Failure Modes and Failure Characteristics of a TDMA driven Ethernet

J. Reisinger
Institut für Technische Informatik
Technische Universität Wien
reising@vmars.at

Version 1.1; April 16, 1991

Abstract

When developing protocols for a specific communication medium, it is very useful to have a detailled fault hypothesis available. In this paper the failure behaviour of the TDMA driven Ethernet in the MARS system will be described.

Contents

1 Introduction **3**

2 Testing Environment **3**

3 Classification of Failure Modes **4**

4 Experimental Results **4**

4.1 Message Mutilation 5

4.1.1 Test Program 5

4.1.2 Test Results 5

4.2 Message Loss 5

4.2.1 Test Program 5

4.2.2 Test Results 6

5 Conclusion **9**

1 Introduction

When developing protocols for LAN's[1], it is very useful to have a detailed fault-hypothesis available. In particular, the failure modes and failure rates of the underlying medium have to be known to optimize a protocol in order to achieve high reliability and speed.

Although it is not absolutely necessary to know the failure-behaviour of the communication-medium to develop reliable protocols (e.g. a PAR[2]-protocol with infinite stop time is able to handle each message loss or duplication [Sun75]) in conventional computer systems, protocols which are used in real-time systems have to achieve maximum reliability within predefined transmission time limits. If there does not exist any fault-hypothesis, neither the probability for a protocol fault can be computed nor the protocol can be optimized to minimize this probability.

The subject of this work is the failure behaviour of a TDMA[3] driven, therefore collision-free, Ethernet. The standard Ethernet protocol, CSMA/CD[4] is not well suited for failure mode analysis, since it does not work collision-free. Failures, which result from the medium will be mixed with failures caused by the protocol in this case (caused by collisions or be exceeding the retransmission limit).

The immediate reason for testing the Ethernet was the development of a membership protocol [Kop89b] for MARS[5] [Kop89a]. To get realistic results, and because in MARS the TDMA protocol is already implemented, the testing was performed within the MARS system.

2 Testing Environment

A MARS Cluster consists of some MARS components (processing nodes) which are interconnected by the MARS bus.

The MARS Bus is a Cheapernet (similar to an Ethernet) cable, which allows the communication between MARS components by the MARS bus protocol.

The MARS Bus Protocol is a slightly modified version of a TDMA protocol. To each component there is assigned a time-slot, during which the component may send a broadcast message. To increase fault tolerance, each

[1] Local Area Networks
[2] Positive Acknowledgement or Retransmission
[3] Time Division Multiple Access
[4] Carrier Sense Multiple Access with Collision Detection
[5] MAintainable Real-time System

message is sent twice on the MARS bus (however, the MARS operating system [Dam89],[Rei89] allows to turn off this feature).

MARS Messages are provided with names. To each TDMA slot there is assigned the name of a message to be sent in it (this is done for a period of time, then the bus-schedule will be repeated cyclically). Therefore, the point in time when a specific message will be sent is known in advance.

3 Classification of Failure Modes

Before starting to develop programs to test the Ethernet, the possible failure modes must be determined:

- Loss of a message at the sender: None of the possible receivers of the message receives the message. Because medium failures cannot be distinguished from sender failures, in the following the term 'sender failure' stands for both.
- Loss of a message at the receiver: Exactly one component does not receive the message, all other do receive it.
- Loss of a message at some receivers: A message is lost at some, but not at all receivers. It has to be determined, if this is an autonomous failure mode or if its only cause is a statistically independent combination of receiver failures.
- Duplication of a message: This failure mode will not be handled here, since the message handling system of MARS automatically handles this mode (in MARS there are only state messages; a newer message will overwrite an older one with the same name). In addition, we see no way to duplicate a message at a broadcast medium caused by the medium itself.
- Mutilation of a message. A message at the receiver differs from the corresponding message at the sender. If this failure occurrs at some receivers, and some receive the correct message, a byzantine failure has happened.

4 Experimental Results

All experiments were made in an experimental cluster consisting of six nodes. Unless explicitly mentioned, the double sending feature was turned off.

4.1 Message Mutilation

The first to test was if there is a measurable probability for message mutilation. The Ethernet controller provides each message with a CRC[6], which drastically reduces this probability. If a CRC error is detected, the message is assumed to be faulty and therefore discarded (this results in a message loss).

