

MEDIA ACCESS CONTROL FOR OPTICAL CDMA NETWORKS  
THROUGH INTERFERENCE AVOIDANCE

by

Purushotham V. Kamath

---

A Dissertation Presented to the  
FACULTY OF THE GRADUATE SCHOOL  
UNIVERSITY OF SOUTHERN CALIFORNIA  
In Partial Fulfillment of the  
Requirements for the Degree  
DOCTOR OF PHILOSOPHY  
(COMPUTER ENGINEERING)

December 2006

Copyright 2006

Purushotham V. Kamath

## **Dedication**

To my parents, Venkatesh and Vatsala Kamath

## **Acknowledgments**

I have had the opportunity to work with several excellent researchers during the course of my doctoral research. I would like to take this opportunity to express my gratitude to them and acknowledge their contribution in shaping me as a researcher. My advisers, Profs. Joseph Bannister and Joseph Touch, supported and advised me through the doctoral process. Their feedback and commentary has greatly influenced my work. Prof. Alan Willner gave me the opportunity to collaborate and work with researchers at USC's Optical Communication Laboratory. Prof. P. Vijay Kumar provided significant feedback on my work and its relationship to coding theory. Prof. Ramesh Govindan provided valuable feedback on the presentation of the work.

This work would not have been possible without the help of several collaborators. In particular, Poorya Saghari, Reza Gholizadeh, Saurabh Kumar, Deniz Gurkan and Paniz Ebhrahimi of the Optical Communications Laboratory at USC helped me significantly in my experimental work. I am grateful to them for discussions concerning the behavior of the optical physical layer. Reza Omrani of the coding theory group at USC provided me with inputs on coding theory, codeset design and tools for generating codesets. Sumit Mediratta provided me feedback on VLSI design and hardware implementation issues. I would also like to thank several graduate students at ISI for time spent in interesting discussions on networks, engineering

and many other topics. Among them were Lars Eggert, Panagiotis Galiotis, Amy Hughes, Kashyap Merchant, Venkata Pingali, Aatash Patel, Josh Train, Yu-shun Wang and Edward Yang.

Finally, I would like to thank my friends and family without whom the last few years would not have been as memorable. Krishna Chintalapudi, Aparna Ram, Rishi Sinha, Ramya Balasundaram, Divya Devaguptapu, Sangram Tidke, Alefiya Hussain, Debojyoti Dutta, Vidya Navalpakkam, Narayanan Sadagopan, the Fettucini gang, Iftikhar Burhanuddin, Shiva Kintali and Spundun Bhatt: thank you all for making the last few years such a great experience in every possible way. My parents, my sister Sadhana and her family constantly supported me in every way possible. I certainly would not have completed my doctoral studies without their support. Finally, I would like to thank my wife Roopa for her constant encouragement and belief in me.

# Table of Contents

<b>Dedication</b>	<b>ii</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xxii</b>
<b>Abstract</b>	<b>xxiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Optical CDMA . . . . .	2
1.2 The problem: Interference . . . . .	5
1.3 The solution: Interference Avoidance . . . . .	8
1.3.1 State of an optical CDMA network . . . . .	10
1.3.2 State estimation . . . . .	10
1.3.3 Transmission scheduling . . . . .	11
1.4 Optical CDMA and the future of optical access networks . . . . .	12
1.4.1 Utilization of the capacity of an optical fiber . . . . .	13
1.4.2 Role of optical CDMA in optical access networks . . . . .	14
1.5 Organization . . . . .	14
<b>2 A primer on optical CDMA</b>	<b>16</b>
2.1 Optical CDMA channel characteristics . . . . .	17
2.1.1 Intensity modulation and unipolar encoding . . . . .	17
2.1.2 Guided, low noise medium . . . . .	19
2.2 Optical CDMA codeset design . . . . .	22
2.3 Optical CDMA transmitter and receiver design . . . . .	25
2.3.1 Optical CDMA transmitters . . . . .	25
2.3.2 Optical CDMA receivers . . . . .	26
2.4 Optical CDMA errors . . . . .	29

