Briefing Slides for a Feasibility Study of
Digitized Voice Distribution via
the XPress Transfer Protocol

Alfred C. Weaver
James McNabb
John Fenton
Robert Simoncic

Computer Science Technical Report No. TR-92-05
February 11, 1992

A Feasibility Study of Digitized Voice Distribution via the Xpress Transfer Protocol

Alfred C. Weaver
James McNabb
John Fenton
Robert Simoncic

Computer Networks Laboratory
University of Virginia
December 2, 1991

Telephone Systems

Telephone systems started moving from analog to digital signaling in the 1980s

- bit error rate for analog signaling over copper is about 10^{-5}; digital transmission over optical fiber is about 10^{-12}

- digital signaling permits mixed voice, computer data, music, television, facsimile, video telephone, etc. to be multiplexed over same circuits

Codecs make 7-bit (U.S.) or 8-bit (Europe) samples at 8000 Hz (Nyquist sampling theorem)

One sample every 125 microseconds

24 voice channels multiplexed together onto a T1 channel (1.544 Mbits/sec)

Computer Systems

Digital voice is a special case of digital data—special because it has different *timing* requirements

Potentially advantageous to combine data and voice

- lower bit error rate
- higher system reliability
- integration of voice, video, and data
- reuse of existing network components
- shared use of single cable plant

Security advantages? Potential reduction of three separate systems (voice, video, data) into one

Digitized Voice Distribution

Voice can not be transmitted sample-by-sample—too much overhead

Voice must first be *packetized*

Steps:

- continuous analog-to-digital conversion
- voice output saved in FIFO
- *n* voice samples collected by user program
- delivered to communications subsystem
- processed by transport and network protocols
- encapsulated within FDDI frame
- physical transmission
- routed over the internetwork
- received as FDDI frame
- processed by network and transport protocols
- data delivered to user process
- enqueued at output FIFO
- continuous digital-to-analog process

Performance Goals

Performance goals are simply stated:

- *empty the A/D converter's FIFO sufficiently often that it never overflows*

- *deliver that data to the D/A converter's FIFO sufficiently often than it never underflows*

These are trivial for single channel—but makes the world's most expensive telephone !

Practical systems system must handle multiple voice channels simultaneously

Highly desirable:

- commercial (vs. proprietary) components

- standard computer networks

- standard computer communications protocols

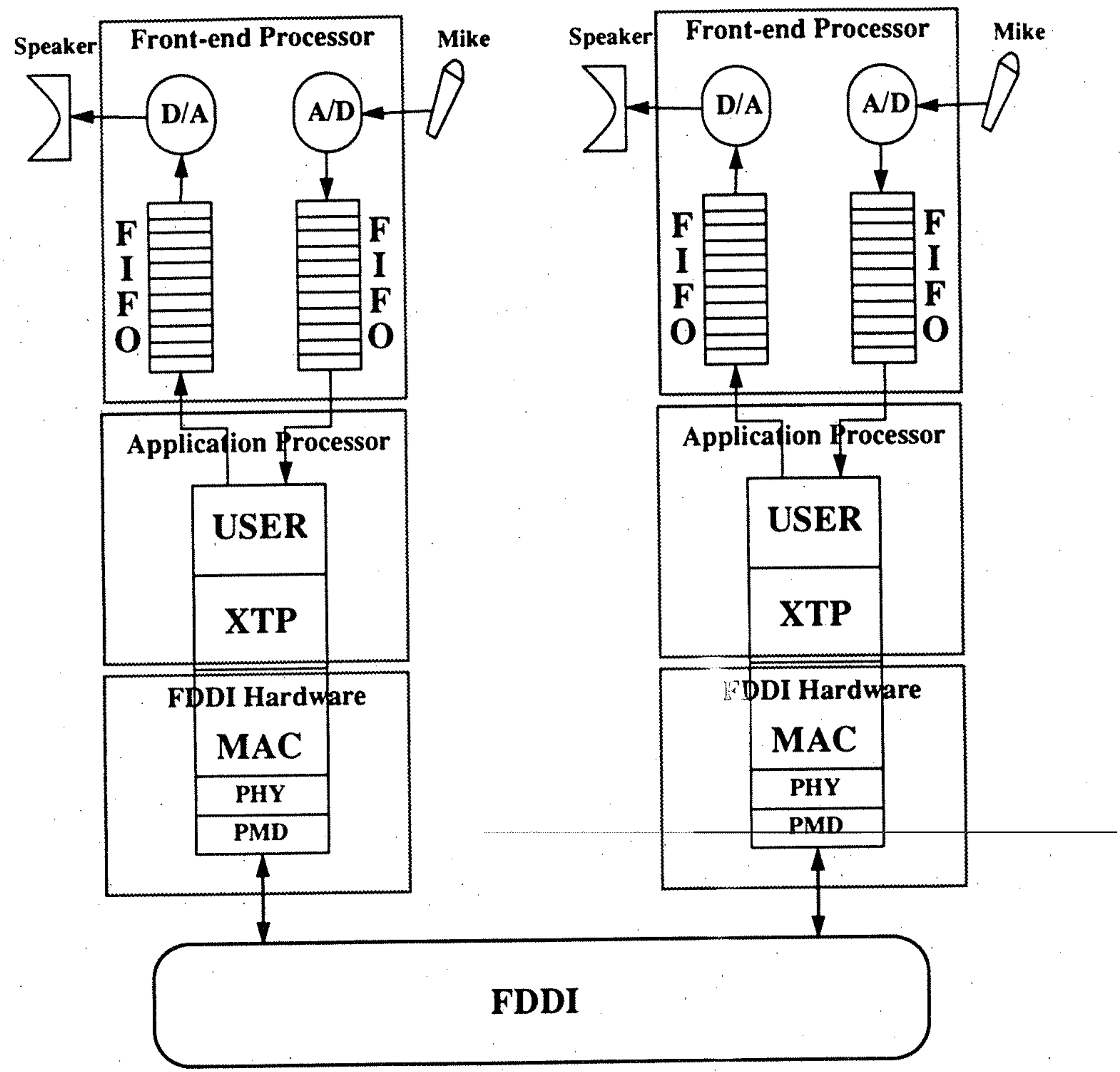


Figure 1.

Architectural Concept

Role of the Communications Protocols

Purpose of protocols: move a stream of data from a continuous source (A/D converter) to a continuous sink (D/A converter)

User collects *n* voice samples and delivers them as a group to the transport protocol (TSDU)

Transport protocol accepts whatever data has collected and processes it (TPDU)

> *Value added:* sequential, in-order delivery without duplicates; transparent error recovery; end-to-end reliability

Network protocol prepends routing information

> *Value added:* packet can be routed over an internet

Datalink protocol frames and transmits data

> *Value added:* link level error detection

Which Protocols to Use?

Transport

- ISO Transport Protocol class 4 (TP4)
- Transmission Control Protocol (TCP)
- Xpress Transfer Protocol (XTP)

Network

- ISO Connectionless Network Layer Protocol (CLNP)
- Internet Protocol (IP)
- Xpress Transfer Protocol (XTP)

Datalink and Physical Layer

- FDDI
- others?

Our task was to investigate the feasibility of using XTP

Experiment Design

Hardware and software

- Motorola 133XT (25 MHz 68020, 4 MB)
- VMEbus backplane
- pSOS real-time operating system
- Martin Marietta FDDI
- Microtec C compiler
- XTP

No physical A/D or D/A converters

No physical microphones or speakers

All voice in FDDI *synchronous* class

Default parameters for AMD SuperNet FDDI

No heroic attempts to increase performance

Experiment Overview

Measurements: throughput, latency, jitter

Primary variable: number of voice samples in a packet (n)

Scenarios:

- basic experiment
- background synchronous FDDI traffic (SPPT)
- background synchronous FDDI traffic (MPPT)
- retransmission
- background processor load (asynchronous)
- multicast
- routers

Commentary:

- datalink vs. transport protocols
- XTP vs. other transport/network protocols
- conclusions

Experimental Results

Timing

Timer resolution is 81.3 microseconds

Reading the clock makes time appear to be discrete rather than continuous

Leads to a "quantum effect" in jitter plots

But timer resolution is adequate—even our shortest measured event is 25 timer ticks

No significant effect on any of our measurements or conclusions

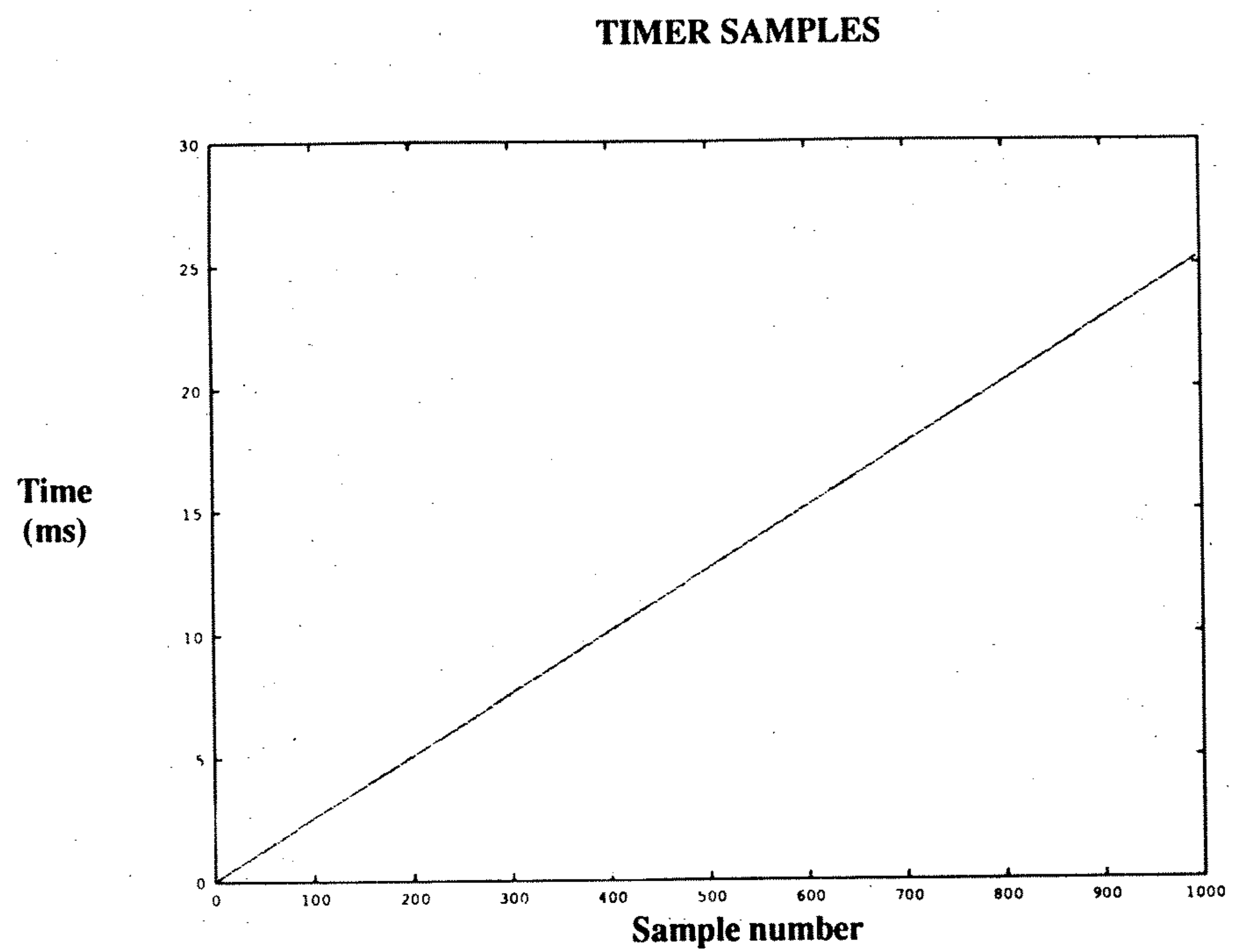


Figure 2.

1000 Timer Samples

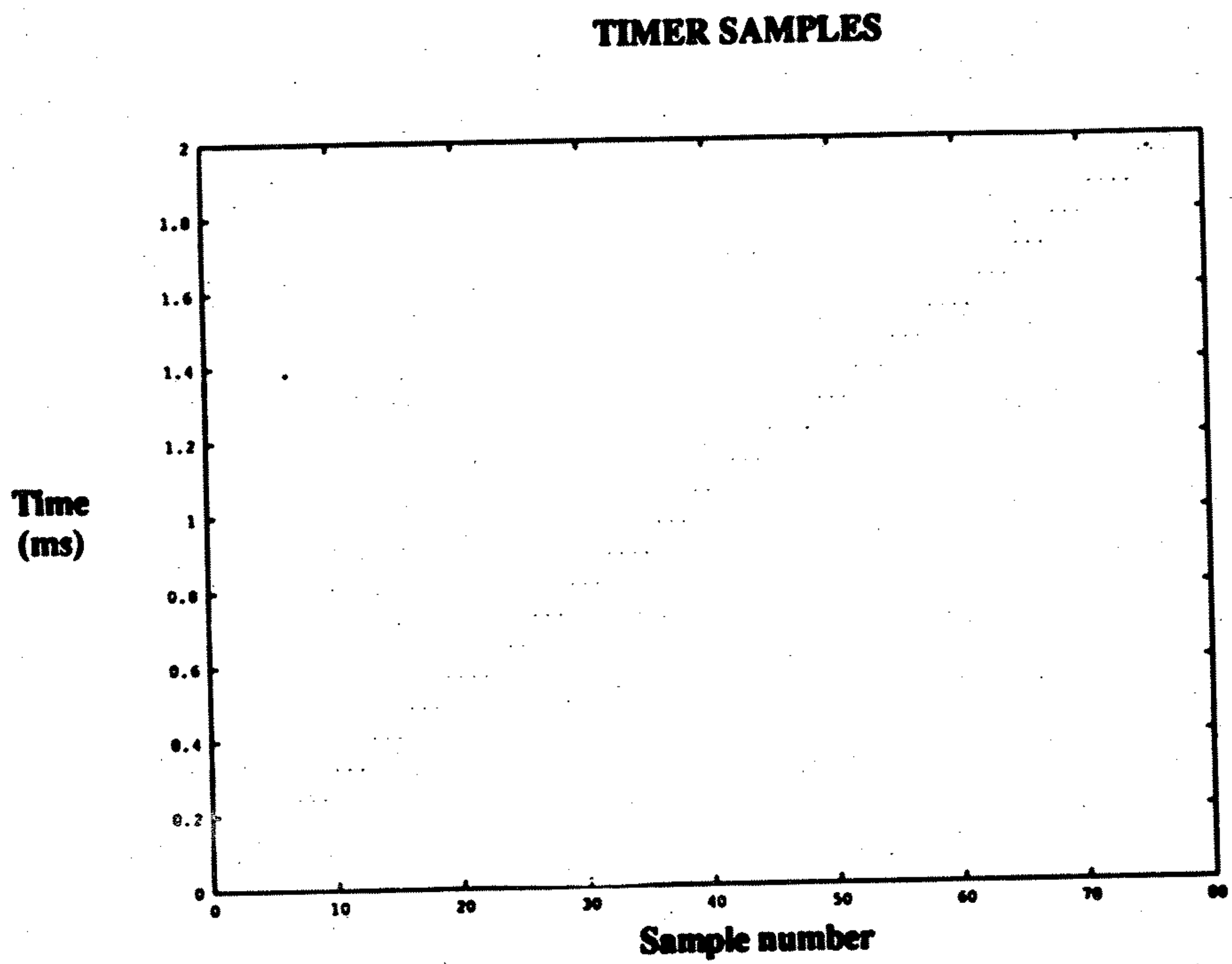


Figure 3.

80 Timer Samples

Experiment 1: Basic Experiment

Scenario:

- read *n* bytes from pseudo-FIFO
- deliver data to XTP
- run transport protocol
- transmit FDDI frame
- receive FDDI frame
- run XTP
- deliver data to user
- write data to pseudo-FIFO

Continue until one megabyte of user data has been moved

Repeat for $n = 8, 16, 32, ..., 2048, 4096$

No background load on FDDI

No background load on processor

Represents a "best case" for given architecture (before any optimizations)

Some Details

Data delivered to XTP in "chunks" of size *n*

XTP adds 44 bytes (40-byte header, 4-byte trailer)

FDDI adds 25 bytes (8 for LLC, 6 for destination MAC, 6 for source MAC, 4 for CRC, 1 for frame status)

XTP allows programmable reliability—we chose fully reliable

XTP permits user to select acknowledgement frequency—we chose one acknowledgement per outgoing data packet

We did not optimize!

