P-25

NASA Technical Memorandum 86660

Verification and Performance Tests of HYCAR Program

Veena Bhatia

IN-61
DATE OVERRIDE
97829

(NASA-TM-86660) VERIFICATION AND PERFORMANCE TESTS OF HYCAR PROGRAM (NASA) 25 p Avail: NTIS HC A02/MF A01 CSCL 09B

N87-29185

Unclas
G3/61 0097829

NASA
National Aeronautics and
Space Administration

Date for general release July 1987

NASA Technical Memorandum 86660

Verification and Performance Tests of HYCAR Program

Veena Bhatia, Ames Research Center, Moffett Field, California

June 1985

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

SUMMARY

HYCAR is a program that simulates the network protocols of HYPERchannel and FODS (Fiber Optic Demonstration System). The FODS protocol was developed by Sperry Flight Systems, Phoenix, Arizona. The user may also simulate other related protocols by modifying the input data file for the program. This report details the verification and performance tests conducted on the program using the FODS protocol and topology.

The report contains the following two sections:

(1) Section 1 provides a description of the verification tests conducted, including the objective, approach, results, and conclusions of each test; and,

(2) Section 2 provides a summary of the FODS protocol performance characteristics under varied loading conditions.

Section 1 presents information that verifies the results of the simulation, and checks the consistency of the results by comparing them with the expected results which were obtained through deterministic and analytical means. Section 2 documents the performance of the protocol under different loading conditions. The performance summary also validates the model of FODS that was used in HYCAR. These results were obtained through extensive experimentation with the simulator.

The five tests verified the following parameters:

(1) Throughput and channel efficiency;

(2) Number of collisions;

(3) Average time to resolve collisions;

(4) Average dead time on the channel;

(5) Independence of the message length.

INTRODUCTION

HYCAR is a program that simulates the network protocols of HYPERchannel and FODS (Fiber Optic Demonstration System). The FODS protocol was developed by Sperry Flight Systems, Phoenix, Arizona. The user may also simulate other related protocols by modifying the input data file for the program. This report details the verification and performance tests conducted on the program using the FODS protocol and topology.

The FODS network uses a star topology with a channel bit rate of 100 Mbps. The protocol is similar to that of HYPERchannel. Both protocols use the same access scheme: Carrier Sense Multiple Access (CSMA), and collision resolution through timeslots. The FODS protocol, however, only cycles once through the timeslots, while the HYPERchannel cycles until all pending messages have been sent.

The two sections of this report provide descriptions of the verification tests that were conducted and a summary of the FODS protocol performance characteristics. The five tests used analytical methods to verify the following parameters:

(1) Throughput and channel efficiency;

(2) Number of collisions;

(3) Average time to resolve collisions;

(4) Average dead time on the channel;

(5) Independence of the message length.

These tests and their results are detailed in Section 1. Extensive experimentation with the simulator produced a set of performance characteristics for the FODS protocol under varied loading conditions. These characteristics are summarized in Section 2.

Revisions

The verification tests of HYCAR revealed two bugs in the program code. When these bugs were resolved, the following three versions of the program resulted.

Version 1 (May 1984) had an intermittent bug; sometimes the program would not count through the time slots correctly in the controlled access mode. As a result, stations would collide in that mode, an occurrence which is not permitted by the protocol. Because the bug appeared intermittently, however, it was possible to obtain results from runs in which the bug did not appear.

Version 2 (July 1984) resulted when this intermittent bug was resolved. This program, however, contained a second bug. Calculations of the throughput and efficiency were not done correctly; the problem involved substitution of data generated for data sent. Throughput should be calculated as

$$\frac{\text{Kbytes sent successfully}}{\text{run-time } (\mu\text{sec})}$$

Instead, throughput was being calculated as

$$\frac{\text{Kbytes generated}}{\text{run-time } (\mu\text{sec})}$$

This error resulted from the changes that were made between versions 1 and 2. Version 1 of HYCAR ran until all of the buffers were emptied; therefore, the data generated equalled the data sent. Version 2, however, ran until the end of the run-time, not until the buffers were emptied. Therefore, the two parameters did not have the same value.

Version 2a (August 1984) resulted when this error was corrected. No bugs were found in this version.

Parameters

Several parameters retained the same values throughout the tests. These parameters and their values are listed in tables 1 and 2.

