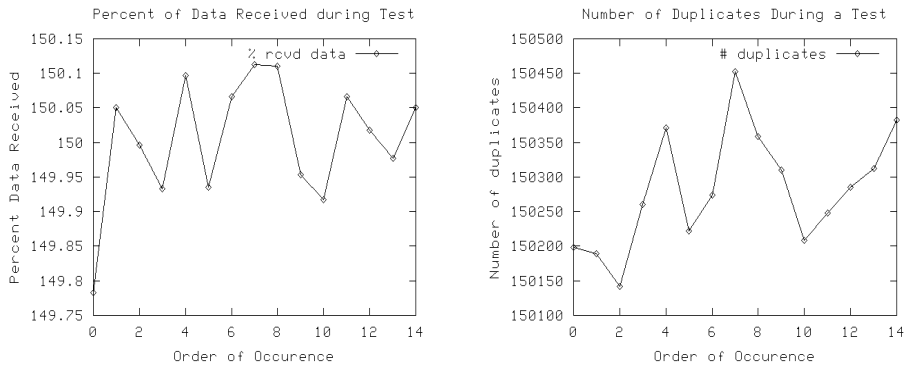

Figure 5: Percent data received and received duplicate packets

of the network as a whole may be in danger. If the switches used in this particular survivable Ethernet configuration had been fully populated with clients, then the duplicate packets could have saturated the bandwidth available on the switches, thereby eliminating any chances of recovering from the network failure in a timely manner. Further analysis would be needed to pinpoint the source of the duplicate packets (e.g., transmitting host NIC, switch, etc.)

The plots in Figure 6 show the importance of a high-fidelity testing methodology for measuring network fault recovery performance. The plot on the left shows measurement of inter-arrival times obtained by manually inserting network faults into the test. After performing the measurement five times, it appears that the network fault recovery time is consistently around 400 milliseconds. The right hand plot of Figure 6 shows the results an automated, high-fidelity measurement made using *nettest*. The network fault recovery performance measured using this testing methodology shows that the network fault recovery times can vary within a broader range of 300 to 800 milliseconds. This shows why an automated, high-fidelity testing methodology is preferable since many more iterations are possible. This leads to a better view of the performance being obtained. Typical *nettest* runs last for hours or days with hundreds or thousands of test iterations.

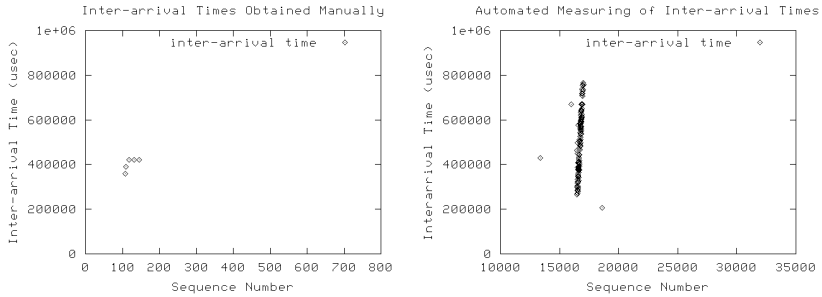

Figure 6: Interarrival times measured using manual and automated techniques

7 Conclusions and Ongoing Work

This paper presents a number of metrics for evaluating network fault recovery performance and a testing methodology for applying these metrics. The utility of the application of these metrics and the testing methodology is shown through a number of example experiments. Unlike other network fault recovery metrics which have been defined, the metrics defined here allow network fault recovery performance to be measured in a manner which is independent of the characteristics of the networking components used or the architecture used to interconnected these components. The benefits of using the automated, high-fidelity testing methodology via the capabilities provided by the fault recovery performance measurement toolset are shown. The practical limitations of manual fault injection (e.g. on the order of 10 iterations) that lead to low-fidelity measurements is contrasted with the high-fidelity

measurements possible using the automated fault injection capabilities of the *nettest* test orchestration tool to run hundreds or thousands of test iterations.

Experiments are ongoing to evaluate components to provide a fault tolerant networking alternative that uses readily obtainable COTS components rather than FDDI components that will no longer be produced. The experiments are looking at the network fault recovery performance of Network Interface Cards (NICs) and switching components as well as architectures based on these components. Seven network architectures have been identified and testing on them is in-progress. A number of failure scenarios have been developed which are applicable to one or more of these architectures. The network fault recovery performance toolset presented in Section 5 will be used to perform these experiments based on the scenarios developed. In addition to helping the Navy transition to a new approach for configuring shipboard networks, the lessons learned through this use of the toolset will be applied to improving these tools and testing methodologies.

8 References

[802.1D] IEEE Draft P802.1w/D2, Supplement to ISO/IEC 15802-3 (IEEE Std 802.1D) Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Common specifications – Part 3: Media Access Control (MAC) Bridges: Rapid Reconfiguration.

[FDDI] ISO-9314-1, Information Processing Systems – Fibre Distributed Data Interface (FDDI) – Part 1: Token Ring Physical Layer Protocol (PHY).

[SMT] *ISO-9314-6, Information Processing Systems – Fibre Distributed Data Interface (FDDI) – Part 6: Station Management*.

[HILES95] Hiles, William S., Marlow, David T., *Approximation of FDDI Minimum Reconfiguration Time*, Proceedings of the IEEE Computer Society 20th Conference on Local Computer Networks, September 1995†.

[HUANG] Huang, J., Song, S., Li, L., Kappler, P., Freimark, R., Gustin, J., Kozlik, T., *An Open Solution to Fault-Tolerant Ethernet: Design, Prototyping, and Evaluation*, 18th IEEE International Performance, Computing, and Communications Conference, February 1999.

[IREY97] Irey, Philip M., Marlow, David T., Harrison, Robert D., *Distributing Time Sensitive Data in a COTS Shared Media Envoronment*, 5th International Workshop on Parallel and Distrbuted Real-Time Systems, pp. 53-62, April 1997†

[IREY98] 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.†

[KESSLER] Kessler, G., Shepard, S., *RFC 1739 - A Primer On Internet and TCP/IP Tools*, December 1994.

[MILLS] Mills, David L., *RFC 1305 - Network Time Protocol (Version 3) Specification, Implementation and Analysis*, March 1992.

[RALPH] Ralph, Stanley F., Ukrainsky, Orest J., Schellak, Robert H., Weinberg, Leonard, *Alternate Path FDDI Topology*, Proceedings of the IEEE Computer Society 17th Conference on Local Computer Networks, September 1992.

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