We are documenting the cost of the most robust service, knowing that many optimizations are available if needed

BASIC EXPERIMENT

voice data size (bytes)	frame size (bytes)	packets sent	user throughput (Mbits/sec)	network throughput (Mbits/sec)	total time (sec)	packet rate (packets/sec)	average latency (ms)	99.9% threshold (ms)
8	77	131,072	0.030	0.288	278.890	470	2.728	2.764
16	85	65,536	0.060	0.319	139.440	470	2.732	2.846
32	102	32,768	0.120	0.383	69.890	469	2.751	2.846
64	133	16,384	0.237	0.493	35.330	464	2.791	2.927
128	197	8,192	0.463	0.713	18.100	453	2.890	3.008
256	325	4,096	0.890	1.130	9.360	438	3.062	3.171
512	581	2,048	1.651	1.873	5.080	403	3.417	3.496
1,024	1,093	1,024	3.039	3.244	2.760	371	4.015	4.146
2,048	2,117	512	5.053	5.223	1.667	307	5.242	5.366
4,096	4,165	256	7.557	7.684	1.110	231	7.699	7.805

69 bytes framing overhead (44 bytes from XTP, 25 from FDDI)
no background processor load
no background FDDI load

Table 1
Basic Experiment: Throughput and Latency

Experiment 1: Analysis

Overhead

varies from 90% (8 bytes payload in 77 byte frame) to 1.5% (4096 bytes payload in 4165 byte frame)

network efficiency argues for largest possible voice data size

other factors make very large packets less attractive

Packet rate

maximum rate is 470 packets/sec

minimum period is 2.1 ms

Latency

average latency varies from 2.7 ms (8 bytes) to 7.7 ms (4096 bytes)

99.9% threshold differs from average by at most 0.1 ms

required vs. observed arrival periods in Table 2

BASIC EXPERIMENT

voice data size (bytes)	new data needed every (ms)	new data generated every (ms)	packet generation rate (packets/sec)
8	1.000	2.127	470
16	2.000	2.127	470
32	4.000	2.132	469
64	8.000	2.155	464
128	16.000	2.208	453
256	32.000	2.283	438
512	64.000	2.481	403
1,024	128.000	2.695	371
2,048	256.000	3.257	307
4,096	512.000	4.329	231

69 bytes framing overhead
no background processor load
no background FDDI load

Table 2
Basic Experiment: Required vs. Observed Arrival Periods

BASIC EXPERIMENT

voice data size (bytes)	user throughput (Kbits/sec)	equivalent voice channels (64 Kbits/sec each)
8	30	0
16	60	0
32	120	1
64	238	3
128	468	7
256	897	14
512	1661	25
1024	3050	47
2048	5115	79
4096	7294	113

69 bytes framing overhead
no background processor load
no background FDDI load

Table 3
Basic Experiment: Voice Channels Available

Experiment 1: Analysis

Voice channels

smallest voice data size fails completely

largest voice data size permits 113 voice channels

channel startup delay is proportional to voice data size (n*125 microseconds plus delivery latency)

startup delay suffered only on connection establishment

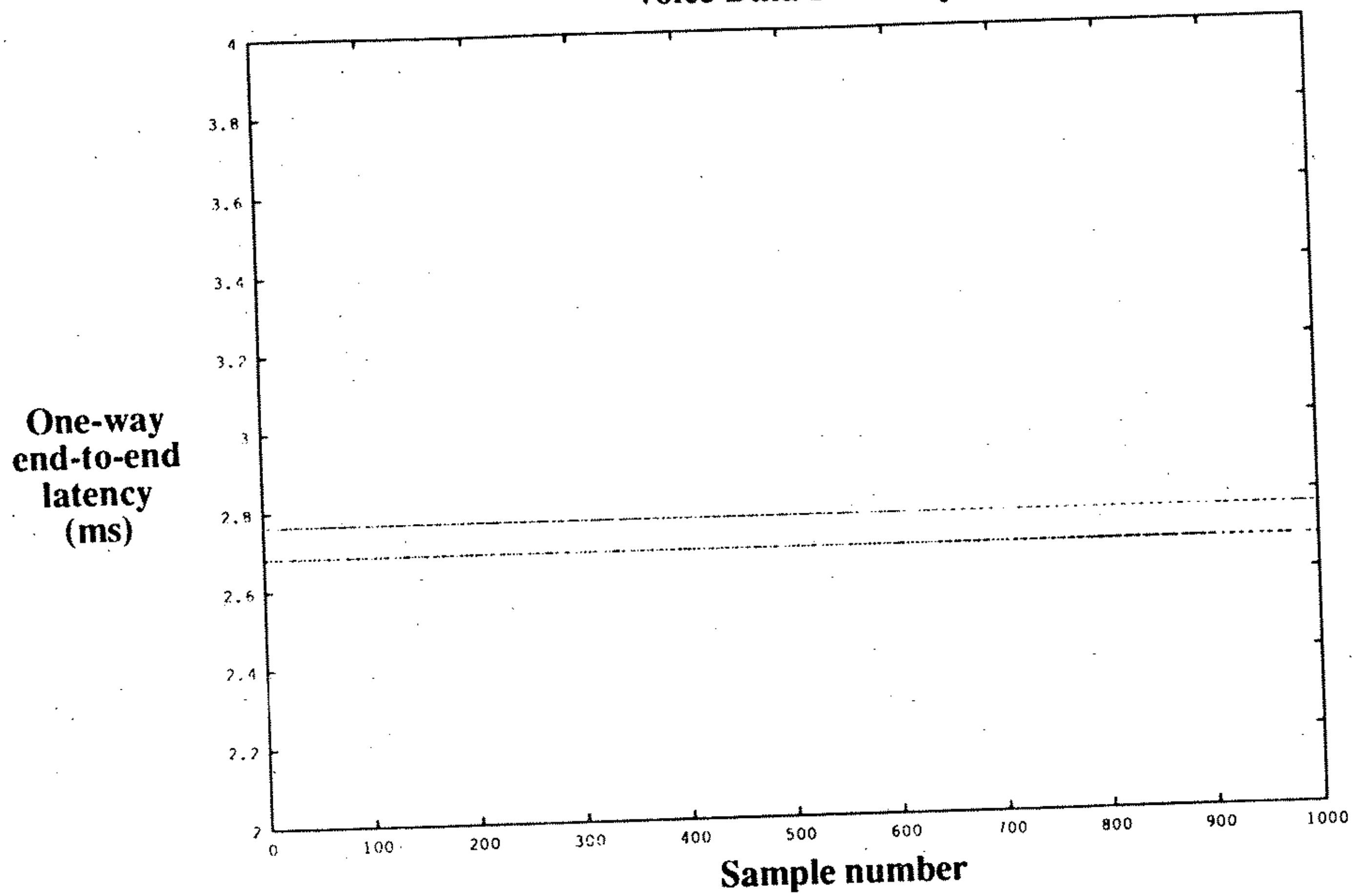
Jitter

ten jitter plots, one per voice data size

average vs. 99.9% latency

JITTER MEASUREMENT

Voice Data Size: 8 bytes



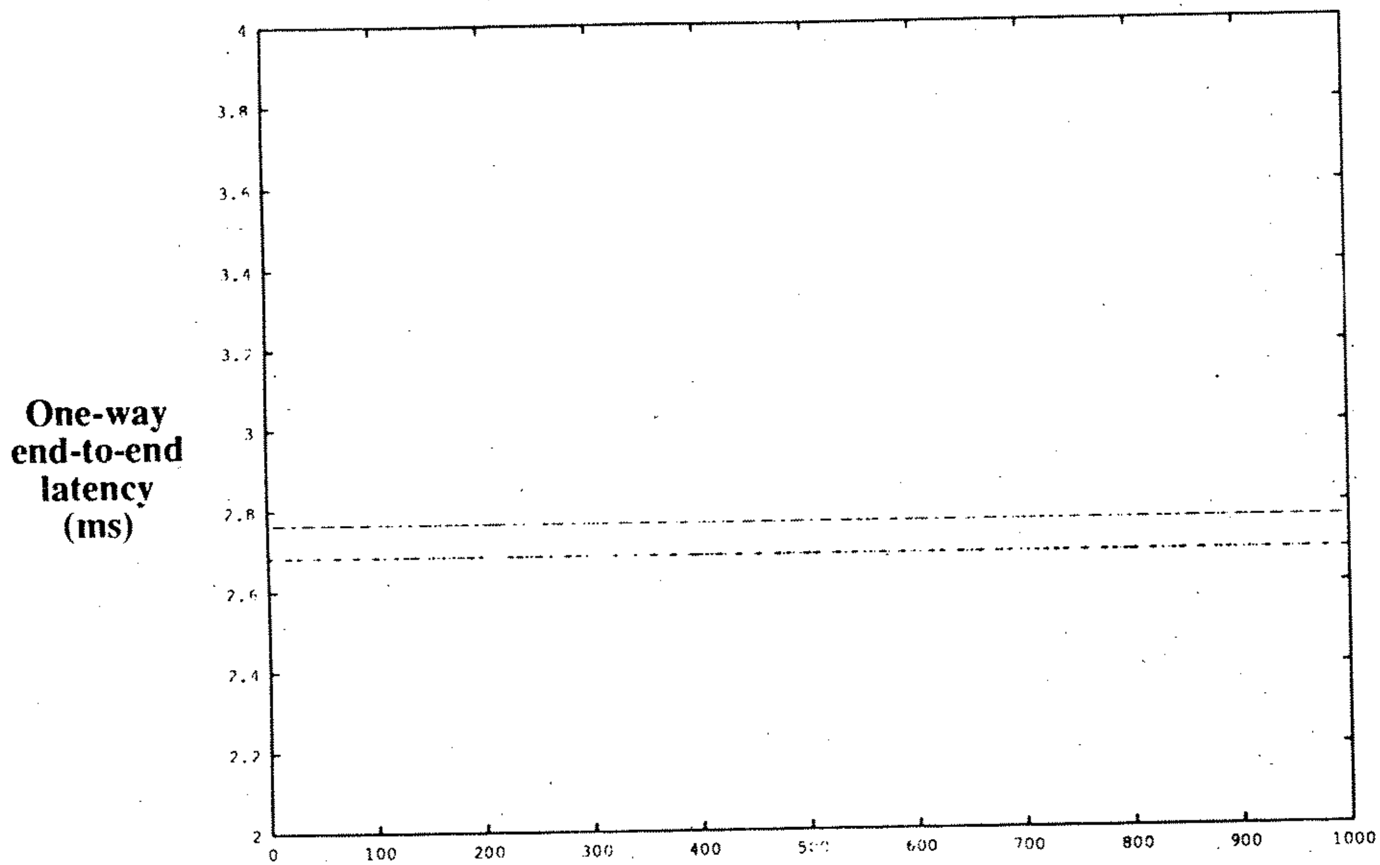
Average latency:	2.728 ms
99.9% threshold:	2.764 ms

1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 16 bytes



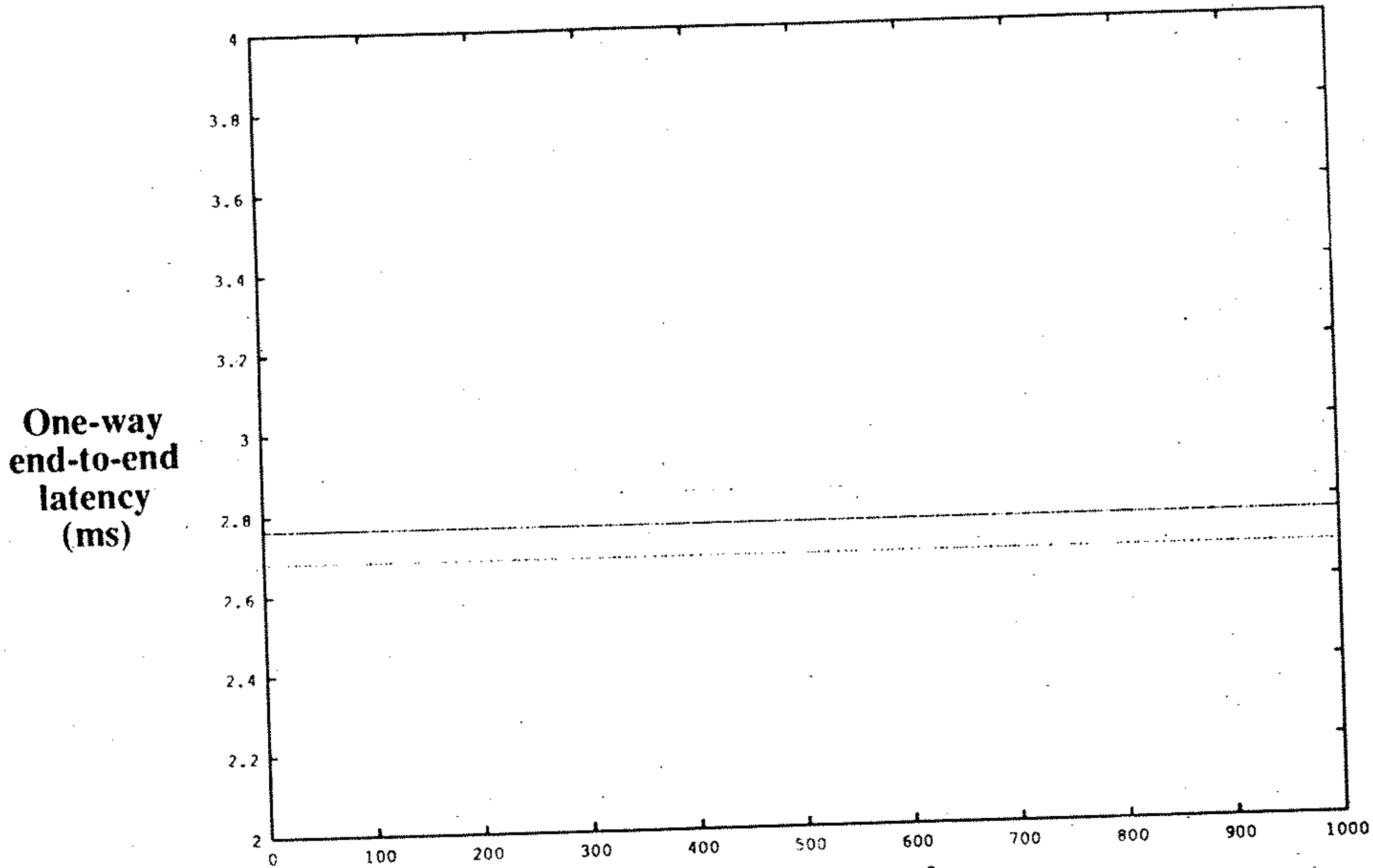
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 2.732 ms
99.9% threshold: 2.846 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 32 bytes



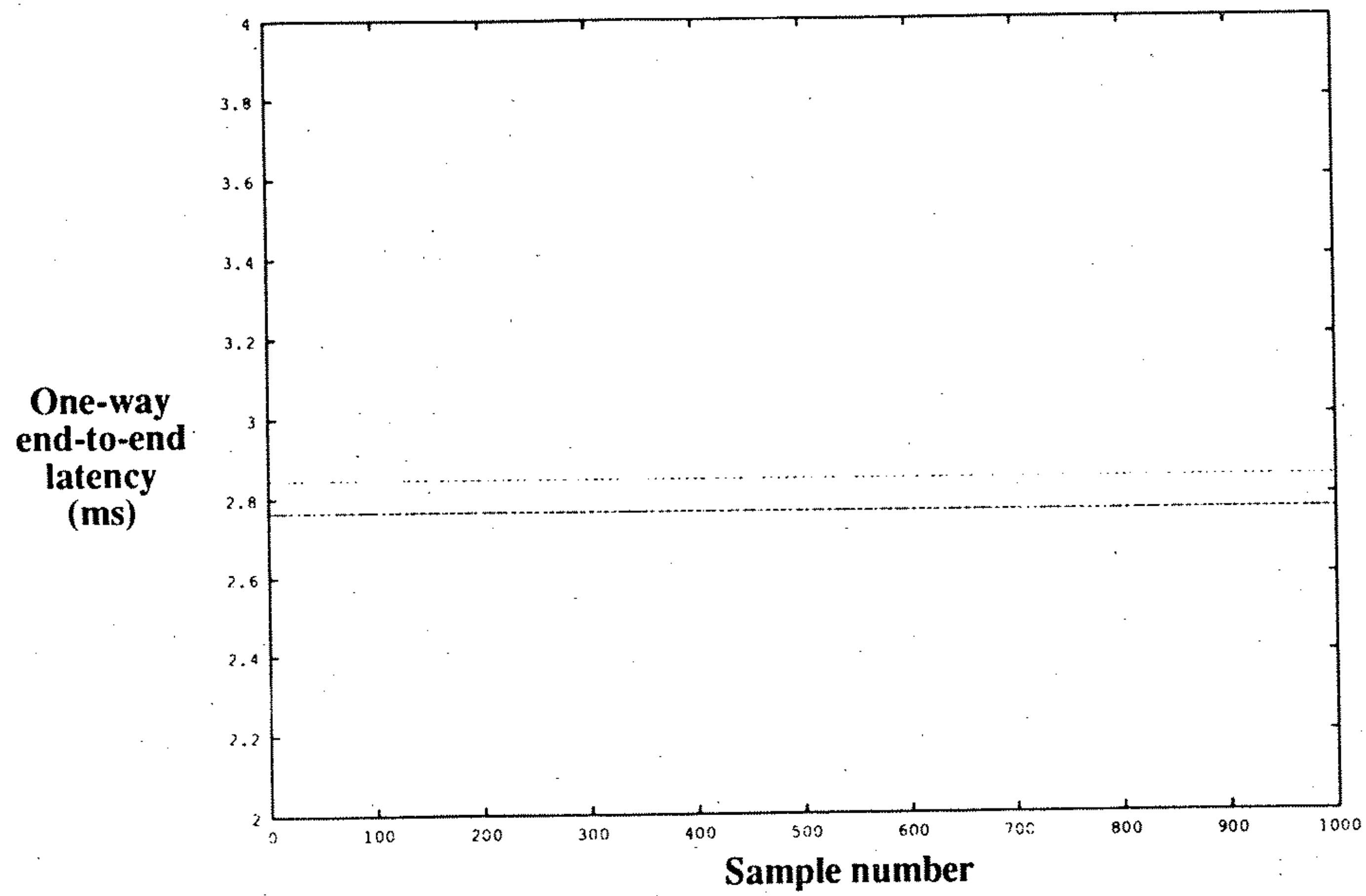
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 2.751 ms
99.9% threshold: 2.846 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 64 bytes



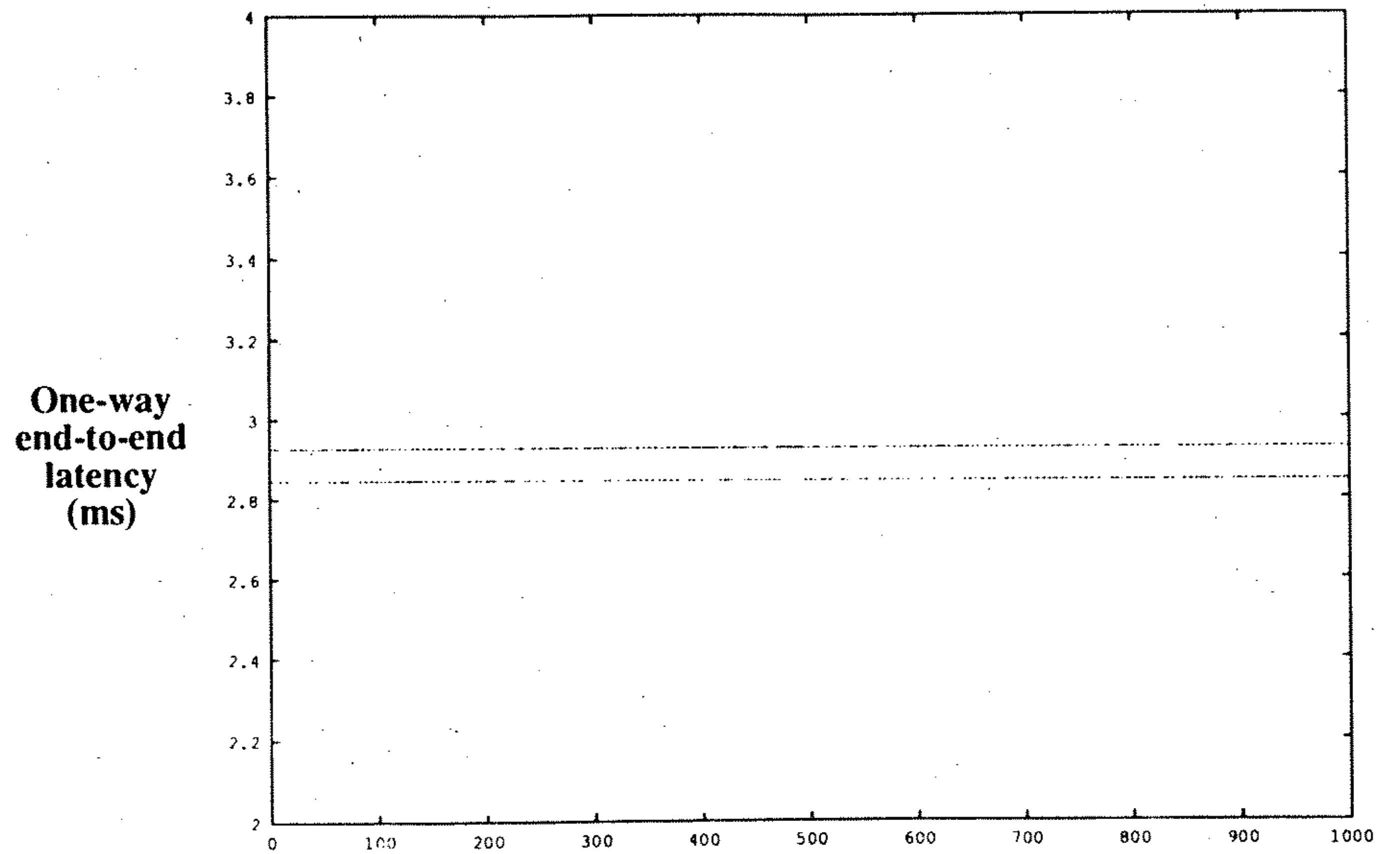
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 2.791 ms
99.9% threshold: 2.927 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 128 bytes