TABLE 1.- TEST PARAMETERS WITH CONSTANT VALUES

Topology	Star
Arm length	50 m each
Number of Bus Interface Units (BIUs)	4
Channel bit rate	100 Mbps
Versions 1, 2 message length	16496 bits[a]
Version 2a message length	16384 bits[a]
Run-time	100000 μsec

TABLE 2.- CROSS REFERENCE OF UTILIZATION AND INTERARRIVAL TIME

Expected channel utilization[a]	Message interarrival time[b]
90%	700 μsec
45%	1500 μsec
20%	3000 μsec

[a]Except in test #5.
[b]Mean of a Poisson process; given on a per BIU basis.

SECTION 1: DESCRIPTION OF TESTS

Test #1: Throughput and Channel Efficiency

Objective

The objective of this test was to verify the following:

(1) That the throughput and channel efficiency is being evaluated correctly; and,

(2) That the results are comparable to those obtained from HYCAR version 1 runs and to the expected results.

Approach

Run three simulations, varying only the message interarrival times. Select the three interarrival times so that the expected channel utilizations are corresponding percentages of the maximum possible (see table 2).

Select the number of nodes to be four, and a run-time of 100000 μsec. This run-time allows between 30 and 140 messages/node/run depending on the interarrival time.

Check that throughput is being calculated as

$$\frac{\text{Kbytes sent successfully}}{\text{run-time } (\mu\text{sec})}$$

and that channel utilization is being calculated as

$$\frac{\text{Throughput (Mbps)}}{\text{100 Mbps}} \times 100\%$$

Compare these data with data taken from runs made with version 1 of HYCAR. Also compare the actual achieved throughput with the attempted throughput to make sure it is in the same range for a non-overloaded channel.

Results

Version 2

VERSION 2 RESULTS FOR TEST #1

	Interarrival time, μsec		
	700	1500	3000
Mean offered load (Mbps)[a] (Versions 1 and 2)	94.3	44.0	22.0
Throughput (Mbps) (Version 1)	88.0	41.8	19.0
Throughput (Mbps) (Version 2)	89.6	46.2	17.8

[a]Mean offered load (bits/sec) =

$$\frac{\text{message size (bits)}}{\text{interarrival time (sec)}} \times \text{number of BIUs}$$

(Load offered to the system; does not include retries of messages involved in collisions; mean of a Poisson process.)

Also see figure 1 for a plot of throughput versus the message arrival rate.

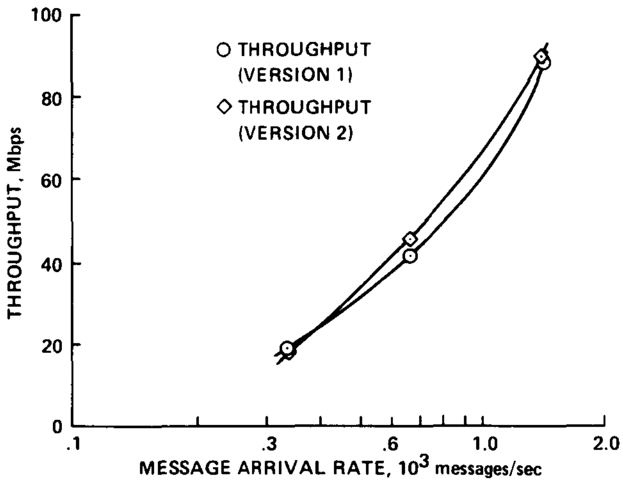

Figure 1.- Throughput vs. message arrival rate, versions 1 and 2.

Version 2a

VERSION 2a RESULTS FOR TEST #1

	Interarrival time, μsec		
	700	1500	3000
Mean offered load (Mbps)[a] (Version 1)	94.3	44.0	22.0
Mean offered load (Mbps)[a] (Version 2a)	93.6	43.7	21.8
Throughput (Mbps) (Version 1)	88.0	41.8	19.0
Throughput (Mbps) (Version 2a)	92.6	45.2	19.7

[a]Mean offered load (bits/sec) =

$$\frac{\text{message size (bits)}}{\text{interarrival time (sec)}} \times \text{number of BIUs}$$

(Load offered to the system; does not include retries of messages involved in collisions; mean of a Poisson process.)

Also see figure 2 for a plot of throughput versus the message arrival rate.