4.1.1 Test Program

Each test-cycle consists of six TDMA-cycles (in one TDMA-cycle each node may send once). Before the first cycle, each component generates a random message. This message is stored in the local memory as well as transmitted to the other five components. During the next five TDMA-cycles, each node sends back each received message. After having received its own message five times, they are compared with the original. Any detected difference is reported. Any mutilation during the transmission of the 36 messages (each of the six components has transmitted six messages) will be detected.

4.1.2 Test Results

Since the length of a TDMA slot is 4 ms, 250 messages were transmitted in one second, $14 * 24 * 60 * 60 * 250 = 302400000$ during the whole experiment, which lasted about two weeks. During the transmission of the 300 million messages no mutilation error was detected. In all further experiments this failure mode was not taken into account.

4.2 Message Loss

During the first experiments it was detected that the probabilities to loose a message are not equal for the different components. As a consequence, not only the overall message loss probabilities were measured, but also the individual probabilities for message losses of the components.

4.2.1 Test Program

In a configuration of six nodes each component sends one message to all others in each TDMA cycle (24 ms). This message contains information about the messages lost in the last round in the component. After a component has received this vector from every other component, it can build a matrix containing all message losses in

[6]Cyclic Redundancy Check

all components (this matrix is the same in each component). Using this matrix, it can be detected whether a message got lost at exactly one component (receiver failure), at all components (sender failure) or at some components (partial failure).

In each node there are some matrices, which contain the sums of all different failures occurred in the system since system startup (for single, partial, and sender failures). This matrices (which are equal in each component) are sent to the host at the end of each experiment.

4.2.2 Test Results

After performing the tests a few times, we were able to assign a characteristical failure probability to each sender – receiver pair. Using these results, a configuration was chosen, which contains components with very different failure probabilities. The ability to test nodes with a high failure probability facilitates the task to determine whether failures are statistically independent (in a configuration where no double failures occur, statistical dependency cannot be determined).

		Receiver						
	→	0	1	2	3	4	5	Σ
S	0	-	521	0	56074	85	3	56683
e	1	3367	-	0	93041	2355	2	98765
n	2	1254276	6	-	13974	299	4	1268559
d	3	84	3	0	-	24	0	111
e	4	280	25	0	24593	-	1	24899
r	5	1891528	15	0	36298	403	-	1928244
	Σ	3149535	570	0	223980	3166	10	3377261

Table 1: Incoming link failures (2 266 587 742 sent messages)

Sending component	0	1	2	3	4	5	Σ
Number of failures	10	1	5	1	1	6	24

Table 2: Outgoing link failures (2 266 587 742 sent messages)

In tables 1 and 2 the results of the first experiment can be seen. This experiment lasts over three months; each four milliseconds a message was sent at the MARS-bus. This results in exactly 2 266 587 742 sent messages. Each element of the matrix specifies the number of lost messages for a sender – receiver pair (the lines of the matrix represent the senders, the columns the receivers). The

maximum number of messages which each node may receive from a specific sender is one sixth of the total number of sent messages (= 377 764 624 messages).

You see that there are orders of magnitude between the failure rates of the components. E.g., component zero lost three million out of the two billion sent messages (0.14%), whereas component two did not loose any (!) of the two billion sent messages.

Partial Failures

Besides outgoing link failures and single incoming link failures, we also measured partial incoming link failures (messages, which get lost at two or more components, but not at all components). Using this information it can be determined if incoming link failures at different components are statistically independent:

1. The message loss probability for each sender – receiver pair is calculated out of the values contained in table 1 (because of the size of the spot-check the calculated probabilities will be very close to the real ones).

2. After this, the probability for a double failure can be calculated for each sender – receiver – receiver triplet (assuming statistical independence of double failures). The result is a matrix whose lines represent the senders, and whose columns represent all possible pairs of receivers.

3. By multiplying each value in this matrix with the total number of sent messages (of each component), a matrix containing the expectation for each kind of double failure can be created. You can see this matrix in table 3. There are two lines provided for each sender; the first line contains the computed expectations, the second line the measured message losses.