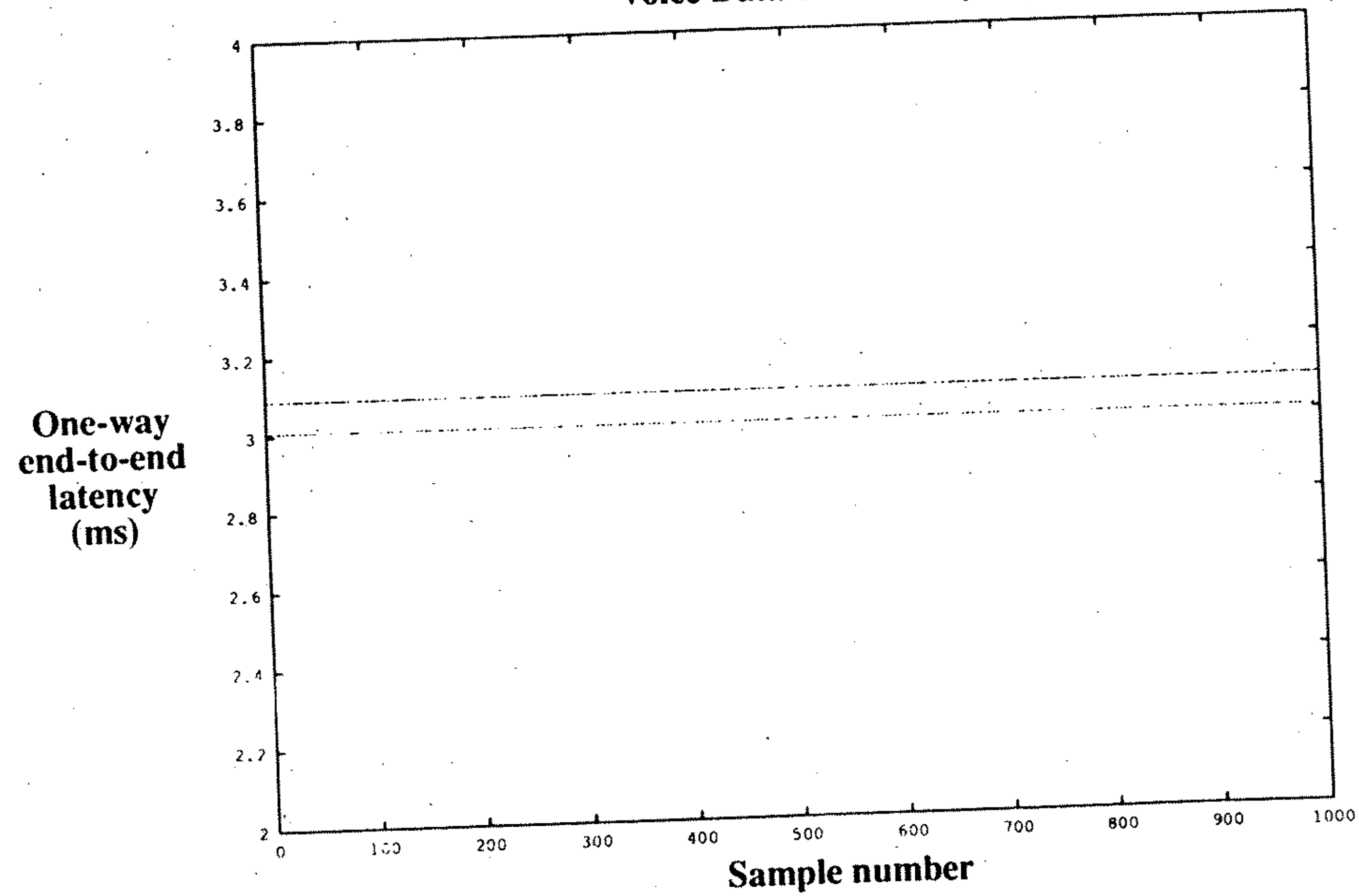


1000 samples	Average latency: 2.890 ms
No asynchronous processor load	99.9% threshold: 3.008 ms
No background FDDI load	
Voice data in FDDI synchronous class	

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 256 bytes



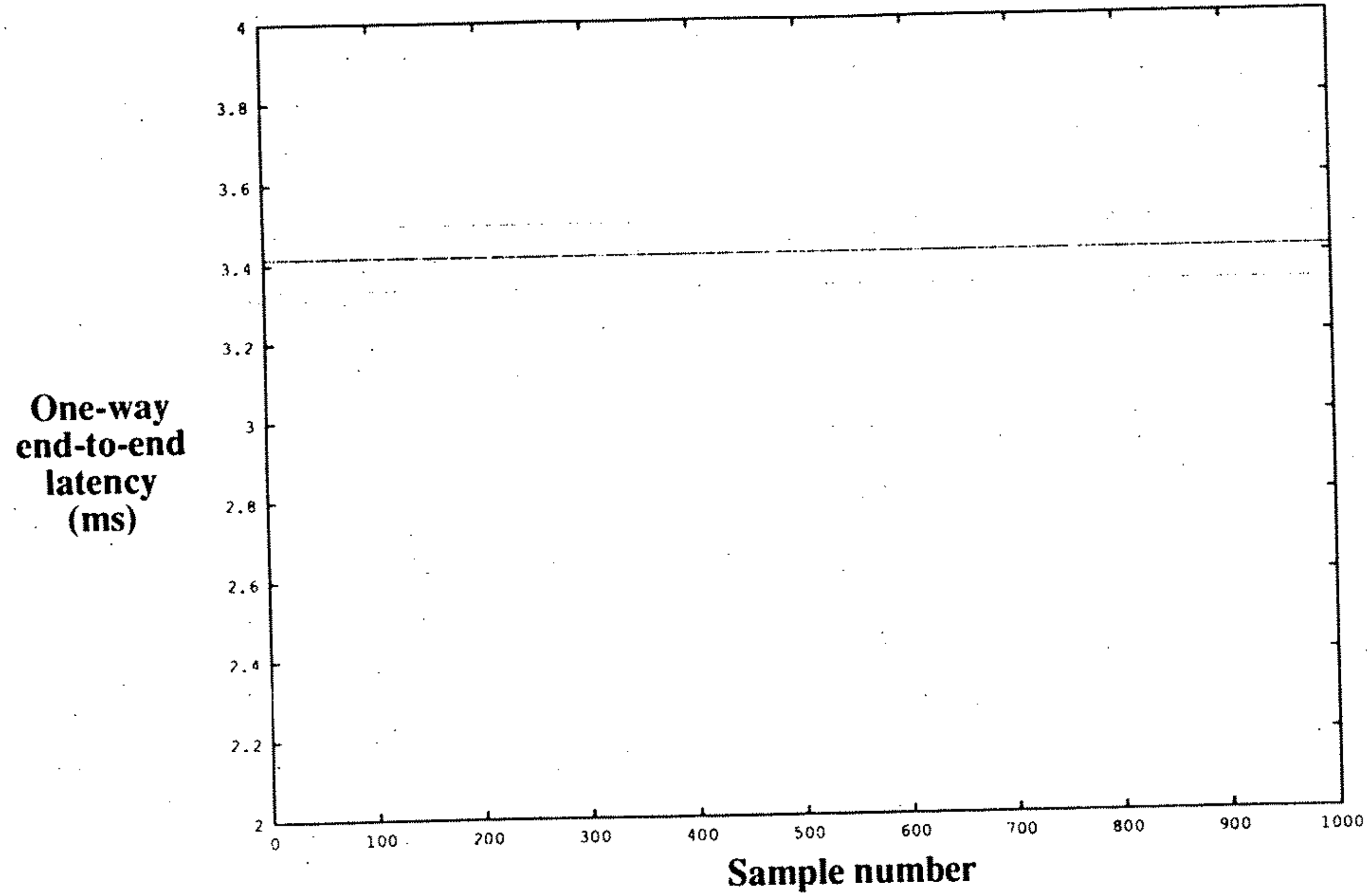
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 3.062 ms
99.9% threshold: 3.171 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 512 bytes



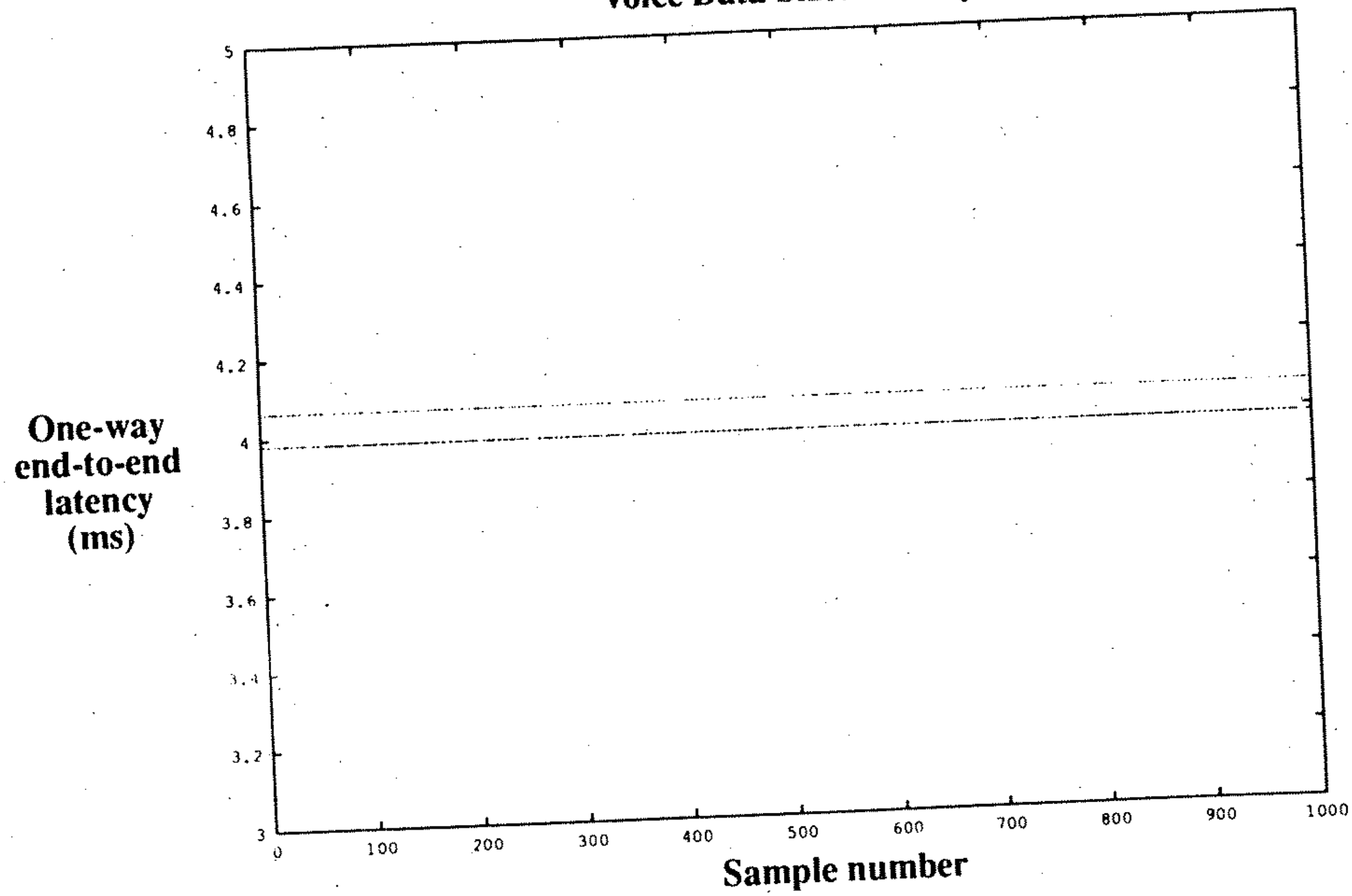
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 3.417 ms
99.9% threshold: 3.496 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



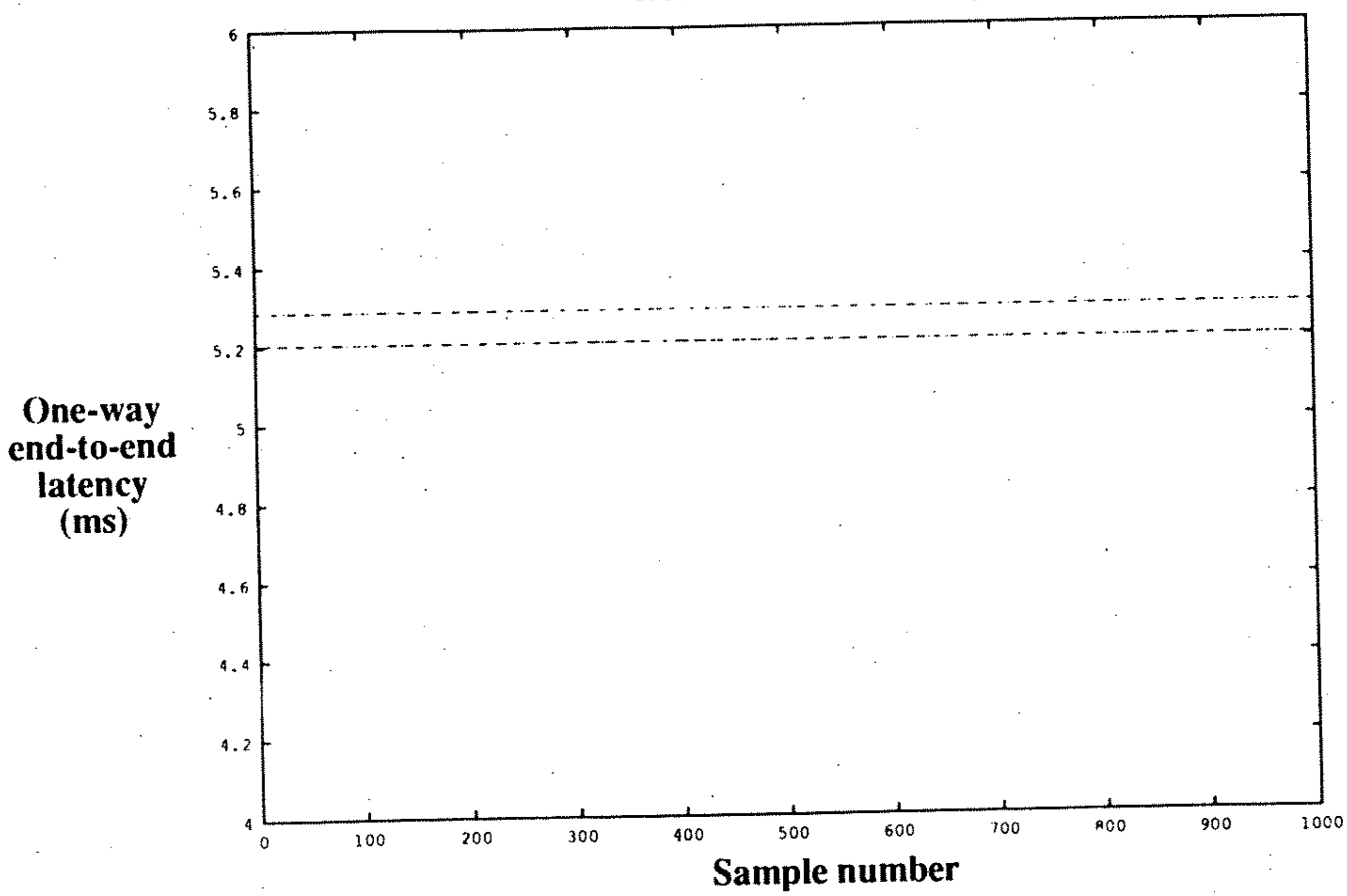
Average latency: 4.015 ms
99.9% threshold: 4.146 ms

1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 2048 bytes



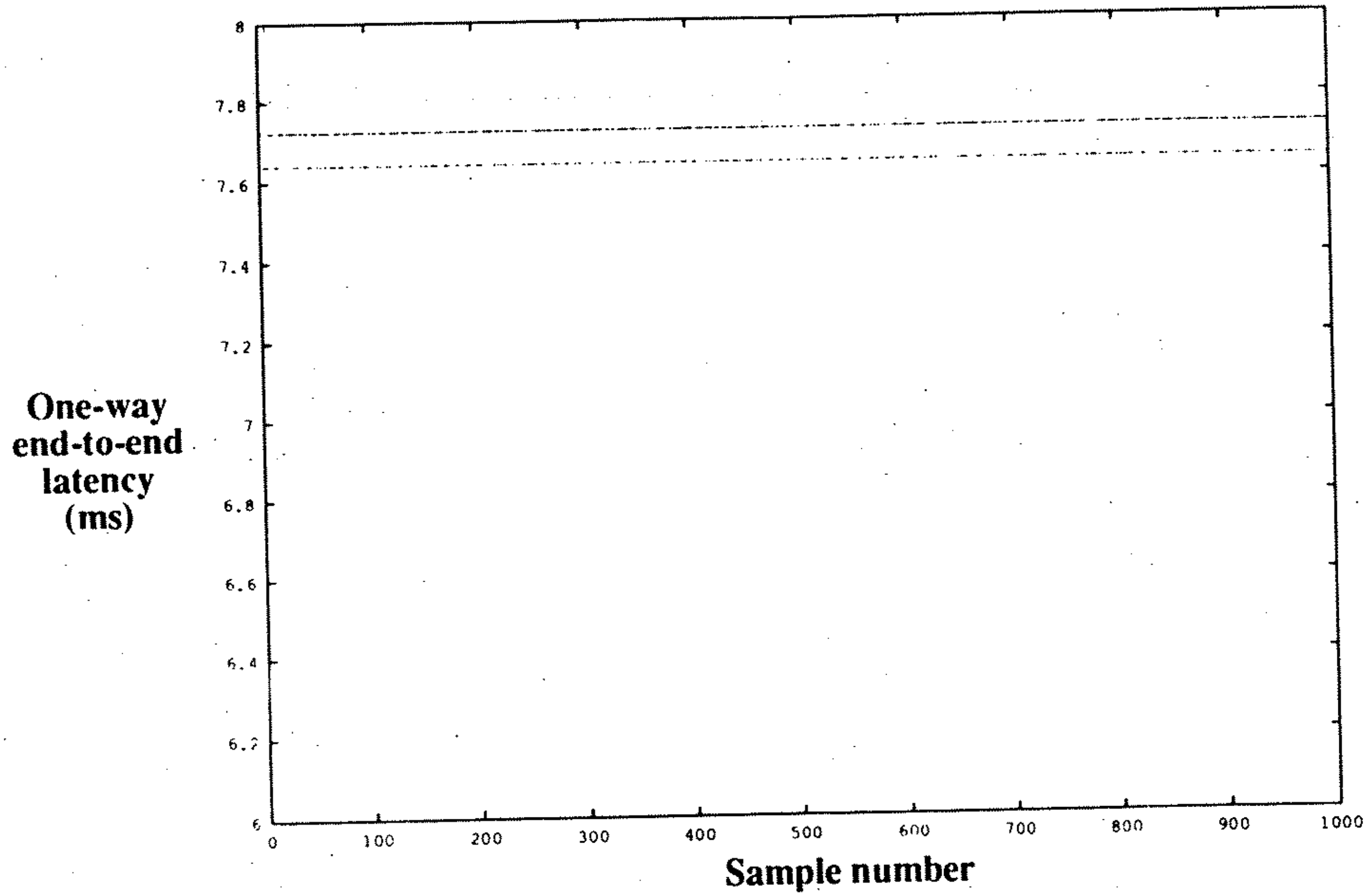
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 5.242 ms
99.9% threshold: 5.366 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 4096 bytes



1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 7.699 ms
99.9% threshold: 7.805 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

BASIC EXPERIMENT

voice data size (bytes)	average latency (ms)	99.9% threshold (ms)
8	2.728	2.764
16	2.732	2.846
32	2.751	2.846
64	2.791	2.927
128	2.890	3.008
256	3.062	3.171
512	3.417	3.496
1,024	4.015	4.146
2,048	5.242	5.366
4,096	7.699	7.805

69 bytes framing overhead
no background processor load
no background FDDI load

Table 4
Basic Experiment: Average and Threshold Latency

Experiment 1: Conclusions

1. System efficiency increases with increasing voice data size.

2. The side effect which argues against making the voice data size arbitrarily large is that the connection startup time is proportional to voice data size, and in fact is equal to n* 125 microseconds plus the end-to-end latency.