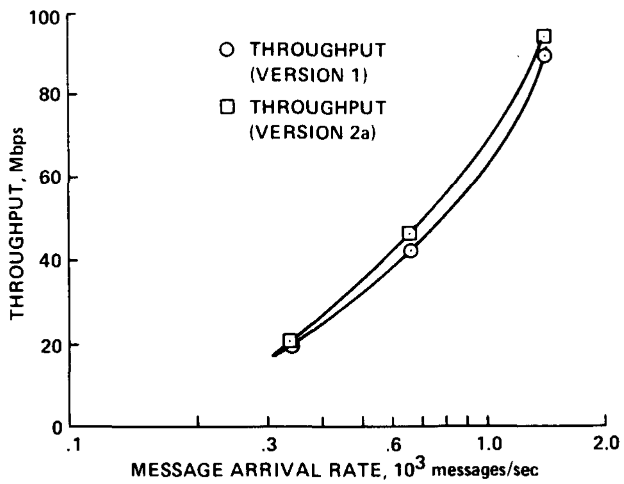

Figure 2.- Throughput vs. message arrival rate, versions 1 and 2a.

Conclusions

Version 2

A. Evaluation of throughput and channel efficiency

Calculations of the throughput and efficiency were not done correctly, as was explained in the introduction.

B. Comparisons with version 1 results

No comparisons can be made as yet because of the error noted in A above.

Version 2a

A. Evaluation of throughput and channel utilization

The throughput and channel efficiency were calculated correctly using the amount of data sent rather than the amount generated.

The mean offered load for version 2a is slightly different from that of version 1. The difference is caused by a change in the message sizes used in the two versions. Version 1 used a message size of 16496 bits, while version 2a used 16384 bits.

The change was made to make the results more accurate: 16496 bits represents the size of the message including both data and header. The size of the data alone is, however, 16384 bits. To calculate the throughput, only the amount of data sent, not header information, is necessary. Therefore, the message size was changed to 16384 bits.

B. Comparison with version 1 results

The throughputs actually achieved in versions 1 and 2 are comparable. They are not exactly the same because of the randomness of the program. Since the program gives different results each time it is run, the variations in the results are as expected.

Comparisons between the mean offered load and the achieved throughput reveal the same situation. Variations exist and are statistically reasonable. The achieved throughput is sometimes greater than the mean offered load because the load is calculated using the message interarrival time, which is a mean of a Poisson process, not a constant value.

Test #2: Number of Collisions

Objective

The objective of this test was to verify the following:

(1) That the count of collision events and number of colliding messages given at the end of the run is correct; and

(2) That the results are comparable to those obtained from HYCAR version 1 runs.

Approach

Run three simulations, varying only the message interarrival times. Select the three interarrival times so that the expected channel utilizations are corresponding percentages of the maximum possible (see table 2).

Select the number of nodes to be four, and a run-time of 100000 μsec. Count, on the event trace, the number of collisions and the number of colliding messages. Verify the end-of-run statistics.

Graph the totals versus the message arrival rate (1/interarrival time) to compare them with results obtained from version 1 runs.

Results

Version 2

VERSION 2 RESULTS FOR TEST #2

	Interarrival time, μsec		
	700	1500	3000
Number of colliding messages (Version 1)	118[a]	30	6
Number of colliding messages (Version 2)	342	34	2
Number of collisions (Version 1)	32[a]	14	3
Number of collisions (Version 2)	138	17	1
Number of messages/collision (Version 1)	5	18	38
Number of messages/collision (Version 2)	4	16	111

[a]Run-time = 30000 μsec.

Also see figure 3 for a plot of the number of collisions versus the message arrival rate.

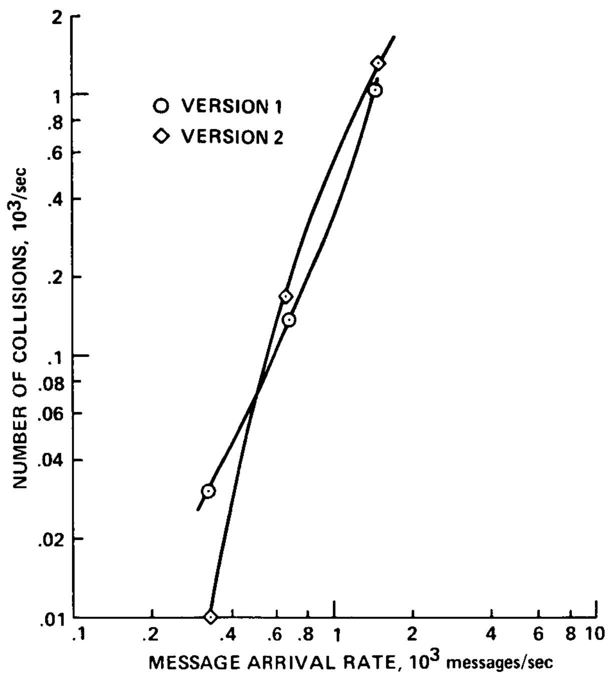

Figure 3.- Number of collisions vs. message arrival rate, versions 1 and 2.