→	0&1	0&2	0&3	0&4	0&5	1&2	1&3	1&4	1&5	2&3	2&4	2&5	3&4	3&5	4&5
0	-	-	-	-	-	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
	-	-	-	-	-	0	0	0	0	0	0	0	1	0	0
1	-	0.00	0.83	0.02	0.00	-	-	-	-	0.00	0.00	0.00	0.58	0.00	0.00
	-	0	1	0	0	-	-	-	-	0	0	0	0	0	0
2	0.02	-	46.40	0.99	0.01	-	0.00	0.00	0.00	-	-	-	0.01	0.00	0.00
	0	-	41	2	0	-	0	0	0	-	-	-	0	0	0
3	0.00	0.00	-	0.00	0.00	0.00	-	0.00	0.00	-	0.00	0.00	-	-	0.00
	0	0	-	0	0	0	-	0	0	-	0	0	-	-	0
4	0.00	0.00	0.02	-	0.00	0.00	0.00	0.00	0.00	0.00	-	-	-	0.00	-
	0	0	0	-	0	0	0	0	0	0	-	-	-	0	-
5	0.08	0.00	181.75	2.02	-	0.00	0.00	0.00	-	0.00	0.00	-	0.04	-	-
	1	0	212	2	-	0	0	0	-	0	0	-	0	-	-

Table 3: Computed and real partial failure number

4. Using this matrix, it can be tested if the measured values are within a 99% confidence interval. The distribution function for the number of message losses is $B_{n,p}$ (binomial distribution function). Since the probability p is very small, $B_{n,p}$ can be approximated with P_{np} (Poisson distribution). In table 4 the confidence intervals can be seen. Each value is within its confidence interval, therefore our assumption of independent failures holds.

→	0&1	0&2	0&3	0&4	0&5	1&2	1&3	1&4	1&5	2&3	2&4	2&5	3&4	3&5	4&5
0	-	-	-	-	-	0.00	0.08	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
	-	-	-	-	-	0	0	0	0	0	0	0	1	0	0
	-	-	-	-	-	[0,0]	[0,1]	[0,0]	[0,0]	[0,0]	[0,0]	[0,0]	[0,1]	[0,0]	[0,0]
1	-	0.00	0.83	0.02	0.00	-	-	-	-	0.00	0.00	0.00	0.58	0.00	0.00
	-	0	1	0	0	-	-	-	-	0	0	0	0	0	0
	-	[0,0]	[0,3]	[0,1]	[0,0]	-	-	-	-	[0,0]	[0,0]	[0,0]	[0,3]	[0,0]	[0,0]
2	0.02	-	46.40	0.99	0.01	-	0.00	0.00	0.00	-	-	-	0.01	0.00	0.00
	0	-	41	2	0	-	0	0	0	-	-	-	0	0	0
	[0,1]	-	[30,65]	[0,4]	[0,1]	-	[0,0]	[0,0]	[0,0]	-	-	-	[0,1]	[0,0]	[0,0]
3	0.00	0.00	-	0.00	0.00	0.00	-	0.00	0.00	-	0.00	0.00	-	-	0.00
	0	0	-	0	0	0	-	0	0	-	0	0	-	-	0
	[0,0]	[0,0]	-	[0,0]	[0,0]	[0,0]	-	[0,0]	[0,0]	-	[0,0]	[0,0]	-	-	[0,0]
4	0.00	0.00	0.02	-	0.00	0.00	0.00	0.00	0.00	0.00	-	-	-	0.00	-
	0	0	0	-	0	0	0	0	0	0	-	-	-	0	-
	[0,0]	[0,0]	[0,1]	-	[0,0]	[0,0]	[0,0]	[0,0]	[0,0]	[0,0]	-	-	-	[0,0]	-
5	0.08	0.00	181.75	2.02	-	0.00	0.00	0.00	-	0.00	0.00	-	0.04	-	-
	1	0	212	2	-	0	0	0	-	0	0	-	0	-	-
	[0,1]	[0,0]	[148,217]	[0,6]	-	[0,0]	[0,0]	[0,0]	-	[0,0]	[0,0]	-	[0,1]	-	-

Table 4: Confidence intervals for the number of partial failures

Double Failures

Since each message is sent twice in a TDMA slot in MARS, it has to be tested whether failures in these two messages are statistically independent or not. To do so, the previous experiment was repeated with the double sending feature of MARS turned on. This experiment lasts about one month; 663 484 194 messages were sent during this experiment. Table 5 shows the results of this experiment (as expected, there were no outgoing link failures).