3. Payloads of 32 bytes or larger are practical for a single voice data channel.

4. The number of simultaneous voice channels which can be supported likewise increases with voice data size. Voice data sizes of 1K, 2K, and 4K bytes would support 47, 79, and 113 voice channels (64 Kbits/sec each) respectively.

5. Jitter plots confirm that there is very little variance in the end-to-end delivery latency.

6. All the throughput, latency, and jitter data reported for the basic experiment represent a "best case" scenario for the chosen architecture. Additional experiments are required to determine the impact of background FDDI load, background processor load, etc.

Experiment 2: Synchronous SPPT Background FDDI Load

Traffic generator (TG) was PC with AMD FastCard—its sole purpose was the make the FDDI ring look busy

TG created FDDI frames of length 4167 bytes (33,336 bits) (330microseconds each to transmit)

Generation could be single-packet-per-token (SPPT) or multiple-packet-per-token (MPPT)

We used SPPT here, MPPT later

TG parameters:

Packet size (bits)	Packet generation rate (packets/sec)	Total offered load (Mbits/sec)
33336	750	25
33336	1500	50
33336	2250	75

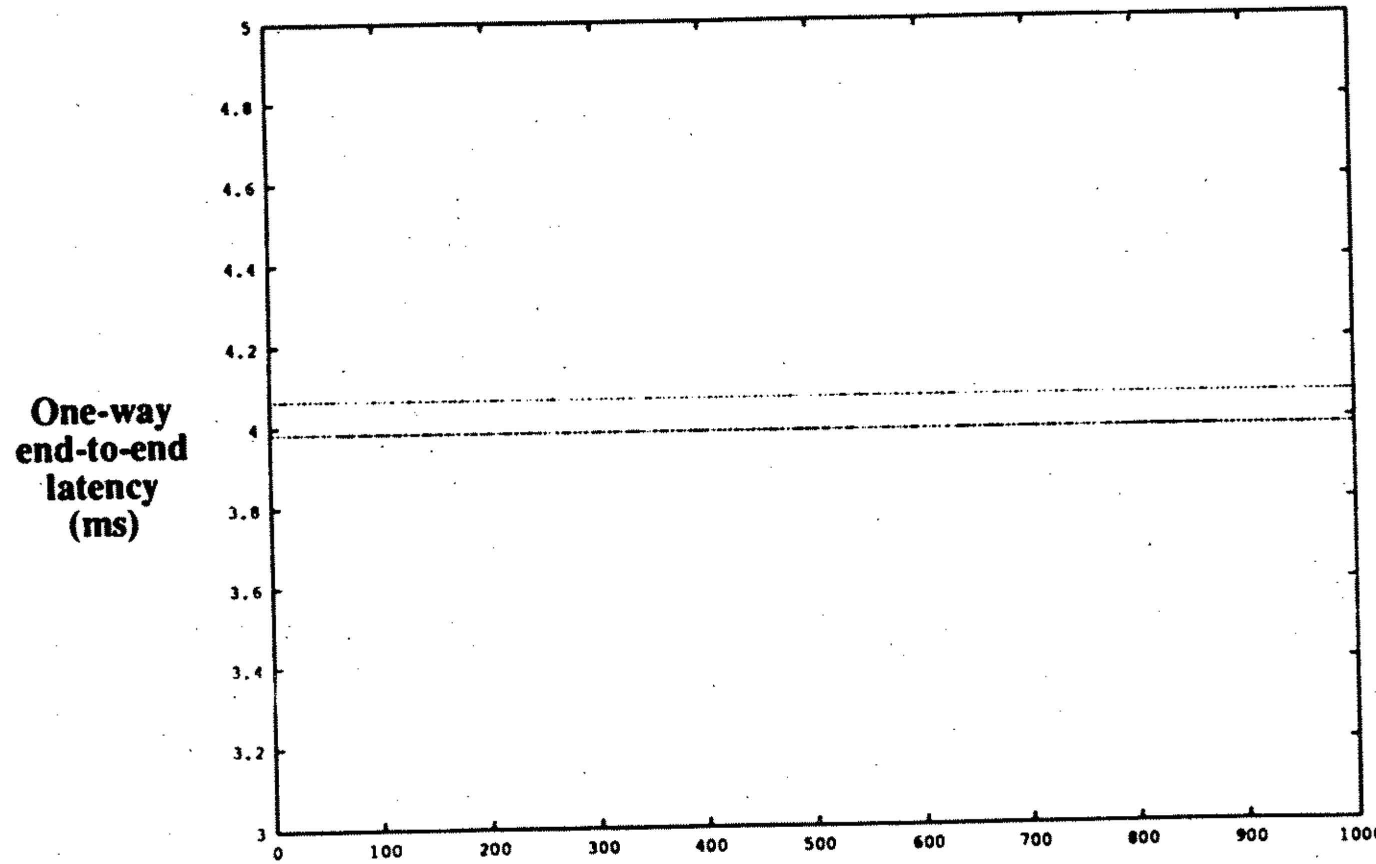
SYNCHRONOUS SPPT BACKGROUND LOAD

voice data size (bytes)	99.9% threshold latencies (ms)			
	0 Mbits/sec	25 Mbits/sec	50 Mbits/sec	75 Mbits/sec
8	2.764	2.927	2.927	3.171
16	2.846	2.927	2.927	3.171
32	2.846	2.927	3.089	3.171
64	2.927	3.008	3.008	3.252
128	3.008	3.089	3.089	3.496
256	3.171	3.333	3.659	3.659
512	3.496	3.740	3.821	3.821
1,024	4.146	4.228	4.390	4.634
2,048	5.366	5.528	5.610	5.772
4,096	7.805	7.967	8.049	8.221

Table 6
99.9% Threshold Latency with Synchronous SPPT Background FDDI Load

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



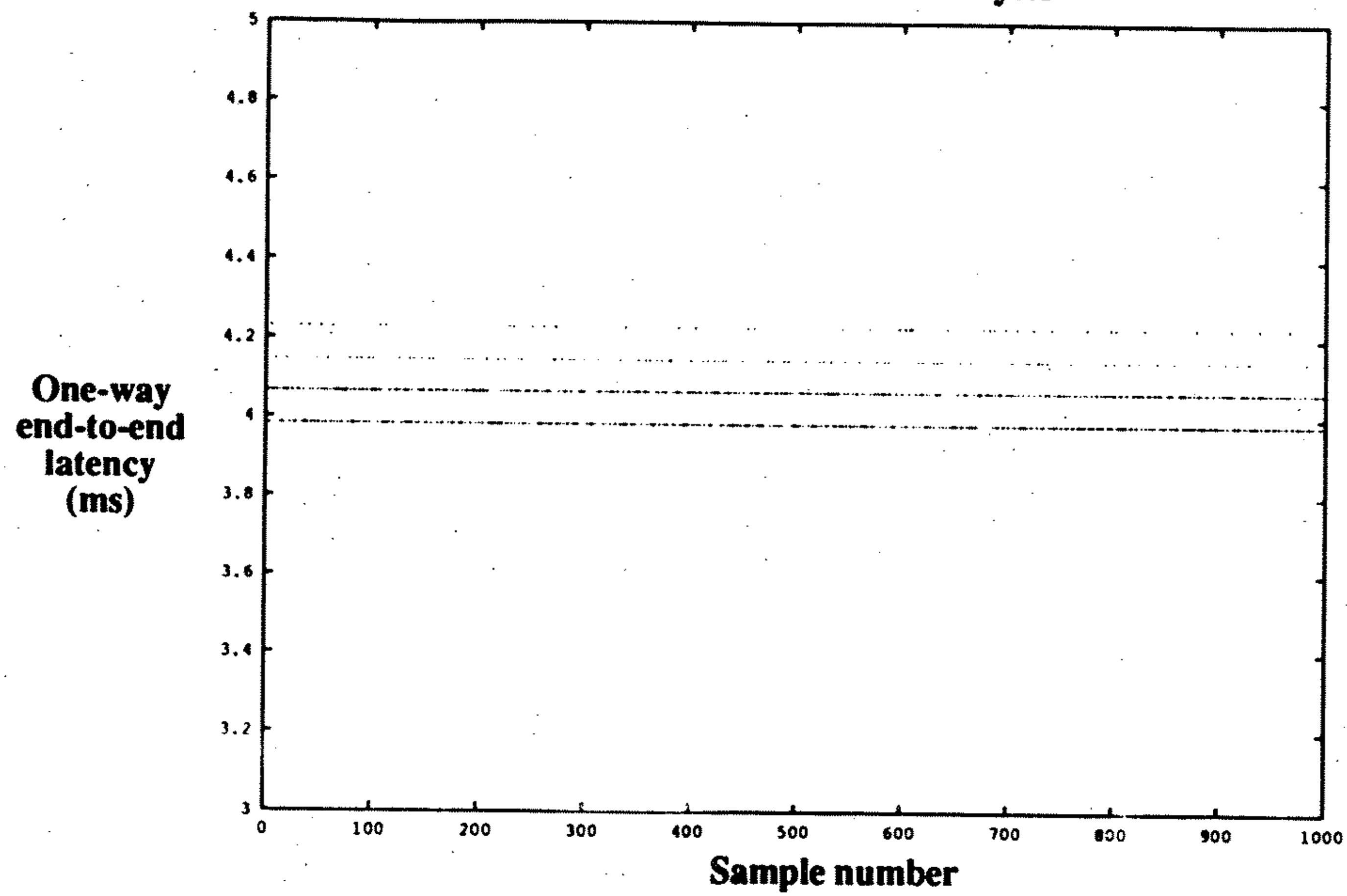
1000 samples
No asynchronous processor load
No background FDDI load
Voice data in FDDI synchronous class

Average latency: 4.015 ms
99.9% threshold: 4.146 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes

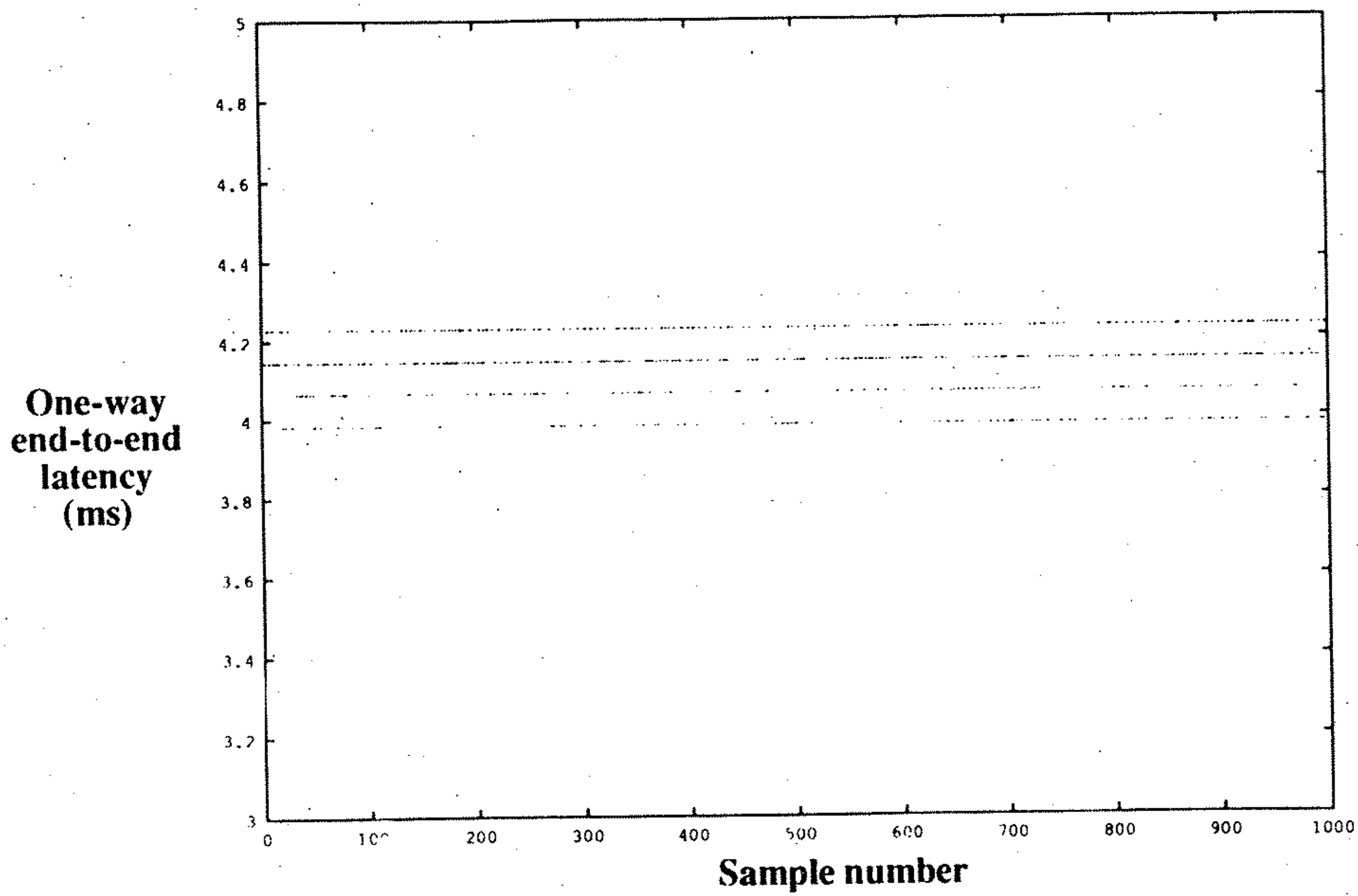


1000 samples	Average latency: 4.042 ms
No asynchronous processor load	99.9% threshold: 4.228 ms
25 Mbits/sec background synchronous FDDI load (single packets/token)	
Voice data in FDDI synchronous class	

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



1000 samples

No asynchronous processor load

50 Mbits/sec background synchronous FDDI load (single packets/token)

Voice data in FDDI synchronous class

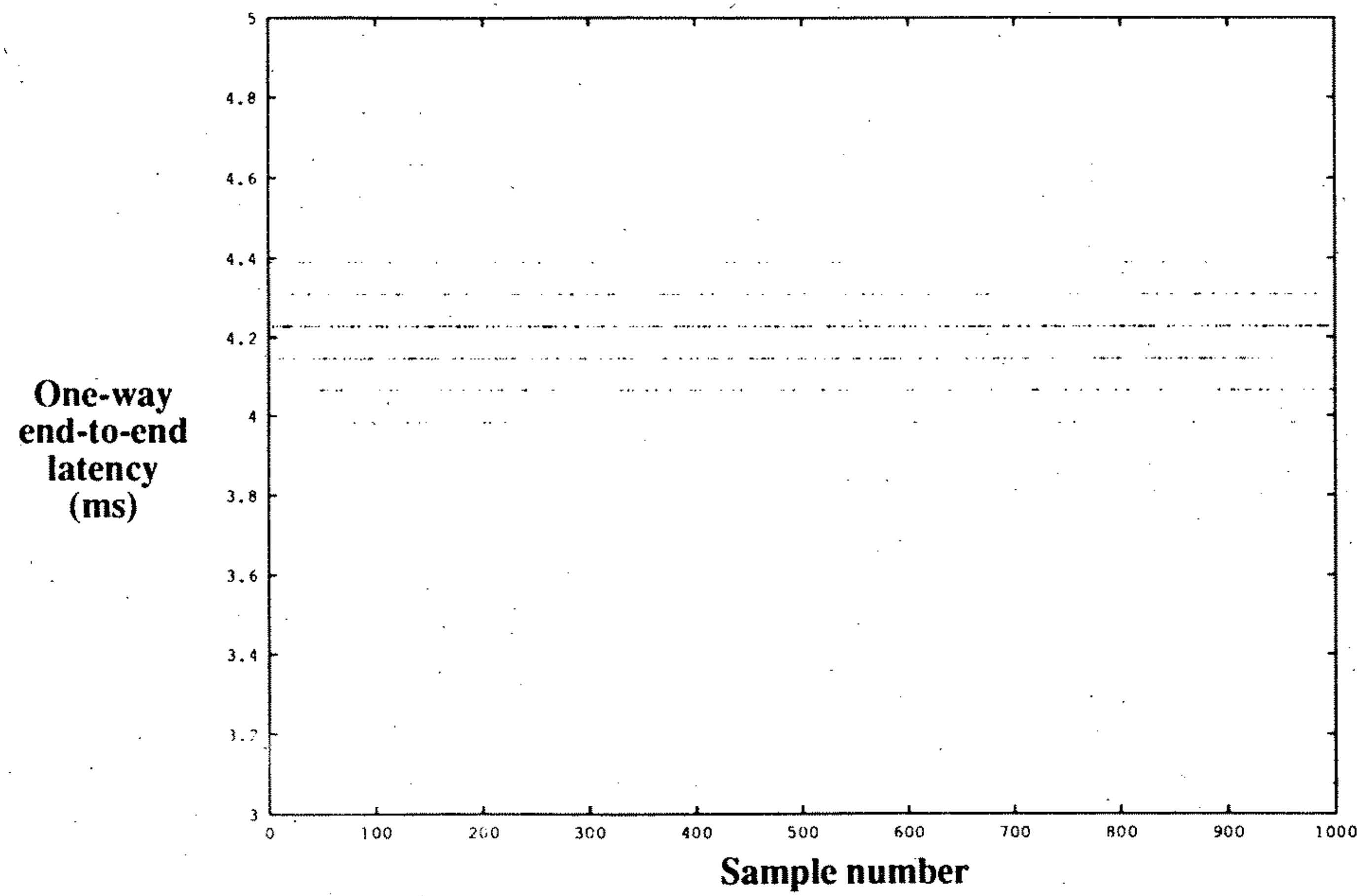
Average latency: 4.127 ms

99.9% threshold: 4.390 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



1000 samples
No asynchronous processor load
75 Mbits/sec background synchronous FDDI load (single packets/token)
Voice data in FDDI synchronous class

Average latency: 4.198 ms
99.9% threshold: 4.634 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

BACKGROUND SPPT LOAD ON FDDI

voice data size (bytes)	0 Mbits/sec (Kbits/sec)	channels	25 Mbits/sec (Kbits/sec)	channels	50 Mbits/sec (Kbits/sec)	channels	75 Mbits/sec (Kbits/sec)	channels
8	26	0	27	0	27	0	27	0
16	51	0	54	0	54	0	55	0
32	104	1	108	1	108	1	110	1
64	217	3	216	3	214	3	216	3
128	424	6	422	6	420	6	421	6
256	827	12	821	12	818	12	816	12
512	1546	24	1540	24	1530	23	1530	23
1024	2792	43	2750	42	2764	43	2735	42
2048	4732	73	4732	73	4755	74	4691	73
4096	7162	111	7080	110	7095	110	7431	116

Experiment 2: Conclusions

1. Heavy background synchronous FDDI traffic had little effect on latency in this configuration. The worst case increase in the 99.9% threshold was less than 0.5 ms, which is insignificant.