Version 2a

VERSION 2a RESULTS FOR TEST #2

	Interarrival time, μsec		
	700	1500	3000
Number of colliding messages (Version 1)	118[a]	30	6
Number of colliding messages (Version 2a)	350	32	8
Number of collisions (Version 1)	32[a]	14	3
Number of collisions (Version 2a)	147	15	4
Number of messages/collision (Version 1)	5	18	38
Number of messages/collision (Version 2a)	4	18	30

[a]Run-time = 30000 μsec.

Also see figure 4 for a plot of the number of collisions versus the message arrival rate.

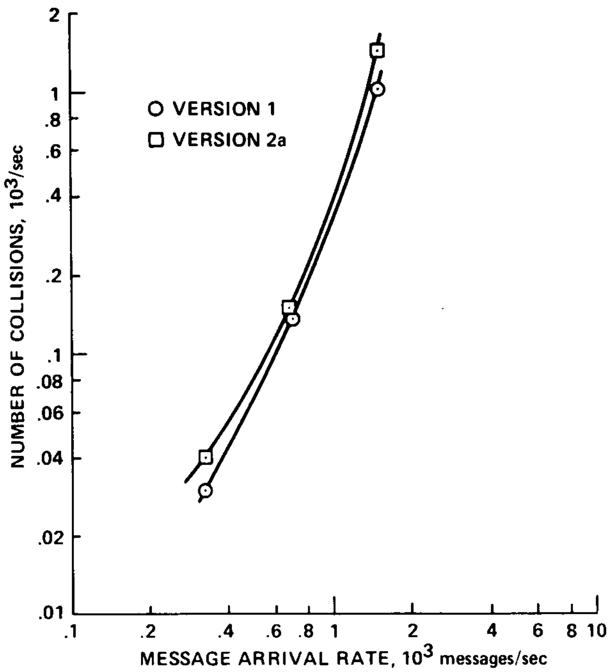

Figure 4.- Number of collisions vs. message arrival rate, versions 1 and 2a.

<u>Conclusions</u>

Version 2

A. Verification of end-of-run statistics

This test was completed by counting the number of collisions and number of colliding messages on an event trace. These counts were then compared to the statistics given at the end of the run. The two counts were the same.

B. Comparison with version 1 results

No comparisons can be made with version 1 because of the error that was detected.

Version 2a

A. Verification of end-of-run statistics

As with version 2, the end-of-run statistics in version 2a are equal to the number of collisions and colliding messages as counted on an event trace.

B. Comparison with version 1 results

The results of versions 1 and 2a are graphed in figure 4. As expected, the number of collisions increased as the message interarrival time decreased.

The maximum limit to the number of collisions that can occur is one collision every four messages. This would occur if four messages are sent in the controlled access mode and a collision occurs each time the random access mode is entered. Studying the simulation results reveals this to be the case under heavy loading conditions.

As the loading is decreased, the number of collisions decreases drastically. Again, comparison with simulation results shows this to be true. In version 2a, with a message interarrival time of 700 μsec, the number of collisions/message was 1/4. This value decreased to 1/30 for a message interarrival time of 3000 μsec. Similar results were obtained from runs of version 1.

Test #3: Resolution Time of Collisions

Objective

The objective of this test was to measure the average amount of time taken to resolve collisions under different loading conditions. (This is the time between the detection of a collision and the beginning of time slot #1 in the controlled access mode.)

Approach

Run three simulations, varying only the message interarrival times. Select the three interarrival times so that the expected channel utilizations are corresponding percentages of the maximum possible (see table 2).

Select the number of nodes to be four, and a run-time of 100000 μsec. For the runs with 45% and 20% utilizations, few collisions may occur; it may be necessary to do additional runs to find an average.

Compare the measured times with calculated limits and average. The calculated limits are based on the number of BIUs in the system and the length of the nonvalid Manchester (NVM) signal.

Results

Version 2

VERSION 2 RESULTS FOR TEST #3

Resolution time (μsec) collision number	Interarrival time, μsec		
	700	1500	3000
1	4.4	4.0	4.0
2	4.3	4.0	4.0
3	4.2	4.0	4.0
4	4.2	4.0	4.0
5	4.0	4.0	4.0
6	4.1	4.0	4.0
7	4.4	4.0	4.0
8	4.5	4.0	4.2
9	4.3	4.0	4.0
10	4.2	4.0	4.2
Average[a]	4.26	4.0	4.02

[a]Average taken over 10 collisions.

Also see figure 5 for a plot of the average time to resolve collisions versus the message arrival rate.