<b>3</b>	<b>Interference and Interference Avoidance</b>	<b>31</b>
3.1	Interference . . . . .	32
3.1.1	Assumptions . . . . .	32
3.1.2	Condition for a bit error . . . . .	33
3.1.3	Condition for a packet error . . . . .	38
3.2	State of the line . . . . .	38
3.2.1	State of the line . . . . .	40
3.2.2	State observations . . . . .	41
3.2.3	State transitions . . . . .	41
3.3	Properties of an optical CDMA network . . . . .	42
3.3.1	Non mutually destructive nature of interference . . . . .	42
3.3.2	Correlation of state . . . . .	45
3.4	Comparison of state and its properties to other physical layers . . . . .	47
3.4.1	Wired networks . . . . .	48
3.4.1.1	Interference . . . . .	48
3.4.1.2	Correlation of state . . . . .	48
3.4.2	Wireless networks . . . . .	49
3.4.2.1	Interference . . . . .	50
3.4.2.2	Correlation of state . . . . .	51
3.5	Interference Avoidance . . . . .	53
3.5.1	Transmission scheduling . . . . .	54
3.5.2	State estimation . . . . .	55
<b>4</b>	<b>Network architecture</b>	<b>56</b>
4.1	Network topology . . . . .	57
4.2	Node architecture . . . . .	59
4.3	Physical layer design . . . . .	61
4.3.1	Codeword assignment . . . . .	61
4.3.2	Codeset design . . . . .	61
4.4	Link layer design . . . . .	63
4.4.1	Addressing . . . . .	63
4.4.2	Address to codeword mapping . . . . .	63
4.4.3	Frame synchronization . . . . .	64
4.4.4	Frame format . . . . .	64
4.5	Upper layer protocol design issues . . . . .	66
<b>5</b>	<b>Transmission scheduling</b>	<b>68</b>
5.1	The transmission scheduling problem . . . . .	69
5.2	Transmission scheduling strategies . . . . .	69
5.2.1	Selfish strategies . . . . .	70
5.2.2	Cooperative strategies . . . . .	71
5.2.3	Pseudo-cooperative strategies . . . . .	72
5.3	Transmission scheduling algorithms . . . . .	73
5.3.1	Pure selfish scheduling . . . . .	74

5.3.2	Threshold scheduling . . . . .	75
5.3.3	Overlap section scheduling . . . . .	76
5.4	Performance study . . . . .	77
5.4.1	Assumptions . . . . .	77
5.4.2	Performance metrics . . . . .	78
5.4.2.1	Normalized offered load . . . . .	78
5.4.2.2	Normalized network throughput . . . . .	79
5.4.3	Analysis . . . . .	80
5.4.4	Simulation . . . . .	82
5.4.5	Discussion . . . . .	84
5.5	Performance limits of transmission scheduling algorithms . . . . .	87
5.5.1	Centralized perfect scheduling . . . . .	88
5.5.2	Distributed perfect scheduling . . . . .	89
5.5.3	Discussion . . . . .	90
5.6	Sensitivity analysis . . . . .	92
5.6.1	Effect of varying the length of the codeset . . . . .	93
5.6.2	Effect of varying the weight of the codeset . . . . .	94
5.6.3	Effect of varying the cross correlation parameter . . . . .	96
5.6.4	Effect of varying the number of wavelengths . . . . .	98
5.6.5	Effect of varying the threshold scheduling algorithm parameter . . . . .	98
5.6.6	Effect of packet size distribution . . . . .	100
5.6.7	Performance with real network traffic . . . . .	106
5.7	Conclusions and future work . . . . .	107
<b>6</b>	<b>State estimation</b> . . . . .	<b>108</b>
6.1	The state estimation problem . . . . .	109
6.2	State estimation algorithms . . . . .	111
6.2.1	Distribution of the state . . . . .	112
6.2.2	Implementation . . . . .	115
6.2.2.1	Observation parameters . . . . .	115
6.2.2.2	Estimation parameters . . . . .	116
6.2.3	State estimation algorithms . . . . .	116
6.3	Causes for packet loss . . . . .	119
6.3.1	Interference because of erroneous transmission scheduling . . . . .	119
6.3.2	Interference because of collisions . . . . .	120
6.3.3	Interference because of erroneous state estimation . . . . .	121
6.4	Performance study . . . . .	122
6.4.1	Perfect state estimation, $a = 0$ . . . . .	123
6.4.2	Perfect state estimation, $a > 0$ . . . . .	125
6.4.3	Realistic state estimation, $a > 0$ . . . . .	128
6.5	Sensitivity analysis . . . . .	128
6.5.1	Effect of varying the codeset length . . . . .	130
6.5.2	Effect of varying the codeset weight . . . . .	131
6.5.3	Effect of varying the cross correlation parameter . . . . .	131