2. The reason that background load had so little effect was because (1) the ring was small (three stations), and (2) service was SPPT. Thus each station was always given a timely opportunity to transmit.

3. SPPT service is highly desirable. Since this is the normal service discipline of FDDI hardware, and since users are not normally aware that there is a choice between SPPT and MPPT service, the default situation of SPPT service is the correct choice for voice traffic in the synchronous class.

Experiment 3
Synchronous MPPT Background FDDI Load

Traffic generator changed to generate MPPT load—15 packets transmitted per physical token claimed

15 packets of 4167 bytes each creates a burst of 500,040 bits (5 ms transmission time)

Burst rate bursts/second)	Burst period (ms between bursts)	Total offered load (Mbits/sec)
50	20	25
100	10	50
150	6.66	75

Experiment 3: Analysis

Compare 1K voice data size jitter plots for SPPT vs. MPPT at 25 Mbits/sec background FDDI load

SPPT: tightly grouped between 4.0 and 4.2 ms

MPPT: two bands, one around 4 ms and the other around 5.7 ms

Upper band in MPPT data results from packets generated and queued while token is captured in traffic generator

Latency increases with MPPT service

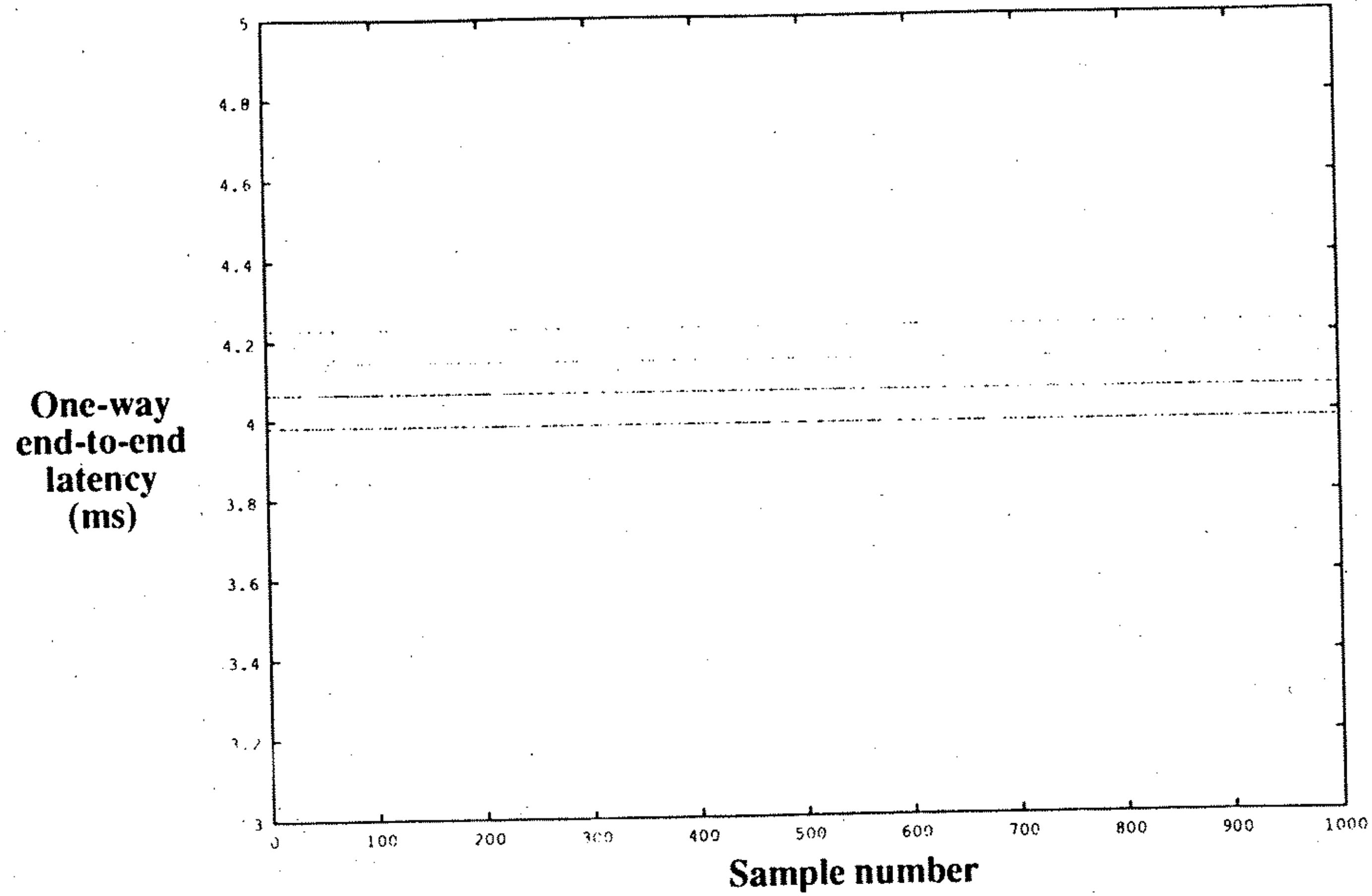
Examine average latencies (Table 8)

Examine 99.9% threshold latencies (Table 9)

Still, the worst case 99.9% threshold latency is 10 ms

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



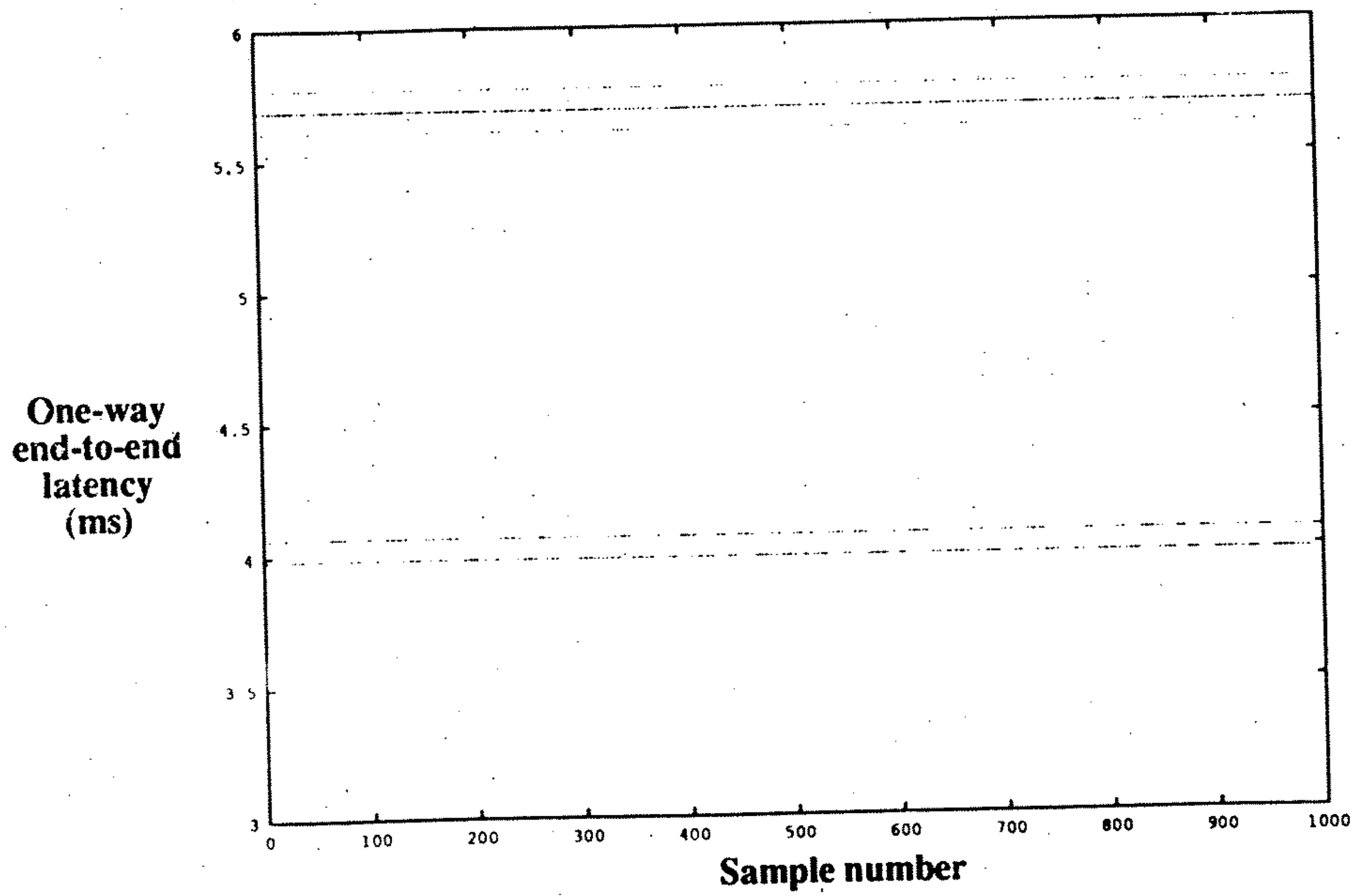
1000 samples
No asynchronous processor load
25 Mbits/sec background synchronous FDDI load (single packets/token)
Voice data in FDDI synchronous class

Average latency: 4.042 ms
99.9% threshold: 4.228 ms

SYNCHRONOUS SPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



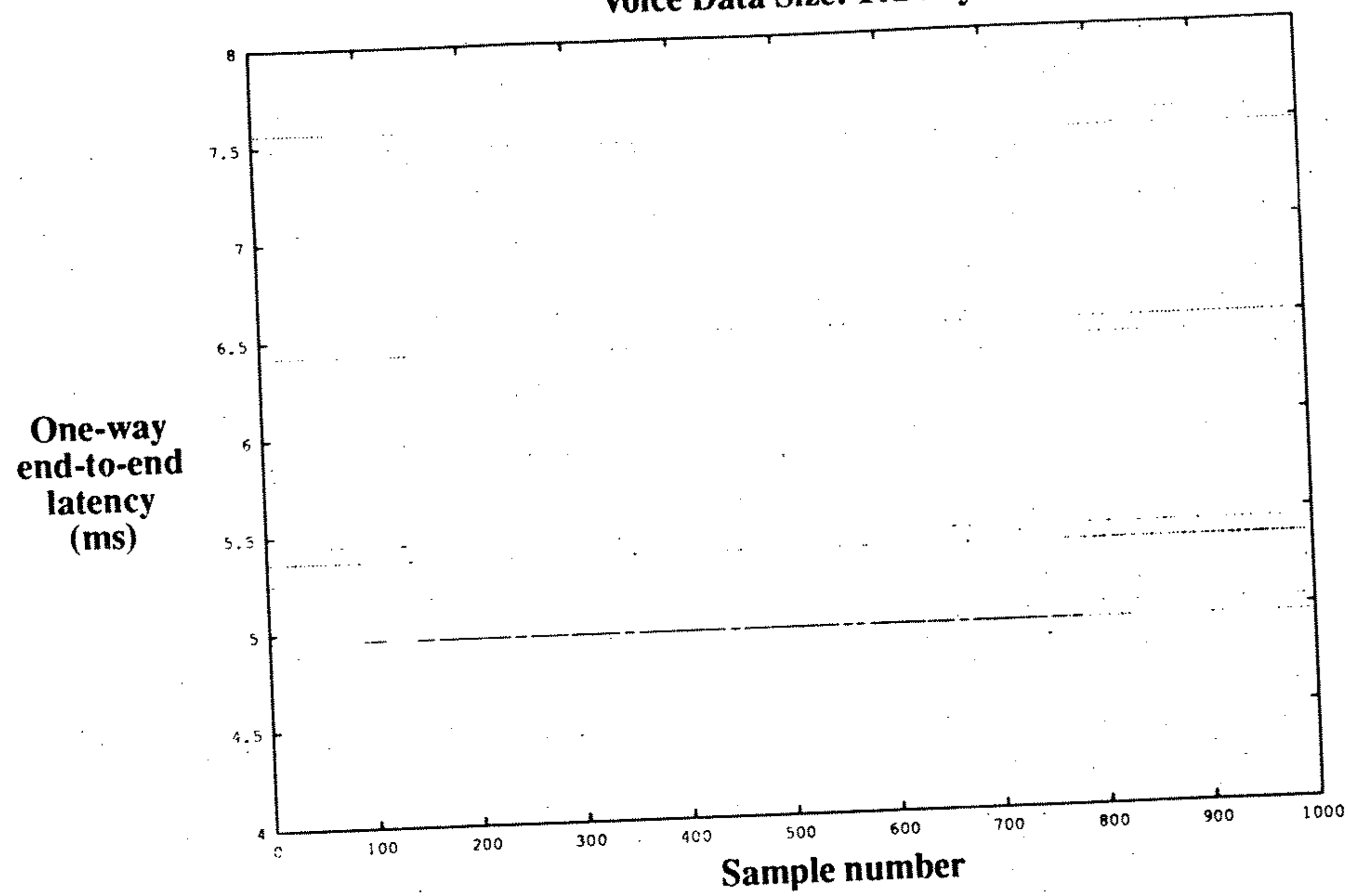
Average latency: 4.861 ms
99.9% threshold: 5.772 ms

1000 samples
No asynchronous processor load
25 Mbits/sec background synchronous FDDI load (15 packets/token)
Voice data in FDDI synchronous class

SYNCHRONOUS MPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



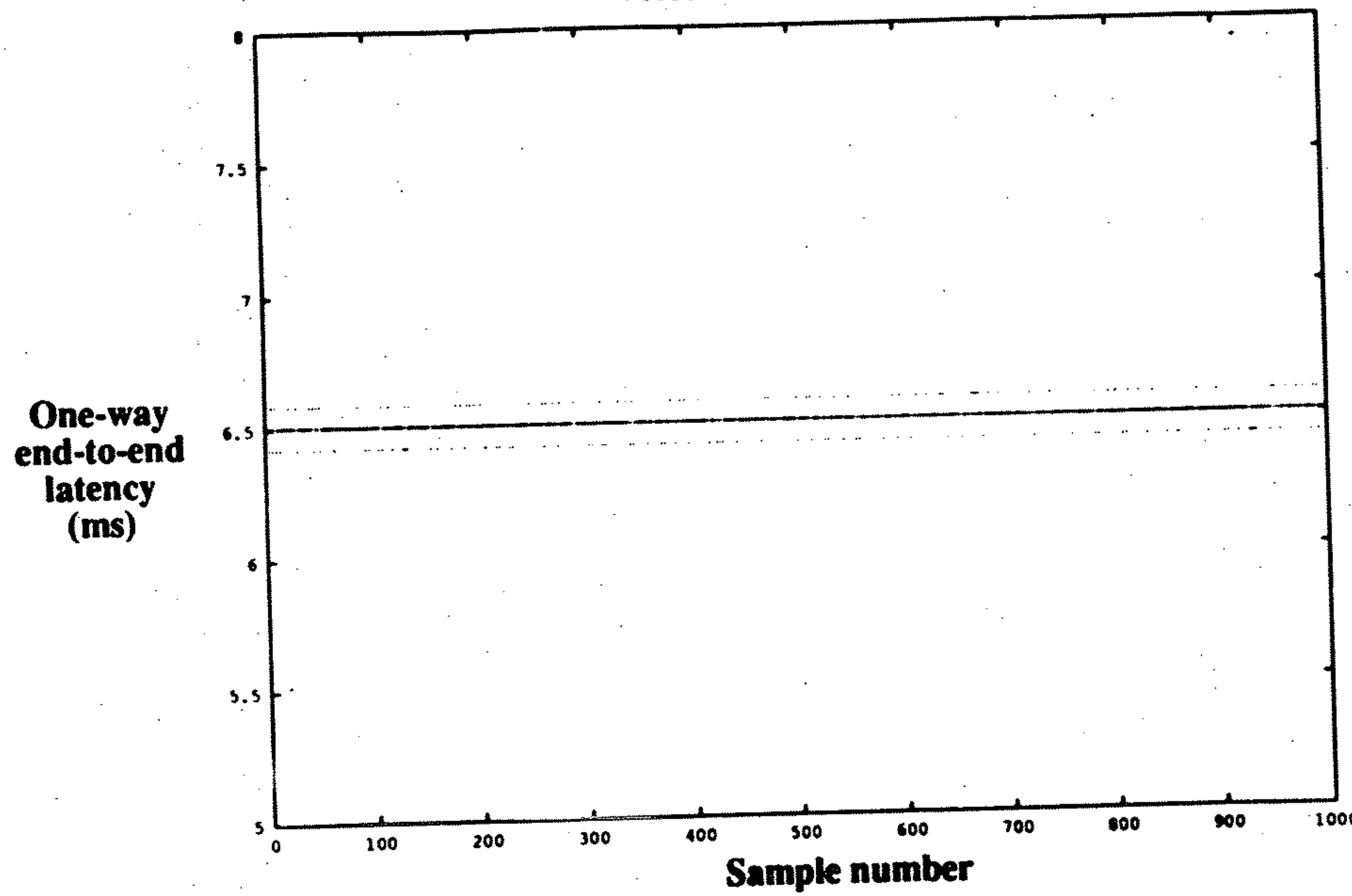
1000 samples
No asynchronous processor load
50 Mbits/sec background synchronous FDDI load (15 packets/token)
Voice data in FDDI synchronous class

Average latency: 5.397 ms
99.9% threshold: 7.967 ms

SYNCHRONOUS MPPT BACKGROUND FDDI LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



1000 samples	Average latency:	6.502 ms
No asynchronous processor load	99.9% threshold:	6.585 ms
75 Mbits/sec background synchronous FDDI load (15 packets/token)		
Voice data in FDDI synchronous class		

SYNCHRONOUS MPPT BACKGROUND FDDI LOAD

SYNCHRONOUS MPPT BACKGROUND LOAD

voice data size (bytes)		average latency (ms)	
	25 Mbits/sec	50 Mbits/sec	75 Mbits/sec
8	3.526	4.958	6.501
16	3.522	4.959	6.501
32	3.525	4.958	6.504
64	3.522	4.956	6.501
128	3.816	4.957	6.503
256	3.884	4.957	6.504
512	4.319	4.958	6.505
1,024	4.861	5.397	6.502
2,048	6.498	9.932	6.503
4,096	9.765	9.940	9.919

Table 8
Average Latency with Synchronous MPPT Background FDDI Load

SYNCHRONOUS MPPT BACKGROUND LOAD

voice data size (bytes)	99.9% threshold latencies (ms)		
	25 Mbits/sec	50 Mbits/sec	75 Mbits/sec
8	5.122	4.959	6.585
16	5.122	4.959	6.585
32	5.122	5.041	6.585
64	5.041	4.959	6.585
128	5.447	4.959	6.992
256	5.203	5.041	6.585
512	5.854	5.041	6.585
1,024	5.772	7.967	6.585
2,048	7.236	10.000	6.667
4,096	9.837	10.000	9.919

Table 9
99.9% Threshold Latency with Synchronous MPPT Background FDDI Load

Experiment 3: Conclusions

1. MPPT service caused end-to-end latencies to depart from a smooth distribution and instead to fall into groups.

2. Latencies in the lower group resulted from packets generated and transmitted between bursts from the load generator; their latencies are not significantly different from those observed in the basic experiment with no background load.