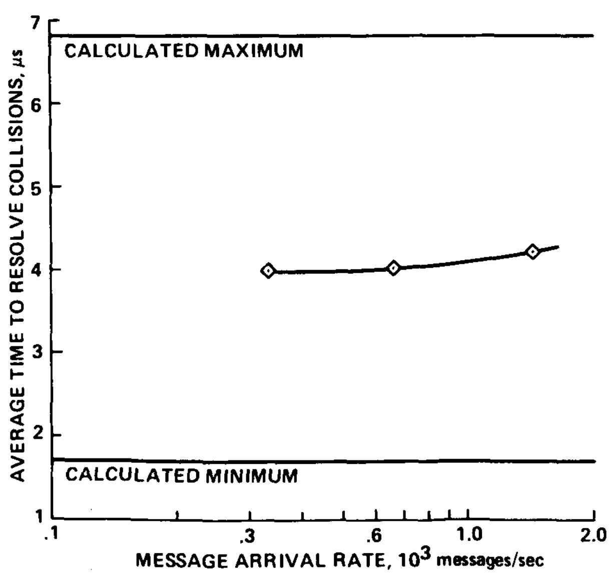

Figure 5.- Average time to resolve collisions vs. message arrival rate, version 2.

Version 2a

VERSION 2a RESULTS FOR TEST #3

Resolution time (μsec) collision number	Interarrival time, μsec		
	700	1500	3000
1	3.73	3.73	3.73
2	3.73	3.73	3.73
3	3.93	3.73	3.73
4	6.61	3.73	3.73
5	3.73	3.73	3.73
6	4.03	3.73	3.73
7	3.73	3.73	3.73
8	3.73	3.73	3.73
9	3.93	3.73	3.73
10	3.73	3.73	3.73
11	4.26	3.73	3.73
12	3.83	3.73	3.73
13	3.73	3.73	3.73
14	3.73	4.03	3.73
15	4.13	3.73	3.73
Average[a]	4.04	3.75	3.73

Calculated maximum = 6.8 μsec[b]
Calculated minimum = 1.7 μsec[c]
Calculated average = 4.25 μsec[d]

[a]Average taken over 15 collisions.

[b]Calculated maximum =
(number of BIUs) × (T_{gap})
+ (number of BIUs) × (length of NVM signal)
= (4 × 1.6 μsec) + (4 × 0.1 μsec)

[c]Calculated minimum =
T_{gap} + length of NVM signal
= 1.6 μsec + 0.1 μsec

[d]Calculated average = $\frac{\text{calculated maximum + calculated minimum}}{2}$

Also see figure 6 for a plot of the average time to resolve collisions versus the message arrival rate.

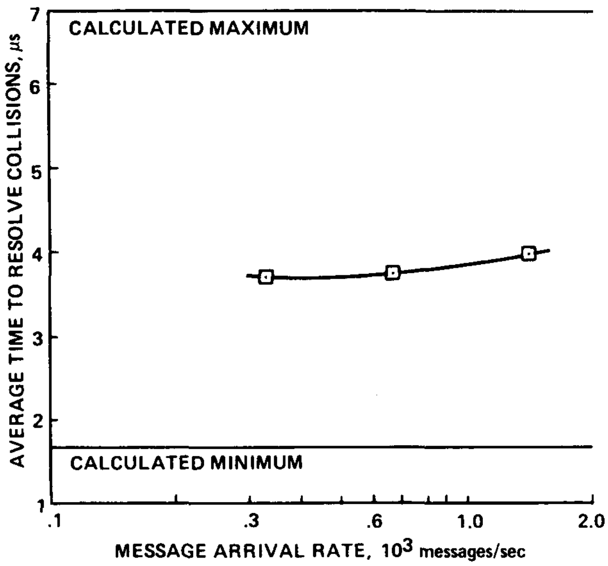

Figure 6.- Average time to resolve collisions vs. message arrival rate, version 2a.

Conclusions

Version 2

No conclusions can be made as yet. This test will be redone when the error in the program is corrected.

Version 2a

Comparison of the measured average time to resolve a collision with the calculated time shows that they are very close to each other. The calculated average was made on the assumption that collisions involving any number of nodes occur at the same frequency. However, collisions between two nodes actually occur much more often than those involving three or four nodes. Therefore, the measured average time to resolve collisions will be lower than that calculated.

Under heavy loading, more messages may collide in any one collision than under lighter loading. Detailed analysis of the event trace shows this to be true. The average time to resolve collisions is, therefore, higher under heavy loading than under lower loading. All three averages are well within the expected range; all individual values are also within the limits.