6.5.4	Effect of varying the number of wavelengths . . . . .	131
6.5.5	Effect of varying threshold . . . . .	133
6.5.6	Effect of different estimation algorithms . . . . .	134
6.5.7	Effect of varying the number of state observations collected . . . . .	135
6.5.8	Effect of varying the average distance from the coupler . . . . .	136
6.5.9	Effect of varying the packet length . . . . .	138
6.5.10	Performance with real network traffic . . . . .	138
6.5.11	Performance under very high load . . . . .	139
6.6	Conclusions and future work . . . . .	142
<b>7</b>	<b>Implementation</b>	<b>143</b>
7.1	Experimental demonstration and measurement of transmission scheduling . . . . .	143
7.1.1	Optical CDMA transmitter component . . . . .	146
7.1.1.1	Modulator . . . . .	147
7.1.1.2	Encoders . . . . .	148
7.1.2	Optical CDMA receiver . . . . .	150
7.1.3	Measurement instruments . . . . .	152
7.1.4	Limitations of the testbed . . . . .	153
7.1.5	Objective of the experiment . . . . .	154
7.1.6	Experimental procedure . . . . .	155
7.1.6.1	Preliminary checks . . . . .	155
7.1.6.2	Measurements . . . . .	156
7.1.7	Results . . . . .	157
7.2	Implementation of a network interface card . . . . .	161
7.2.1	Bus interface . . . . .	163
7.2.2	State observation collection module . . . . .	164
7.2.3	State estimation module . . . . .	166
7.2.4	Controller module . . . . .	169
7.2.5	Transmission scheduling module . . . . .	169
7.2.6	Optimizing the state observation collection and state estimation module . . . . .	172
7.2.7	Ranging and synchronization modules . . . . .	173
7.2.8	Optical CDMA transmitter . . . . .	174
7.2.9	Optical CDMA receiver . . . . .	176
7.2.10	Timing . . . . .	177
7.3	Conclusions . . . . .	179
<b>8</b>	<b>Related work</b>	<b>180</b>
8.1	Wired networks with a single channel . . . . .	181
8.1.1	Contention protocols . . . . .	181
8.1.2	Reservation protocols . . . . .	183
8.2	Wired networks with multiple orthogonal channels . . . . .	184
8.3	Wired networks with pseudo-orthogonal channels . . . . .	185
8.4	Wireless networks . . . . .	185
8.4.1	Channel load sensing . . . . .	186

8.4.2	Multi-user interference estimation . . . . .	187
8.4.3	MIMO and CSIT . . . . .	187
8.4.4	Other related work in transmission scheduling . . . . .	188
8.5	Other relevant work in optical CDMA networks . . . . .	189
8.5.1	Codeset design . . . . .	189
8.5.2	Bit error rate performance analysis . . . . .	190
8.5.3	Forward error correction . . . . .	190
<b>9</b>	<b>Conclusions and future directions</b>	<b>192</b>
9.1	Conclusions . . . . .	192
9.2	Future directions . . . . .	195
9.2.1	Analysis of Interference Avoidance . . . . .	195
9.2.1.1	Optimal transmission scheduling algorithms . . . . .	195
9.2.1.2	Delay . . . . .	196
9.2.1.3	Fairness and quality of service . . . . .	196
9.2.1.4	Error control . . . . .	196
9.2.2	Architectural issues . . . . .	197
9.2.2.1	Other areas . . . . .	197
9.2.3	Power control in optical CDMA LANs . . . . .	198
9.2.4	Optical CDMA switch design . . . . .	198
	<b>Reference List</b>	<b>199</b>
	<b>Appendix A</b>	
	State representation . . . . .	206
	<b>Appendix B</b>	
	Packet error rate and number of codewords multiplexed on the line . . . . .	210
B.1	Packet error rate . . . . .	210
B.2	Number of codewords multiplexed on the line . . . . .	211
	<b>Appendix C</b>	
	Equilibrium state probabilities . . . . .	213
C.1	Assumptions . . . . .	213
C.2	Admissible transmissions . . . . .	214
C.2.1	Aloha-CDMA . . . . .	214
C.2.2	Pure selfish scheduling . . . . .	215
C.2.3	Threshold scheduling . . . . .	215
C.2.4	Overlap section scheduling . . . . .	215
C.3	Arrival state transition probabilities . . . . .	216
C.4	Departure state transition probabilities . . . . .	217
C.5	Balance equations . . . . .	217
	<b>Appendix D</b>	
	Arrival state transition diagram . . . . .	219

**Appendix E**

Throughput of Centralized Perfect Scheduling . . . . . 221  
E.1  $N_{offered} \geq N - (w - 1)$  . . . . . 223  
E.2  $N_{offered} < N - (w - 1)$  . . . . . 224