3. Latencies in the higher group resulted from packets generated while a burst was in progress; their latency was affected by the duration of the burst.

4. The increase in latency due to MPPT service was bounded here by the fact that there was only one load generator; had their been more than one then the effect would have been more dramatic.

5. MPPT service marginally increases the efficiency of a station sending non-voice traffic in its synchronous class while increasing the variance of all the voice traffic. Thus MPPT service is not recommended.

Retransmission

Modern fiber optic LANs rarely lose data, but when they do it is the responsibility of the transport protocol to recover

TP4 and TCP use *go-back-n*; XTP uses *selective retransmission*

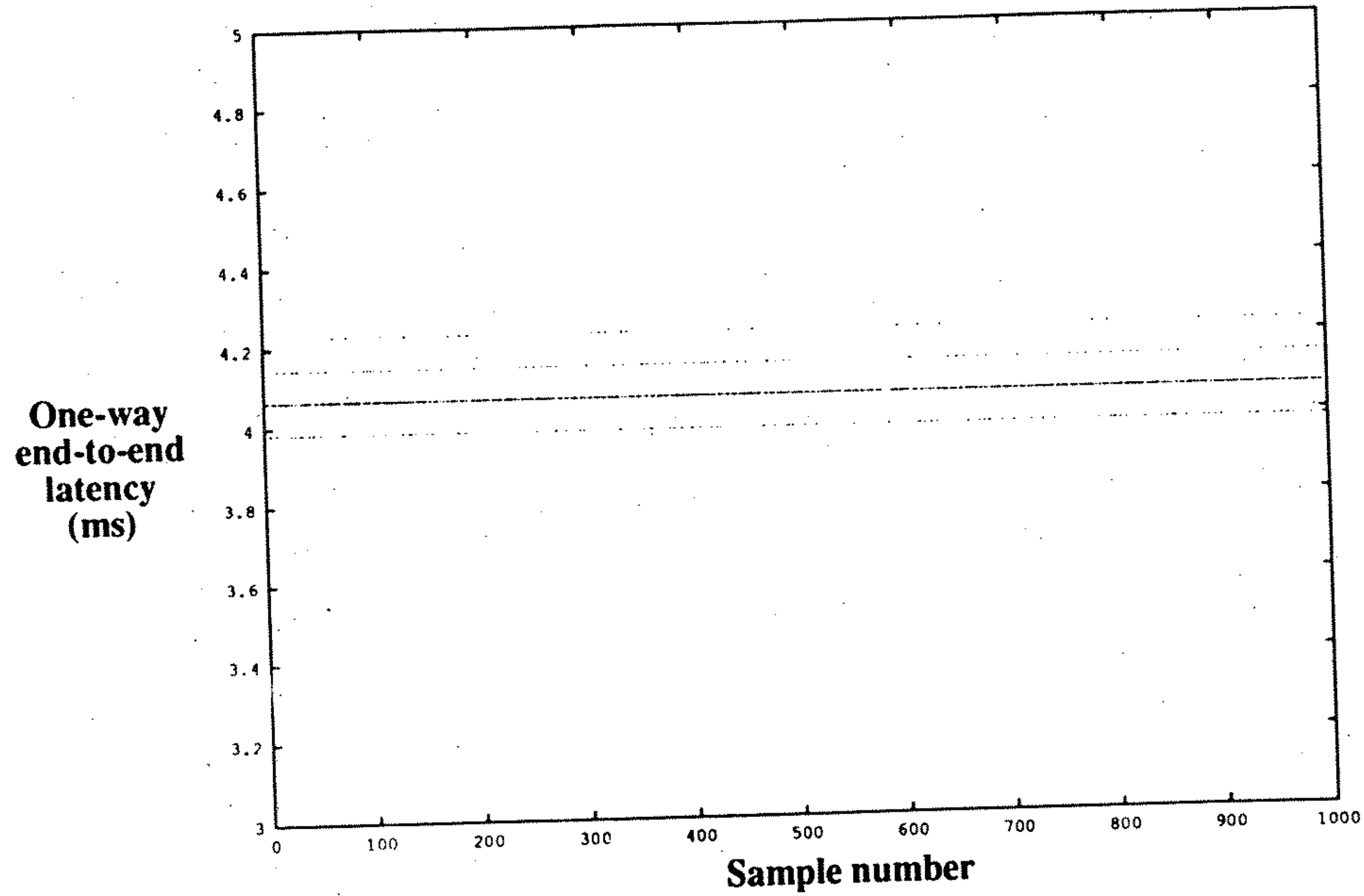
XTP acknowledges *spans*; from that the transmitter retransmits *gaps*

We modified XTP to randomly fail to acknowledge 1%, 5%, and 10% of all data packets received

XTP error repair is very efficient; average latency increased less that 0.1 ms between 0% and 10% loss rates

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



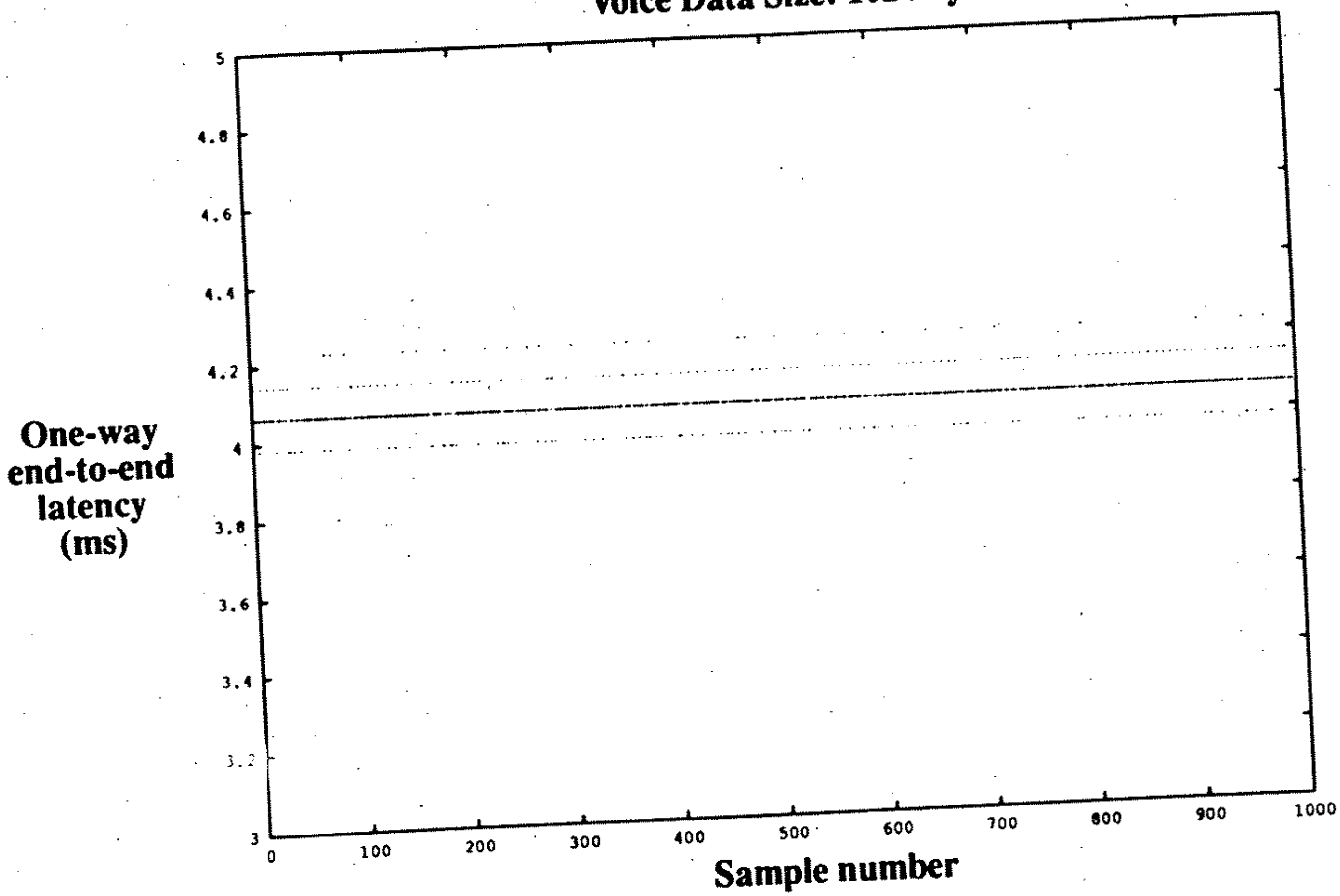
Average latency:	4.068 ms
99.9% threshold:	4.228 ms

1000 samples
No asynchronous processor load
25 Mbits/sec background synchronous FDDI load (single packets/token)
1% packet loss
Voice data in FDDI synchronous class

RETRANSMISSIONS

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



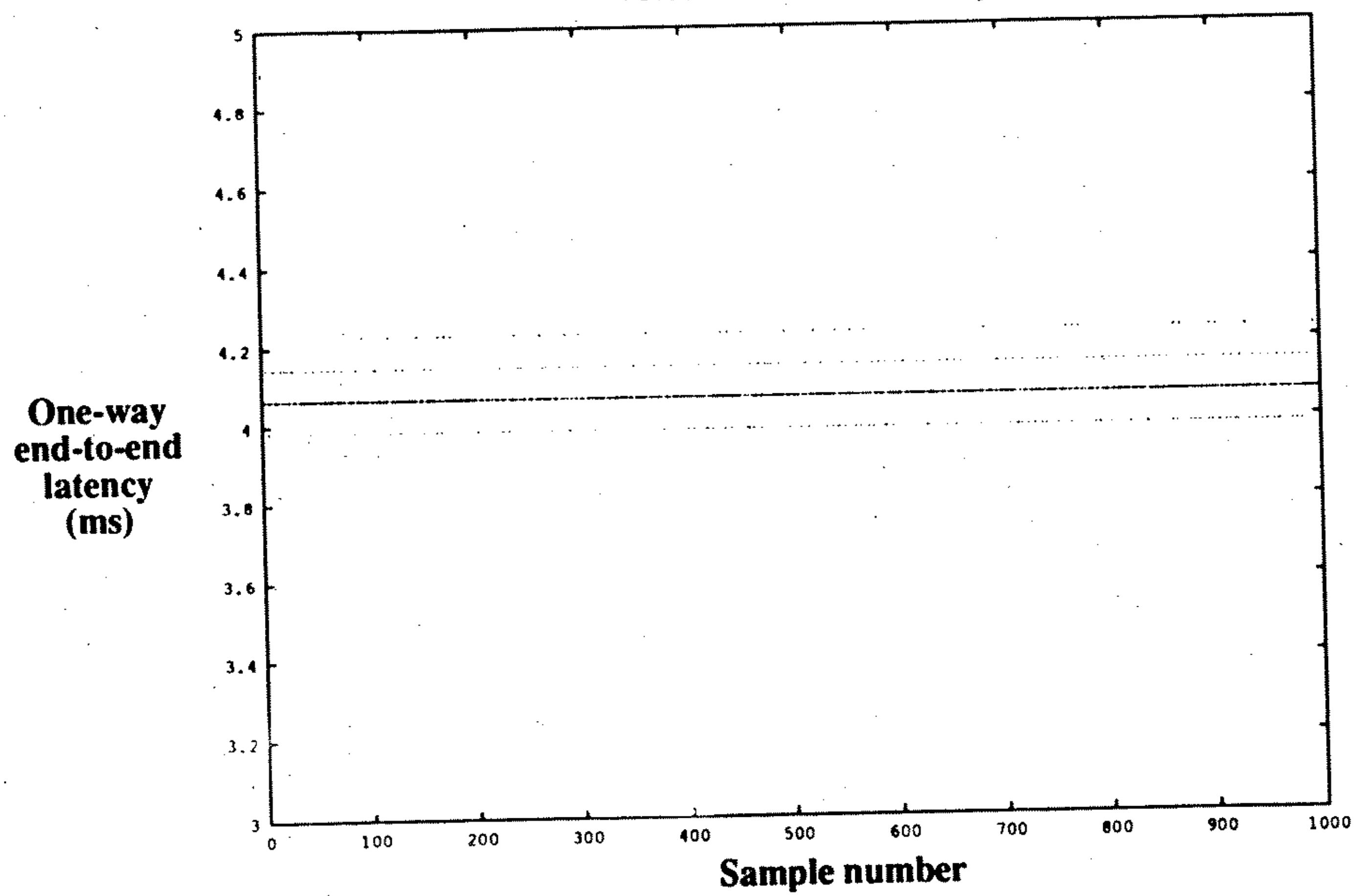
Average latency: 4.070 ms
99.9% threshold: 4.228 ms

1000 samples
No asynchronous processor load
25 Mbits/sec background synchronous FDDI load (single packets/token)
5% packet loss
Voice data in FDDI synchronous class

RETRANSMISSIONS

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



1000 samples
No asynchronous processor load
25 Mbits/sec background synchronous FDDI load (single packets/token)
10% packet loss
Voice data in FDDI synchronous class

Average latency: 4.071 ms
99.9% threshold: 4.228 ms

RETRANSMISSIONS

RETRANSMISSIONS

voice data size (bytes)	average latency (ms)			
	0% loss	1% loss	5% loss	10% loss
8	2.728	2.838	2.837	2.838
16	2.732	2.838	2.837	2.838
32	2.751	2.940	2.838	2.838
64	2.791	2.980	2.979	2.985
128	2.890	2.989	2.988	2.988
256	3.062	3.122	3.117	3.120
512	3.417	3.492	3.487	3.491
1,024	4.015	4.068	4.070	4.071
2,048	5.242	5.316	5.317	5.313
4,096	7.699	7.779	7.776	7.776

Table 10
Average Latency with 1%, 5%, and 10% Packet Loss

Experiment 4: Conclusions

1. Modern fiber optic networks such as FDDI have a very low packet loss rate.

2. Even so, when errors do occur, repair was swift and efficient. When comparing a loss rate of 0% to a loss rate of 10% in this architecture, the worst case increase in average end-to-end delay was less than 0.1 ms.

3. XTP, using selective retransmission, provided a very effective mechanism for end-to-end reliability.

Experiment 5:
Background Asynchronous Processor Load

E-Systems was concerned with background asynchronous communications load on the processor

"Average" case was 10 messages/sec, "worst case" was 120 messages/sec

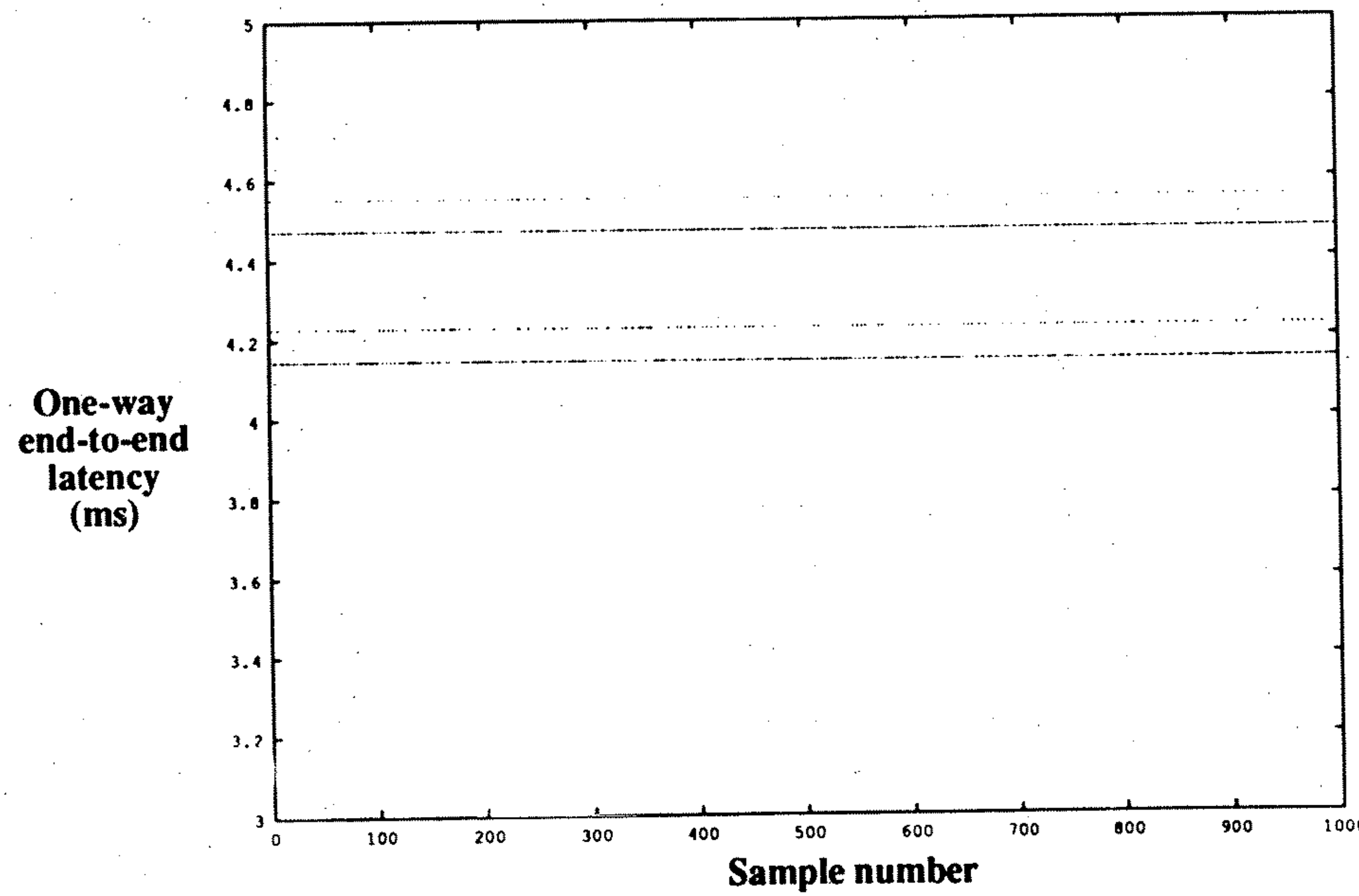
Message length (payload) distribution was defined to be half with 40 bytes/message and half with 512 bytes/message

"Average" case had negligible effect

We experimented with "worst case" background processor load and added 25 and 50 Mbits/sec of background synchronous FDDI load