Test #4: Dead Time on the Channel

Objective

The objectives of this test were:

(1) To measure the average amount of time between transmissions (dead time) in the random access mode, under different loading conditions; and,

(2) To correlate the amount of dead time with the channel utilization.

Approach

Run three simulations, varying only the message interarrival times. Select the three interarrival times so that the expected channel utilizations are corresponding percentages of the maximum possible (see table 2).

Select the number of nodes to be four, and a run-time of 100000 μsec. The measurements will be taken near the middle of the run so that start-up transients are not included; only steady-state data will be taken.

The average dead time should increase significantly as the load is decreased and should account, in part, for the low channel utilizations. Compare the percentage of dead time to the channel utilization for correlation.

Results

Version 2

VERSION 2 RESULTS FOR TEST #4

	Interarrival time, μsec		
	700[a]	1500[a]	3000[b]
Average dead time (μsec)[c]	16.58	197.36	663.94
Percent dead time[d]	9.3	55.4	75.7
Percent channel utilization	89.6	46.2	18.6

[a]Average taken over 50 messages.

[b]Average taken over 35 messages.

[c]Average dead time, μsec =

$$\frac{\sum_{1}^{N} \text{Time between transmissions in random access mode}}{N}, \text{ where } N = 35 \text{ or } N = 50,$$

as noted above in a and b.

[d]Percent dead time =

$$\frac{\text{Average dead time} \times \text{number of messages sent}}{\text{run-time}} \times 100\%$$

See also figure 7 for a plot of the average dead time versus the message arrival rate.

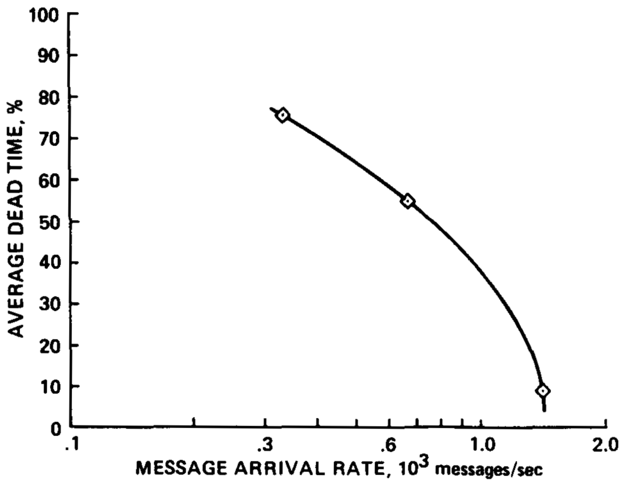

Figure 7.- Average percent dead time vs. message arrival rate, version 2.

Version 2a

VERSION 2a RESULTS FOR TEST #4

	Interarrival time, μsec		
	700[a]	1500[a]	3000[b]
Average dead time (μsec)[c]	8.4	188.1	617.0
Percent dead time[d]	4.7	51.9	74.0
Percent channel utilization	92.6	45.2	19.7

[a]Average taken over 50 messages.

[b]Average taken over 35 messages.

[c]Average dead time, μsec =

$$\frac{\sum_{1}^{N} \text{Time between transmissions in random access mode}}{N}$$, where N = 35 or N = 50,

as noted above in a and b.

[d]Percent dead time =

$$\frac{\text{Average dead time} \times \text{number of messages sent}}{\text{run-time}} \times 100\%$$

Also see figure 8 for a plot of the average dead time versus the message arrival rate.

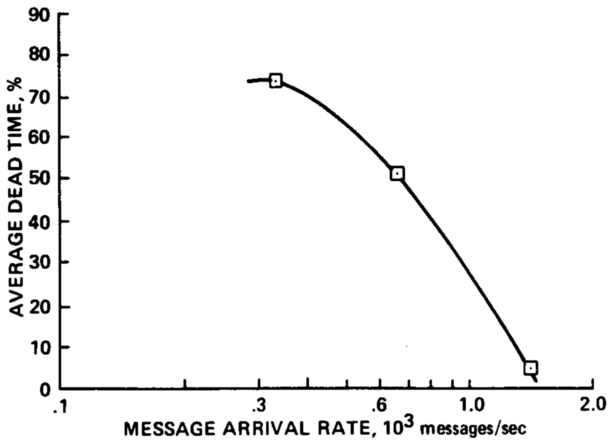

Figure 8.- Average percent dead time vs. message arrival rate, version 2a.

Conclusions

Version 2

The average dead time on the channel was measured by calculating the amounts of time between messages in the random access mode and then taking their average. Correlating the amount of dead time and the channel utilization is not possible because of the error in calculating the utilization.