**Appendix F**

Interference Avoidance is a generalization of Carrier Sensing and CDMA . . . . . 227

## List of Figures

1.1	A typical optical CDMA network. The network consists of $M$ nodes. The nodes are connected by transmit and receive fibers to a passive star coupler forming a shared medium LAN. . . . .	2
1.2	An example of multi-user interference showing three codewords $C0$ , $C1$ , and $C2$ . The two codewords ( $C1$ and $C2$ ) interfere with another codeword ( $C0$ ). (a) shows the line (fiber) when $C0$ is OFF (transmitting a 0 bit) and (b) shows the line when $C0$ is ON (transmitting a 1 bit). In (a), $C0$ experiences an interference error. . . . .	6
1.3	The normalized network throughput vs. normalized offered load for Aloha-CDMA. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is $(10, 3, 3)$ . The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	7
1.4	The effect of Interference Avoidance. The codewords are the same as in Figure 1.2. $C0$ has been transmitted one chip time later compared to Figure 1.2. (a) shows the line (fiber) when $C0$ is OFF and (b) shows the line when $C0$ is ON. In either case, $C0$ does not have an interference error. . . . .	9
2.1	The ON-OFF keying of codewords. The lower line shows the signal on the optical fiber when a single codeword (0100011) is being transmitted. The upper line shows the corresponding data. . . . .	18
2.2	Packet, codeword and chip overlaps. This figure depicts chip synchronous transmission. The effect of chip synchronous and asynchronous transmission is discussed in Chapter 3. . . . .	20
2.3	Three codewords of an optical CDMA codeset whose parameters are codeset length $N = 25$ , weight $w = 3$ and maximum crosscorrelation parameter $\kappa = 1$ . . . . .	23

2.4	An optical CDMA receiver. It is a hard-limiting correlation detector that consists of a hard-limiter, decoder, a photodetector, and a threshold detector. The receiver is tuned to the codeword 1110000. The power in the 1 <sup>st</sup> , 2 <sup>nd</sup> and 3 <sup>rd</sup> chip positions is summed by the decoder. The photodetector converts the signal to current and the threshold detector detects a 1 bit. . . . .	26
3.1	A bit error. The figure shows four codewords on the optical fiber. In (a) codeword <i>C0</i> is OFF. A false positive error occurs at the receiver due to interference from <i>C1</i> and <i>C2</i> or <i>C3</i> . . . . .	33
3.2	Four codewords multiplexed on an optical fiber. The figure shows the receiving codeword, interfering codewords, the error generating codeset and the error generating delays. . . . .	35
3.3	The different cases that can occur when codewords overlap. (a) shows the case when all members of the EGC are ON and the receiving codeword is OFF (causes a bit error). (b) shows the case when all members of the EGC are ON and the receiving codeword is ON (does not cause a bit error). (c) shows the case when one member of the EGC is OFF and the receiving codeword is OFF (does not cause a bit error). (d) shows the case when one member of the EGC is OFF and the receiving codeword is ON (does not cause a bit error). . . . .	36
3.4	Four overlapping packets on an optical fiber. Each packet may be divided into sections at instants where a packet has arrived or departed. . . . .	37
3.5	The state of the line at a point on a receiver fiber. The state of the line at time <i>t</i> at distance <i>d</i> is [2 2 1 2 2 3 0 1]. The second bit of codeword <i>C1</i> is OFF. To calculate the state, <i>C0</i> is assumed to be ON and the codewords are added. . . . .	39
3.6	The four possible packet transmission events that are a consequence of the non-mutually destructive nature of interference. . . . .	43
3.7	Examples of the four packet transmission events. In each example, the different transmission time of codeword <i>C0</i> has caused a different transmission event. The circled chips mark the ‘1 chips’ that do not have overlaps and the corresponding codewords are preserved. . . . .	44
3.8	The correlation between the state at the estimation point at estimation time and the merging point at merging time for an optical CDMA network as the diameter of the network increases. The network has 100 nodes distributed uniformly over the length of the network. The network uses Aloha-CDMA. Packet arrivals are Poisson with offered load 1 and packet lengths are exponentially distributed with an average length of 1000 bytes. . . . .	45