JITTER MEASUREMENT

Voice Data Size: 1024 bytes

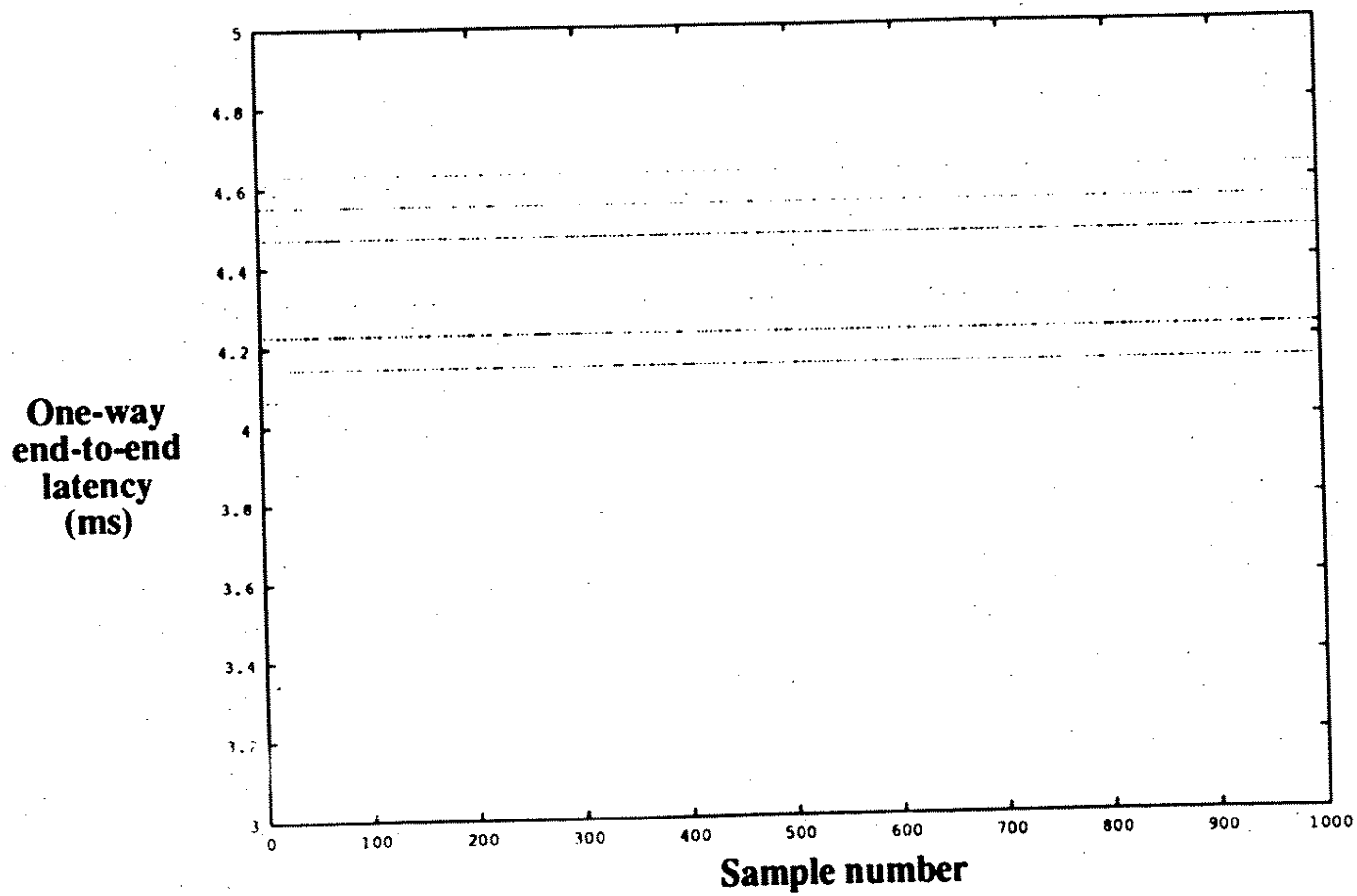


1000 samples	Average latency: 4.324 ms
120 msg/sec asynchronous processor load	99.9% threshold: 4.553 ms
No background FDDI load	
Voice data in FDDI synchronous class	

ASYNCHRONOUS PROCESSOR LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



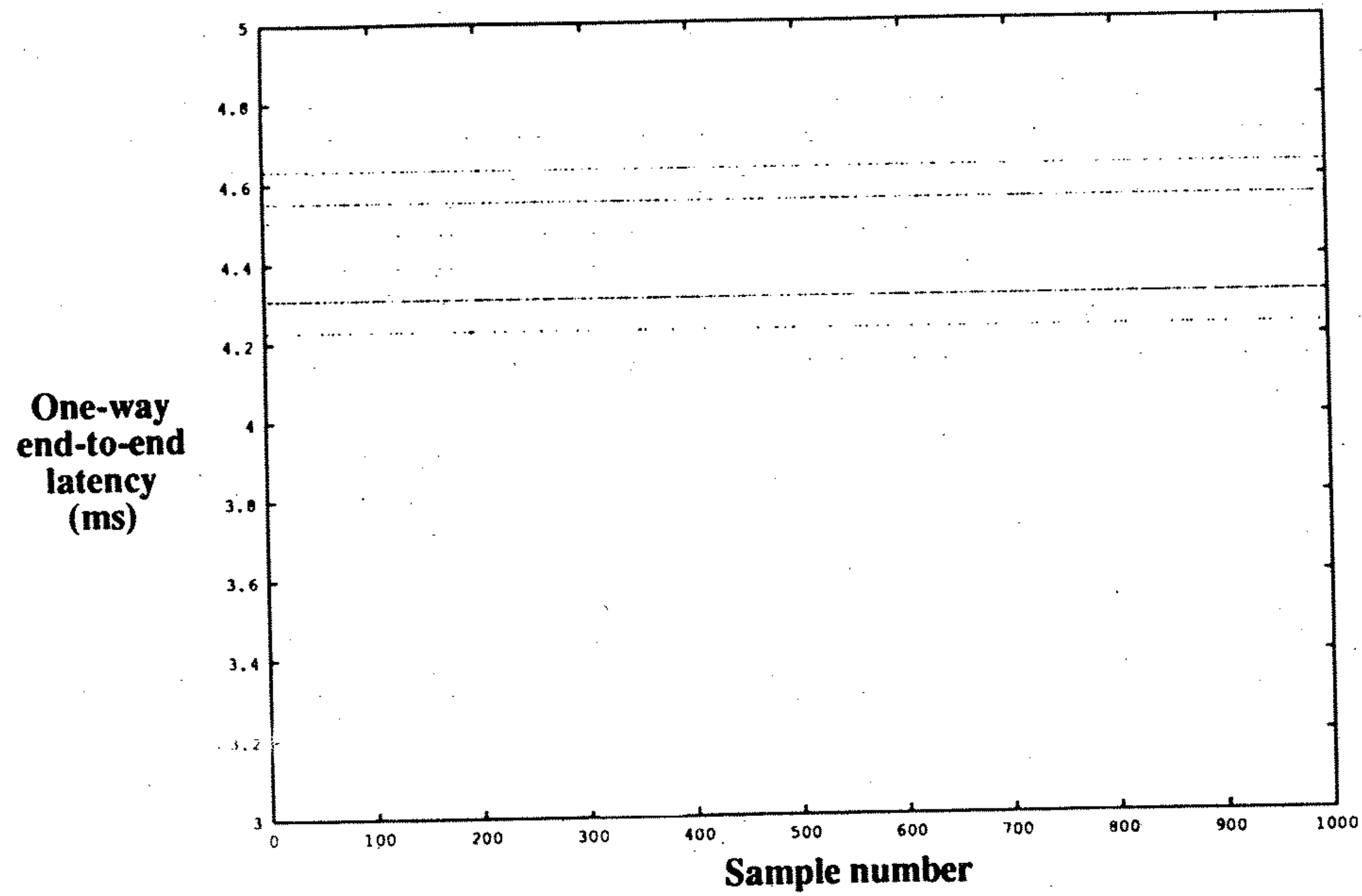
Average latency:	4.350 ms
99.9% threshold:	4.634 ms

1000 samples
120 msg/sec asynchronous processor load
25 Mbits/sec background synchronous FDDI load (single packets/token)
Voice data in FDDI synchronous class

ASYNCHRONOUS PROCESSOR LOAD

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



1000 samples
120 msg/sec asynchronous processor load
50 Mbits/sec background synchronous FDDI load (single packets/token)
Voice data in FDDI synchronous class

Average latency: 4.443 ms
99.9% threshold: 4.797 ms

ASYNCHRONOUS PROCESSOR LOAD

ASYNCHRONOUS PROCESSOR LOAD
120 messages/sec
50% of length 40 bytes, 50% of length 512 bytes

voice data size (bytes)	average latency (ms)		
	0 Mbits/sec background load	25 Mbits/sec background load	50 Mbits/sec background load
8	3.689	3.740	3.775
16	3.688	3.740	3.776
32	3.693	3.743	3.778
64	3.709	3.743	3.776
128	3.745	3.781	3.859
256	3.850	3.890	3.939
512	4.024	4.045	4.108
1,024	4.324	4.350	4.443
2,048	5.241	5.272	5.278
4,096	7.704	7.746	7.907

Table 11
Average Latency for Asynchronous Processor Load
with Synchronous Background FDDI Load

Experiment 5: Analysis

Asynchronous processor load had little effect

Comparing processor load vs. no processor load (with no background FDDI load), average latency increased by at most 1 ms for small packets; almost no increase for large packets

Comparing basic experiment vs. 120 messages/sec plus 50 Mbits/sec FDDI load, average latency increased by 1.1 ms for small packets and 0.2 ms for large packets

Reason: total load imposed by 120 messages/sec was 0.25 Mbits/sec, which is negligible compared to 7+ Mbits/sec of voice traffic (4096 bytes voice data size) and 50 Mbits/sec background FDDI load

Experiment 5: Conclusions

1. The addition of a "worst case" asynchronous processing load (defined to be 120 messages/sec, half of length 40 bytes and half of length 512 bytes) did not significantly increase the latency of the voice traffic.

2. The voice traffic in FDDI's synchronous class was effectively insulated from additional communications load in the asynchronous class.

Experiment 6:
Multicast

Reliable *multicast* is unique to XTP

Multicast is an effective technique for *1-to-many* delivery of identical data with a single transmission

Note that this is *transport layer*, not datalink layer, so it is end-to-end reliable

We ran multicast with two non-homogeneous receivers; second receiver was 25 MHz 386

Results for *n* receivers would be similar

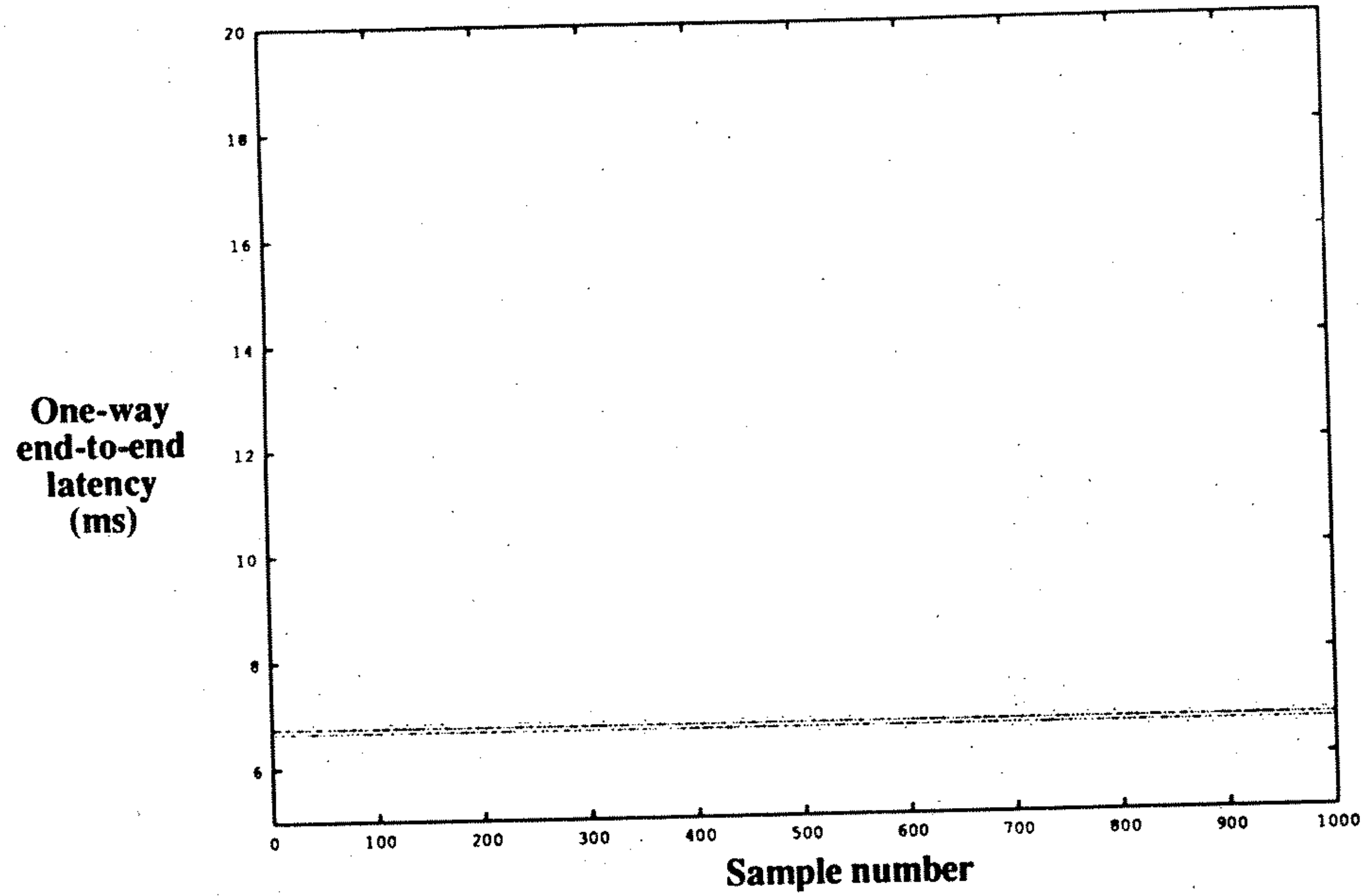
AMD SuperNet does not support MAC layer group addresses, so multicast was implemented with broadcast; National Semiconductor FDDI supports group addresses and would have been more efficient

Data transmission is reliable; error repair is *go-back-n* and is completely transparent

Group management is not a part of XTP

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



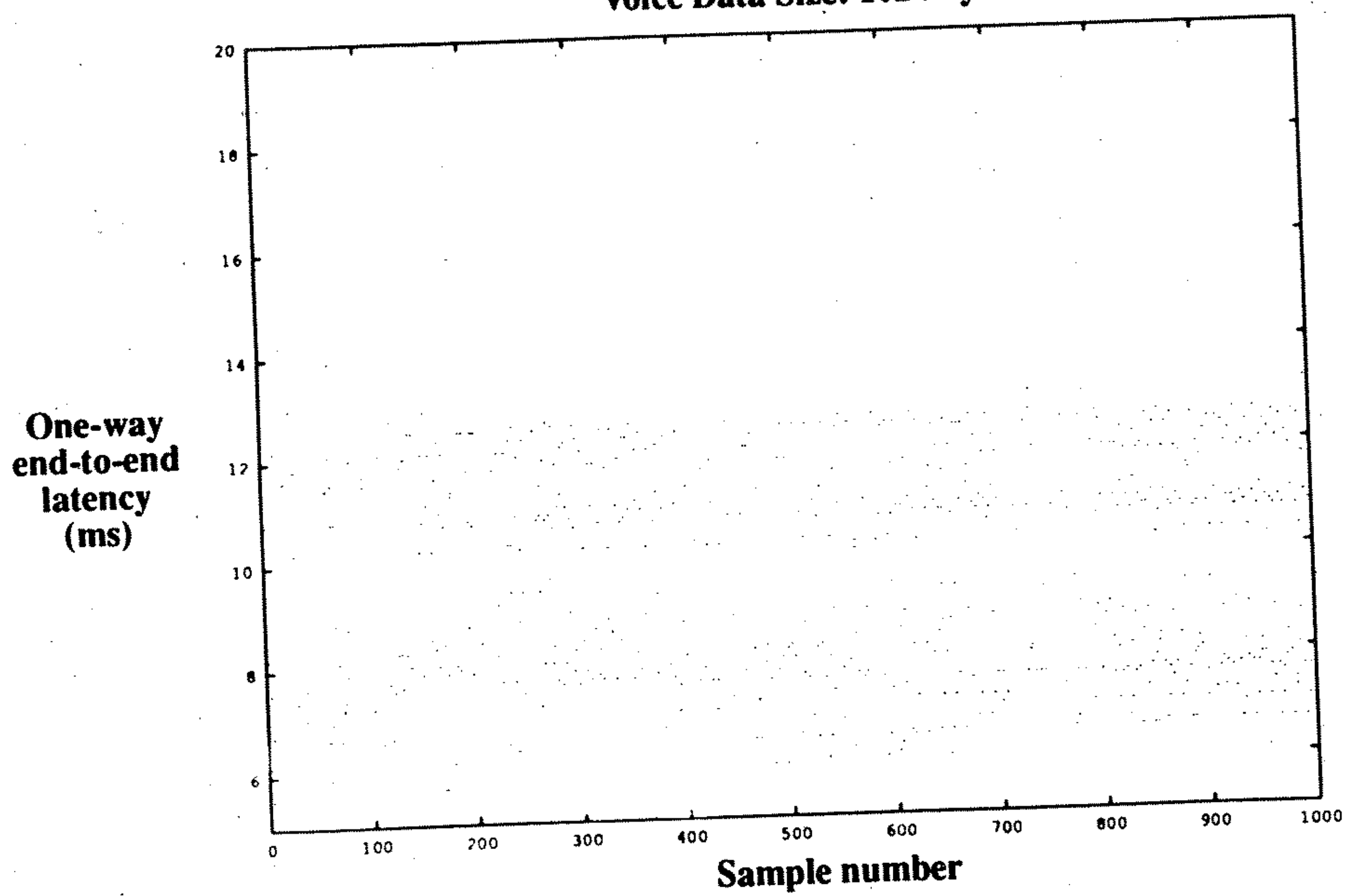
Average latency:	6.712 ms
99.9% threshold:	6.829 ms

1000 samples
No asynchronous processor load
No background FDDI load
Multicast with 2 receivers
Voice data in FDDI synchronous class