Version 2a

The average dead time on the channel was measured as explained above. The amount of dead time correlates very well with the channel utilization under each loading condition. The sum of the dead time and channel utilization should be near 100%. In all three cases the sums are very close to 100%.

Not included in this sum are the percentage of time taken to resolve collisions and the percentage of time taken to switch from the controlled access to the random access mode. Also not included is the time spent counting through unused time slots in the controlled access mode. These three times can be quantified, however, by using data from previous tests. They are listed in table 3 for each of the three interarrival times.

Differences between the percentages of time expended and 100% may be a result of using average values for most of the parameters. The sums may be closer to 100% if the percent of dead time were calculated by summing the time between transmissions over all the messages that were sent in the random access mode, and dividing that sum by the run-time. The current method, however, uses the average dead time (calculated over 35 or 50 messages) and multiplies this by the total number of messages.

TABLE 3.- PERCENTAGE OF OPERATION TIME FOR VARIOUS ACTIVITIES

Activity	Interarrival time, μsec		
	700	1500	3000
Percent time to resolve collisions[a]	0.59	0.056	0.015
Percent time to switch modes[b]	.24	.024	.006
Percent time counting unused timeslots[c]	.38	.045	.013
Sum (%)	1.21	.125	.034

[a]Percent time to resolve collisions =

$$\frac{\text{Average time to resolve collisions (μsec)} \times \text{number of collisions}}{\text{run-time, μsec}} \times 100\%$$

[b]Percent time to switch modes =

$$\frac{\text{Maximum time to switch modes (μsec)} \times \text{number of collisions}}{\text{run-time, μsec}} \times 100\%$$

[c]Percent time counting through unused timeslots:

$$\text{Average number of BIUs/collision} = \frac{\text{number of messages colliding}}{\text{number of collisions}}$$

Average number of timeslots unused/collision =
total number of BIUs - average number of BIUs/collision

Average time counting through unused timeslots, μsec =
average number of timeslots unused/collision × number of collisions
× 1.6 μsec/timeslot

$$\text{Percent time} = \frac{\text{Average time counting through unused timeslots, μsec}}{\text{run-time, μsec}} \times 100\%$$

Test #5: Independence of the Message Length

Objective

The objective of this test was to verify that the load interarrival time, as measured from the data, is independent of the message length.

Approach

Run three simulations, varying only the message lengths. Select the message interarrival time so that the channel utilization is near 80% for the run with the largest message length (message interarrival time = 800 μsec/BIU).

The maximum message length in version 2 is 16496 bits (the maximum allowed by the FODS protocol specification), and in version 2a it is 16384 bits. The other two message lengths are 7900 and 550 bits. The minimum value is chosen by taking 1/30 of the maximum. As explained in the conclusions-version 2a section of Test #1, the maximum message length was changed from 16496 bits in version 1 to 16384 bits in version 2a. The change was made to make the results depend only on the size of the data, not on the size of the header and data.

The number of nodes is four. The run-time is 100000 μsec, allowing about 125 messages/node/run.

Results

Version 2

VERSION 2 RESULTS FOR TEST #5

	Message length, bits		
	550	7900	16496
Calculated message interarrival time[a]	846	832	868

[a]Calculated message interarrival time, μsec =

$$\frac{\text{Message length (bits)}}{\text{throughput, Mbit/sec}} \times \text{number of BIUs}$$

(Throughput must be divided by the number of BIUs here since it is the system throughput, and the message interarrival time is for each BIU.)

Version 2a

VERSION 2a RESULTS FOR TEST #5

	Message length, bits		
	550	7900	16384
Calculated message interarrival time[a]	846	839	866

[a]Calculated message interarrival time, μsec =

$$\frac{\text{message length, bits}}{\text{throughput, Mbits/sec}} \times \text{number of BIUs}$$

(Throughput must be divided by the number of BIUs here since it is the system throughput, and the message interarrival time is for each BIU.)

Conclusions

Version 2

This test is also invalidated because of the error described in the introduction. Because the calculated message interarrival time is determined using the throughput, which was not correctly calculated, these results are not correct.

Version 2a

The message interarrival time is independent of message length, as expected. Although the measured message interarrival times are not equal, they are very close to each other. Furthermore, the differences between the measured interarrival times and those specified are small and can be attributed to the random nature of the program.

SECTION 2: SUMMARY OF PERFORMANCE CHARACTERISTICS

Table 4 summarizes the performance characteristics of the FODS protocol in version 2a. The loading is given in terms of the message interarrival time and the number of bits offered to the bus. These are not absolute results. Because of the inherently random nature of the simulation program, the same results are not obtained each time the program is used. These results are, however, representative of the types of data which can be obtained from the program.