Like in the previous chapter, the expectations for double failures and the confidence intervals were calculated out of table 1 (assuming statistical independence of double failures). The results are shown in table 6. You see, that our hypothesis of statistical independence holds, because each measured value lies within its confidence interval.

→	0	1	2	3	4	5	Σ
0	-	0	0	6	0	0	6
1	0	-	0	5	0	0	5
2	1177	0	-	2	0	0	1179
3	0	0	0	-	0	0	0
4	0	0	0	0	-	0	0
5	2803	0	0	1	0	-	2804
Σ	3980	0	0	14	0	0	3994

Table 5: Incoming link failures (663 484 194 sent messages) with double sending feature turned on

5 Conclusion

Based on our experiments, the failure modes of an Ethernet can be characterized as follows:

- Most of the failures are incoming link failures: One component does not receive the message, all other do receive it. The rate of outgoing link failures is very small compared to it. During our experiments there was no occurrence of a message mutilation.
- There is a characteristic failure rate for each sender – receiver pair. There are orders of magnitude between some of these rates.
- Concurrent receiver failures at different nodes are statistically independent.
- Losses of two succeeding messages at one component are statistically independent.

Based on our experiments, we found a new way to decrease the Ethernet failure rate significantly: Selecting 'good' components (components with low failure rates) out of a pool of components and building a cluster with low failure rate (probably the failure rate depends on one or two chips, therefore it will be enough to select the chips and not the components).

To verify this hypothesis, we tested a cluster of selected components (the results can be seen in table 7). Out of this experiment, a MTTF[7] of about one year has been computed for double failures (= loss of a single message when using the double sending feature of the operating system).

[7] Mean Time To Failure

→	0	1	2	3	4	5
0	-	0.00	0.00	2.44	0.00	0.00
	-	0	0	6	0	0
	-	[0,0]	[0,0]	[0,7]	[0,0]	[0,0]
1	0.01	-	0.00	6.71	0.00	0.00
	0	-	0	5	0	0
	[0,1]	-	[0,0]	[1,14]	[0,0]	[0,0]
2	1219.05	0.00	-	0.15	0.00	0.00
	1177	0	-	2	0	0
	[1130,1310]	[0,0]	-	[0,2]	[0,0]	[0,0]
3	0.00	0.00	0.00	-	0.00	0.00
	0	0	0	-	0	0
	[0,0]	[0,0]	[0,0]	-	[0,0]	[0,0]
4	0.00	0.00	0.00	0.47	-	0.00
	0	0	0	0	-	0
	[0,0]	[0,0]	[0,0]	[0,3]	-	[0,0]
5	2772.44	0.00	0.00	1.02	0.00	-
	2803	0	0	1	0	-
	[2638,2909]	[0,0]	[0,0]	[0,4]	[0,0]	-

Table 6: Confidence intervals for the number of double failures

→	0	1	2	3	4	5	Σ
0	-	4	0	139	0	1	144
1	38	-	1	188	0	7	234
2	3	414	-	470	0	2	889
3	583	0	0	-	0	0	583
4	198	5	83	13	-	1	300
5	42	9	3	261	0	-	315
Σ	864	432	87	1071	0	11	2465

Table 7: Incoming link failures (579 524 472 messages, selected components)

References

[Dam89] A. Damm, J. Reisinger, W. Schwabl, and H. Kopetz. The Operating System of MARS. *ACM Operating System Review*, 23(3):141–157, July 1989.

[Kop89a] H. Kopetz, A. Damm, Ch. Koza, M. Mulazzani, W. Schwabl, Ch. Senft, and R. Zainlinger. Distributed Fault-Tolerant Real-Time Systems: The MARS Approach. *IEEE Micro*, 9(1):25–40, Feb. 1989.

[Kop89b] H. Kopetz, G. Grünsteidl, and J. Reisinger. Fault-Tolerant Membership Service in a Synchronous Distributed Real-Time System. In *Int. Working Conference on Dependable Computing for Critical Applications*, Santa Barbara, CA, USA, Aug. 1989.

[Rei89] J. Reisinger. *Inside the MARS Operating System.* Research Report 7/89, Institut für Technische Informatik, Technische Universität Wien, Vienna, Austria, July 1989.

[Sun75] C. Sunshine. Interprocess Communication Protocols for Computer Networks. *Stanford Electronic Lab, TR-105*, 258 pp., Dec 1975.