MULTICAST

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



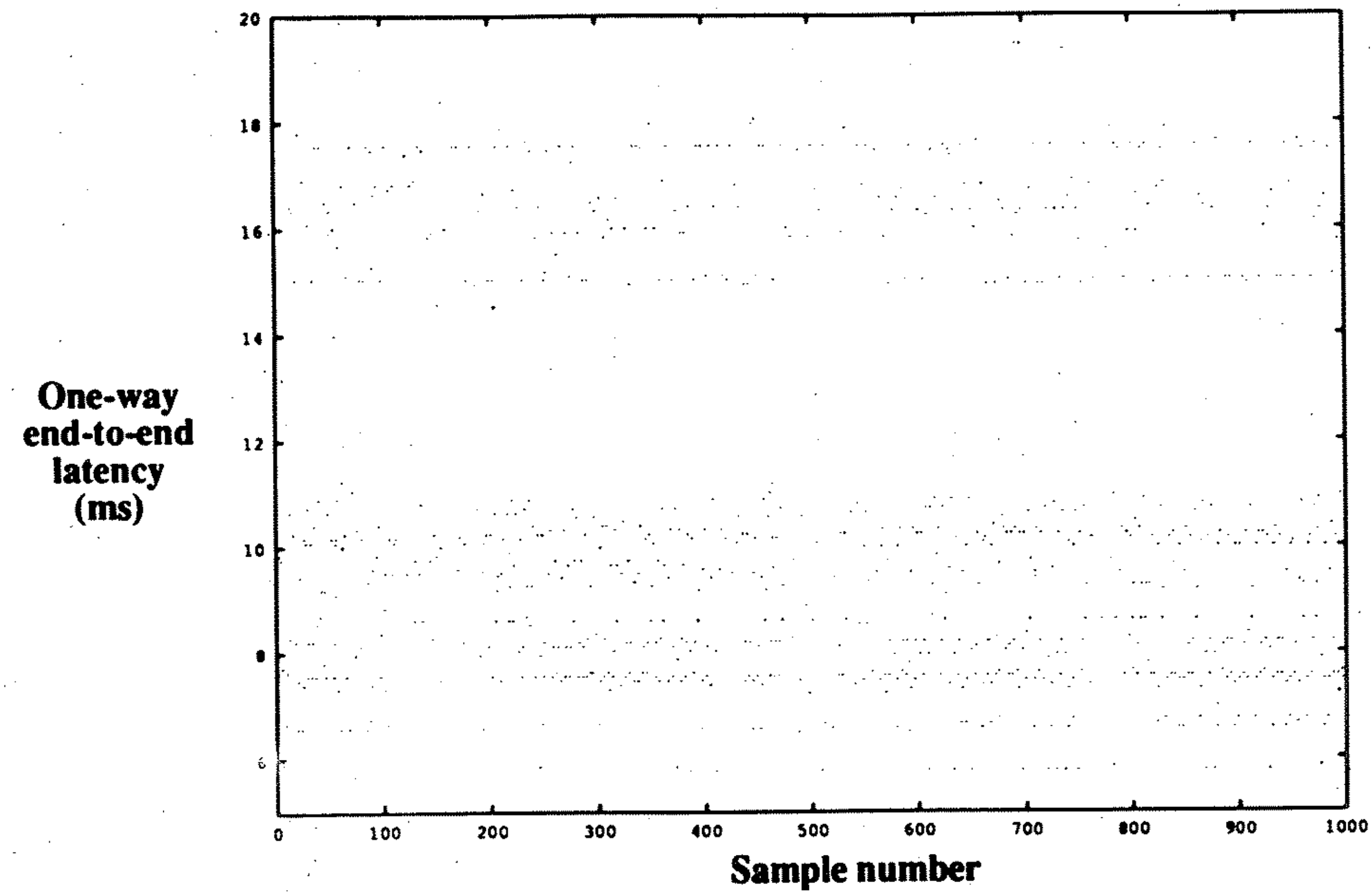
1000 samples
No asynchronous processor load
25 Mbits/sec background synchronous FDDI load (15 packets/token)
Multicast with 2 receivers
Voice data in FDDI synchronous class

Average latency: 9.515 ms
99.9% threshold: 13.171 ms

MULTICAST

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



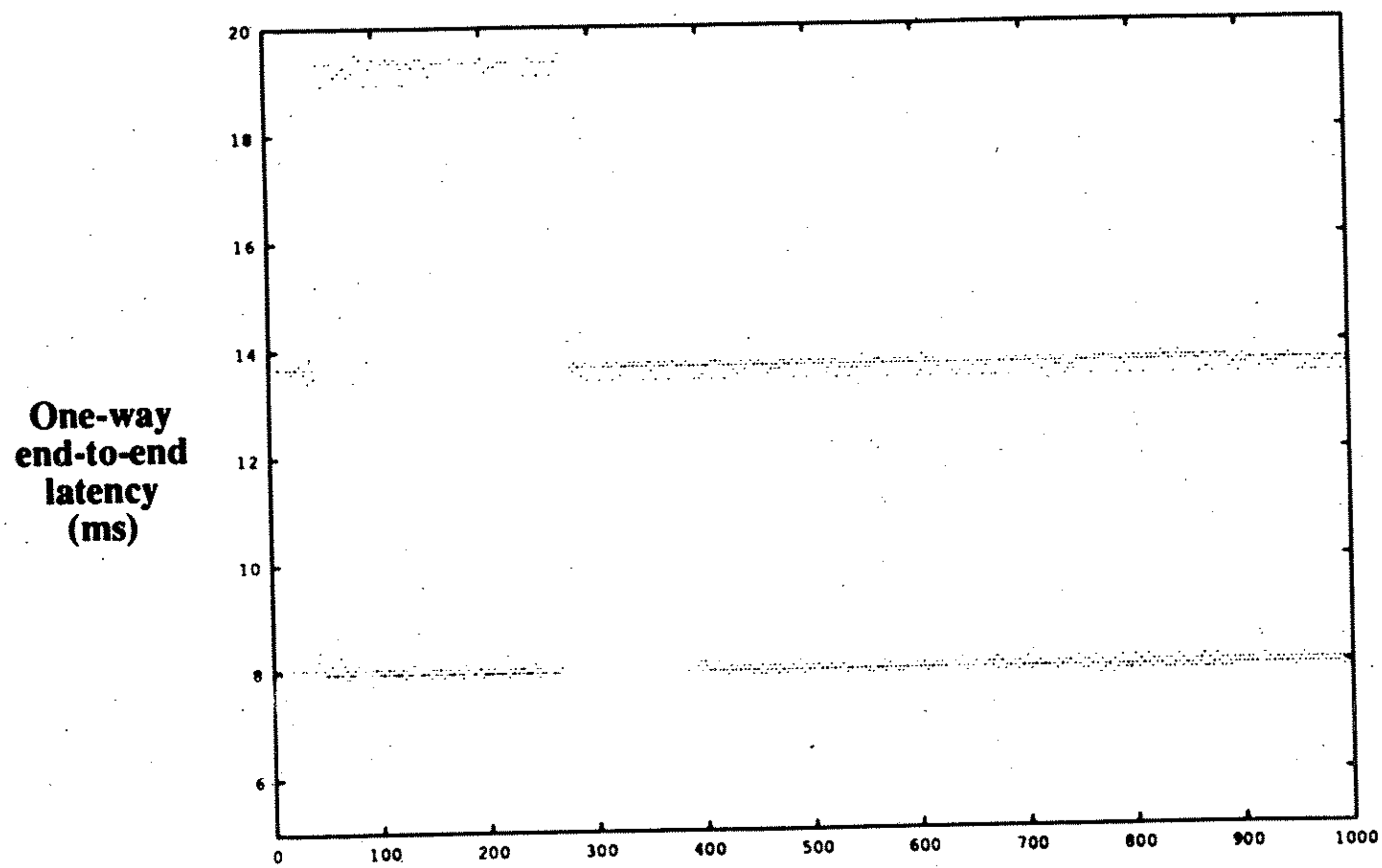
1000 samples
No asynchronous processor load
50 Mbits/sec background synchronous FDDI load (15 packets/token)
Multicast with 2 receivers
Voice data in FDDI synchronous class

Average latency:	10.460 ms
99.9% threshold:	17.967 ms

MULTICAST

JITTER MEASUREMENT

Voice Data Size: 1024 bytes



1000 samples
No asynchronous processor load
75 Mbits/sec background synchronous FDDI load (15 packets/token)
Multicast with 2 receivers
Voice data in FDDI synchronous class

Average latency: 10.804 ms
99.9% threshold: 19.512 ms

MULTICAST

MULTICAST
two receivers

voice data size (bytes)	average latency (ms)			
	0 Mbits/sec background load	25 Mbits/sec background load	50 Mbits/sec background load	75 Mbits/sec background load
8	5.266	6.311	6.209	7.559
16	5.273	6.318	6.213	7.564
32	5.277	6.293	6.240	7.573
64	5.336	6.330	6.247	7.588
128	5.431	6.375	6.290	7.608
256	5.603	6.412	6.609	7.673
512	5.894	6.461	8.136	7.770
1,024	6.712	9.515	10.460	10.804
2,048	7.997	11.174	13.761	11.608
4,096	10.402	14.134	15.247	14.575

Table 12
Average Latency for Multicast
with Synchronous Background MPPT FDDI Load

MULTICAST
two receivers

voice data size (bytes)	99.9% threshold (ms)			
	0 Mbits/sec background load	25 Mbits/sec background load	50 Mbits/sec background load	75 Mbits/sec background load
8	5.366	7.724	7.724	7.805
16	5.447	7.724	7.724	8.537
32	5.336	7.724	9.756	8.347
64	5.447	7.724	7.480	8.293
128	5.528	7.805	7.805	8.211
256	5.772	7.886	10.488	9.431
512	6.016	7.967	10.163	8.293
1,024	6.829	13.171	17.967	19.512
2,048	8.211	19.837	20.081	21.463
4,096	10.569	24.878	24.146	33.984

Table 13
99.9% Threshold for Multicast
with Synchronous Background MPPT FDDI Load

Experiment 6: Analysis

With no background load, average latency using multicast was less that two serial unicasts for all n

For n=8, average unicast latency was 2.7 ms vs. 5.2 ms for one two-receiver multicast

For n=4096, average unicast latency was 7.7 ms vs. 10.4 ms for multicast

Multicast had less total latency than serial unicast for all voice data sizes

Background MPPT load on FDDI increases latency and jitter dramatically

At 75 Mbits/sec FDDI load, average latency varied from 99.9% threshold by 1 ms for n=8, but for n=4096 the average was 14.6 ms and the 99.9% threshold was 34 ms

"Banding" of data caused by MPPT service in traffic generator

Experiment 6: Conclusions

1. For the given system configuration, a multicast to two receivers was accomplished with lower total latency than two serial unicasts for all voice data sizes.

2. Multicast is an effective technique for distributing identical data to multiple receivers. The transport multicast capability is unique to XTP.

3. Multicast would have been even more effective if it had been supported by a link level multicast (i.e., MAC group address), rather than the link layer broadcast provided by the AMD SuperNet chips.

Experiment 7: Routers

One of the significant advantages of a transport protocol is that it can be routed over an internet

We did not have access to either XTP routers (none exist) or to FDDI-to-FDDI routers, so data is from commercial Ethernet routers

Wellfleet Link Node performance

512/1024/1518 bytes: 2000/1500/1000 packets/sec

64/512/1024 bytes: 1.4/1.5/1.4 ms delay

Cisco performance

512/1024/1518 bytes: 1500/1000/700 packets/sec

64/512/1024 bytes: 0.1/0.5/1.0 ms delay

Proteon p4200 performance

512/1024/1518 bytes: 800/600/400 packets/sec

64/512/1024 bytes: 1.0/2.8/3.3 ms delay

Experiment 7: Analysis

Router data is inconclusive

XTP router does not (yet) exist

Ethernet router data does not reflect FDDI performance

Ethernet packet lengths not representative of FDDI packet lengths

Impact of packet length was variable

Only a fraction of total network traffic would go through routers

Under the assumption that FDDI routers will be equal or better in performance to the best Ethernet routers, and that performance is constant (rather than linear) with packet length, we project a delay of about 1 ms through a FDDI router—and should be less through an XTP router

Experiment 7: Conclusions

1. It is difficult to estimate the delay introduced by an XTP router, since none exist. Using existing Ethernet and Internet routers as a guide, we estimate a delay of approximately 1 ms through an XTP and FDDI router for packets of size 1024 bytes.

2. The ability to route digitized voice over an arbitrary network topology is a very significant benefit. If it can be done with an additional delay on the order of 1 ms per one kilobyte packet, it is well worth the investment.

Datalink vs. Transport Protocols

Literature survey reveals

- acceptable jitter between packets is 20 ms
- acceptable loss rate is 1-2%
- historically, packets are small (20-50 ms of voice) so that loss of any one packet has small effect
- total delay should be less than 250 ms to avoid start/stop effect

Datalink protocol over single segment FDDI should be perfectly adequate, but so should XTP if parameters are properly chosen

Datalink Protocols

If we restrict the comparison to the environment in which datalink protocols will work (e.g., single segment LANs), then

- datalink protocol probably more efficient than transport protocol

- packet loss rate perfectly adequate

- user would have to manage user-process addressing (MAC addresses just steer packets to the host)

Transport Protocols

Advantages appear when we leave the single segment network environment

Transport protocol provides guaranteed, in-order delivery

End-to-end packet loss rate is zero!

Comparison requires sub-questions:

(1) Does the system operate over an internet? Then must have bridges or routers. If routers, only a transport plus network protocol is feasible

(2) Does the system interconnect with wide area networks? If so, only a transport protocol is feasible.

(3) Would the system benefit from multicast? If so, only XTP is feasible.

(4) Can the system "afford" a transport protocol. For n=1024, need packet every 128 ms. Excluding multicast, worst case observed had latency of 10 ms. Becomes a question of how many voice channels can be supported simultaneously.

XTP vs. Other Transport Protocols

We used XTP because

- it was readily available
- we had intimate knowledge of its operation
- it alone provided a multicast capability

XTP has all the features of TCP and TP4, plus more

XTP somewhat more efficient than Wollongong TCP and more efficient than Intel TP4 or Motorola TP4

It is not a question of functionality: all will work

It is a question of performance: XTP will support more simultaneous channels than the other two

Many optimizations still possible

XTP in hardware (the *Protocol Engine*) will provide transport layer services at 100+ Mbits/sec rates

TCP and TP4 are standards; XTP standardization just beginning

CONCLUSIONS

Basic Experiment

1. System efficiency increases with increasing voice data size (*n* 8-bit voice samples). The side effect which prohibits arbitrarily large voice data size is the connection startup time for a channel, which is *n** 125 microseconds plus end-to-end latency.

2. In the basic configuration, the number of voice channels which could be supported simultaneously is a function of voice data size as follows:

voice data size (bytes)	voice data throughput Kbits/sec	equivalent voice channels
128	468	7
256	897	14
512	1661	25
1024	3050	47
2048	5115	79
4096	7294	113

3. For a voice data size of *n*, data must be delivered within *n** 125 microseconds to avoid FIFO underflow in the receiver. For all *n*>=128, the average delivery latency and the 99.9% threshold latency were well below the required delivery latency.

voice data size (bytes)	required latency (ms)	average latency (ms)	99.9% threshold (ms)
128	16	2.890	3.008
256	32	3.062	3.171
512	64	3.417	3.496
1,024	128	4.015	4.146
2,048	256	5.242	5.366
4,096	512	7.699	7.805

4. In the basic configuration, jitter was extremely low. The worst deviation of measured latency from average latency was 0.2 ms.

5. The basic configuration was a "best case" scenario for the given architecture, so the impact of additional background FDDI load, additional processor load, packet loss and retransmission, etc., had to be confirmed by experimentation.

Synchronous SPPT Background FDDI Load

6. In this configuration the impact of a heavy (75 Mbits/sec) background load in FDDI's synchronous class was insignificant; in the worst case, the 99.9% threshold latency increased by less than 0.5 ms. The reason that the effect was so small was that the ring was small (three stations) and the background load was generated using a single-packet-per-token (SPPT) service discipline.

7. SPPT service is highly desirable; it provides frequent opportunities for the voice stations to transmit.

Synchronous MPPT Background FDDI Load

8. A multiple-packet-per-token (MPPT) service discipline is permitted by FDDI, and its use increases network efficiency by reducing per-packet overhead. However, its use is injurious to voice traffic, since synchronous voice can then be delayed by non-voice synchronous data.

9. The delays which might be encountered when using MPPT service are bounded only by the correct operation of FDDI's Station Management (SMT) protocol, which limits the amount of synchronous data which any one station can send on any one token cycle. If MPPT service is used, SMT should be operational.

10. MPPT service marginally increases the efficiency of a station sending non-voice synchronous traffic, while increasing the variance of all voice traffic. Thus MPPT service is not recommended; use SPPT instead.

Packet Loss and Retransmission

11. Modern fiber optic networks such as FDDI have a very low packet loss rate. Even so, when errors do occur, XTP repairs them swiftly and efficiently. When comparing a packet loss rate of 0% to a loss rate of 10% in this configuration, the worst case increase in average end-to-end delay was less that 0.1 ms.

12. XTP's selective retransmission algorithm is very effective for error repair.

13. One advantage of using a transport protocol is that the residual end-to-end packet loss rate is zero; that is, every packet transmitted will be received.

Asynchronous Processor Load

14. The addition of an asynchronous processing load (defined to be 120 messages/sec, half of length 40 bytes and half of length 512 bytes) did not significantly increase the latency of voice traffic.

15. Voice traffic in FDDI's synchronous class was effectively insulated from any additional communications load in the asynchronous class.

Multicast

16. Multicast is an effective technique for delivering identical data to multiple receivers with a single transmission. For the given system configuration, and for all voice data sizes, a multicast to two receivers was accomplished with lower overall latency than two serial unicasts.

17. Multicast would have been even more effective if it had the support of MAC group addresses (as with the National Semiconductor FDDI), rather than the more limited link layer broadcast supported by the AMD SuperNet FDDI.

18. XTP is the only protocol which supports a transport layer reliable multicast; there is no such concept in TCP or TP4.

Routers

19. It is difficult to predict the performance of an XTP router over FDDI, since none exist. However, judging from the performance of Internet routers over Ethernet, it seems reasonable to assume a router delay of less than 1 ms for 1024 byte packets.

20. If an XTP router can be built which yields sub-millisecond delays, then voice distribution over multiple segment networks becomes truly feasible.

Datalink vs. Transport Protocols

21. A datalink protocol over FDDI and XTP over FDDI both satisfy the requirements for voice distribution over a single segment network. The datalink protocol could be reasonably expected to be somewhat more efficient (since it is less powerful), but the transport protocol would probably be easier to use since it provides a variety of services to the user.

22. If a network is multi-segment, then it must be connected via bridges or routers. If routers are chosen, then a transport plus network protocol is the natural choice.

XTP vs. Other Transport Protocols

23. No formal comparison was made between XTP and either TCP or TP4. Experience with all three suggests that XTP would have slightly higher throughput than TCP and significantly better throughput than TP4 in this environment. XTP could therefore be expected to carry more voice channels simultaneously than the others.

24. Performance will improve markedly when XTP is available in hardware as the *Protocol Engine*. The goal of the Protocol Engine is to provide transport services at the 100+ Mbits/sec rate of FDDI or similar media.

25. TCP is a military standard and TP4 is an international standard. Standardization efforts for XTP have just recently begun through ANSI. It is impossible to predict whether and when XTP might become a recognized ANSI or ISO standard.

BACKGROUND SPPT LOAD ON FDDI

voice data size (bytes)	0 Mbits/sec (Kbits/sec)	channels	25 Mbits/sec (Kbits/sec)	channels	50 Mbits/sec (Kbits/sec)	channels	75 Mbits/sec (Kbits/sec)	channels
8	26	0	27	0	27	0	27	0
16	51	0	54	0	54	0	55	0
32	104	1	108	1	108	1	110	1
64	217	3	216	3	214	3	216	3
128	424	6	422	6	420	6	421	6
256	827	12	821	12	818	12	816	12
512	1546	24	1540	24	1530	23	1530	23
1024	2792	43	2750	42	2764	43	2735	42
2048	4732	73	4732	73	4755	74	4691	73
4096	7162	111	7080	110	7095	110	7431	116