TABLE 4.- PERFORMANCE SUMMARY--VERSION 2a

Ontime, μsec	Load, μsec	Load, Mbps	Packets sent	Maximum queue	Number of collisions	Throughput, Mbps
30000	700	93.63	173	11	44	94.48
60000	700	93.63	343	12	90	93.66
90000	700	93.63	487	7	117	88.66
120000	700	93.63	658	11	144	89.84
30000	800	81.92	144	5	31	78.64
60000	800	81.92	288	5	62	78.64
90000	800	81.92	446	7	90	81.19
120000	800	81.92	600	5	124	81.92
30000	1000	65.54	115	4	15	62.81
30000	1200	54.61	99	3	11	54.07
30000	1400	46.81	81	2	4	44.24
30000	1600	40.96	65	2	2	35.50
30000	1800	36.40	62	2	4	33.86
30000	2000	32.77	59	2	0	32.22
100000	5000	13.11	74	2	0	12.12
150000	10000	6.55	53	2	0	5.79

CONCLUSIONS

Five verification tests were performed on the simulation program HYCAR, using the FODS protocol. The five tests were as follows:

(1) Verify the throughput and channel efficiency;

(2) Check the number of collisions;

(3) Measure the average time to resolve collisions;

(4) Measure the average dead time on the channel;

(5) Verify the independence of the message length.

When these tests were conducted, two bugs were found in the program. In version 1 of the program, the first bug caused an improper counting sequence in the controlled access mode, resulting in collisions in that mode. In version 2, the second bug caused incorrect calculations of the throughput and efficiency. Both problems were successfully resolved in version 2a.

The second bug in the program appeared most obviously in test #1. Once the bug was resolved, however, the calculations of throughput and efficiency were consistent with the expected results. The data were comparable to the mean offered load.

In test #2, the number of collisions was counted. Also, the actual number of messages between collisions was compared to the calculated upper limit of four messages between each collision. Under heavy loading conditions (90% utilization), this upper limit was met. Under minimal loading conditions (20% utilization), this value decreased to 30 messages between each collision.

The objective of test #3 was to measure the average amount of time needed to resolve collisions. The actual data were compared with the calculated minimum, average, and maximum values. The actual data were slightly lower than those calculated because the calculations were made under the assumption that collisions involving any number of nodes occur with the same frequency. The simulation data show, however, that collisions involving two nodes are much more frequent than those involving more nodes. Nevertheless, all average and individual values were within the calculated limits.

In test #4, correlation between the amount of dead time on the channel and the channel utilization was attempted. These parameters were well correlated. To make the correlation better, the following parameters were also quantized:

(1) The percentage of time taken to resolve collisions;

(2) The percentage of time taken to switch from the controlled access to the random access mode;

(3) The percentage of time spent counting through unused time slots.

Differences between the sum of the calculated percentages and 100% may be a result of using average values for the parameters.

The objective of test #5 was to verify that the load interarrival time and the message length were independent parameters. This test was successful. Differences between the measured and the specified interarrival times were very small. Furthermore, these differences could be attributed to the random nature of the program.

Overall, the five tests were successful in validating the simulation program, HYCAR. Bugs, which otherwise may not have been discovered until much later, were found and corrected. The tests also helped in summarizing the performance of the FODS protocol. The results presented in section 2 are representative of the information which can be obtained from this simulation program.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-86660		

4. Title and Subtitle	5. Report Date
VERIFICATION AND PERFORMANCE TESTS OF HYCAR PROGRAM	June 1985
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Veena Bhatia	85040
	10. Work Unit No.
9. Performing Organization Name and Address	
Ames Research Center Moffett Field, CA 94035	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code 506-58-11

15. Supplementary Notes

Point of Contact: Veena Bhatia, Ames Research Center, MS 244-7,
Moffett Field, CA 94035 (415) 694-6520 or FTS 464-6520

16. Abstract

HYCAR is a program that simulates the network protocols of HYPERchannel and FODS (Fiber Optic Demonstration System), as well as other related protocols. Verification tests of the program were conducted using the FODS protocol, which was developed under contract by Sperry Flight Systems, Phoenix, Arizona. The tests validated the operation of the program through deterministic and analytical means. Extensive experimentation with the simulator produced a set of performance characteristics for the FODS protocol under varied loading conditions. These characteristics were consistent with those expected, and are documented with the validation tests.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computer networking Network simulation	[redacted] Subject category — 32

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	25	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161