3.9	Interference Avoidance prevents occurrence of a bit error. No error is caused when the same codewords in Figure 3.1 are sent with a different set of transmission delays. The figure shows the same codewords with codeword <i>C0</i> delayed by 1 chip time. . . . .	52
4.1	A typical Fiber to the Home (FTTH) network topology. It shows the upstream link where Interference Avoidance may be used as a media access mechanism.	59
4.2	Block diagram of an Interference Avoidance Network Interface Card. . . . .	60
4.3	The frame format for Interference Avoidance frames. The synchronization sequence is used for the frame synchronization operation, the length field is used to determine the size of the payload, and the checksum field is used for error detection. . . . .	65
5.1	The transmission scheduling strategies and the events that they attempt to achieve. . . . .	73
5.2	Normalized network throughput vs. normalized offered load for different transmission scheduling algorithms based on analysis. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (10,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	82
5.3	This graph shows the normalized network throughput vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (10,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	83
5.4	The average number of codewords multiplexed at a point on a receiver fiber vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (10,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	84

5.5	Packet error rate vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is $(10, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	85
5.6	The normalized throughput vs. normalized offered load for the Centralized and Distributed Perfect Scheduling algorithms and threshold scheduling. The results are based on both simulation and analysis. The parameters are detailed in the text. The codeset is $(10, 3, 3)$ . The threshold algorithm used a threshold of 0.5. . . . .	91
5.7	The normalized network throughput at normalized offered load of 1 vs. the codeset length $N$ for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset weight is 3 and $\kappa = 3$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	94
5.8	Normalized network throughput at normalized offered load of 1 for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length is 100 and $\kappa = 3$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	96
5.9	Normalized network throughput at normalized offered load of 1 for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length $N = 42$ and weight $w = 6$ . The cross correlation parameter $\kappa$ is varied from 1 to 6. For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	97

5.10	Normalized network throughput at normalized offered load of 1 vs. the number of wavelengths for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length is 10, weight is 3 and $\kappa = 3$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	99
5.11	Normalized network throughput at normalized offered load of 1 vs. threshold parameter $\alpha$ for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (100,3,3). The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	100
5.12	Normalized network throughput vs. average packet size at a normalized offered load of 1. The results are based on simulation. The traffic model is Poisson arrivals and exponentially distributed packet sizes. The codeset is (100,3,3). The algorithm was threshold scheduling; the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	101
5.13	Normalized network throughput vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with a trimodal packet size distribution (70% 40 bytes, 20% 1500 bytes and 10% 500 bytes). The codeset is (100,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. . . . .	102
5.14	The squeeze through effect. The throughput is maximized when the fraction of short packets is 0.9. The graph shows both analytical and simulation results. The transmission scheduling algorithm is pure selfish. The packet sizes are 50 bytes and 1000 bytes. The codeset is (100,3,3). . . . .	105
5.15	Normalized network throughput vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model was based on real network traffic traces (see description). The codeset is (100,3,3) and codewords are allocated to addresses. For the threshold scheduling algorithm, the threshold parameter was set to 0.5 . . . .	107

6.1	The points on the optical fiber where state is estimated and transmissions merge. The state is estimated by a node at its estimation point at the estimation time. The node transmits a packet at the transmission time and it merges with other packets at the merging point (coupler) at the merging time. . . . .	109
6.2	Observations of the state at a point. The state estimation algorithm calculates a state estimate from the state observations. . . . .	111
6.3	The mean number of bits between a state change for different codeset lengths at an offered load of 1. The network is an Aloha-CDMA network with Poisson arrivals and exponentially distributed packet lengths. The average packet length is 1000 bytes. . . . .	113
6.4	The state estimation parameters. The figure shows 10 state observations. The collection of state observations begins at $t_s$ . Estimation is run after $n_e = 5$ observations are collected at time $t_e$ . . . . .	117
6.5	Collisions on an optical CDMA network. Transmissions from nodes A and B collide because neither can see the others' transmission. . . . .	120
6.6	Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay $a = 0$ . The results are based on simulation. The codeset length is 10. All other parameters are as specified in Table 6.2 . . . . .	124
6.7	Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay $a = 0$ . The results are based on simulation. The codeset length is 100. All other parameters are as specified in Table 6.2 . . . . .	124
6.8	Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay $a = 0$ . The results are based on simulation. The codeset length is 200. All other parameters are as specified in Table 6.2 . . . . .	124
6.9	Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 10. All other parameters are as specified in Table 6.2 . . . . .	125

6.10	Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 100. All other parameters are as specified in Table 6.2 . . . . .	125
6.11	Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 200. All other parameters are as specified in Table 6.2 . . . . .	125
6.12	Normalized network throughput vs. normalized offered load for continuous state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 10. All other parameters are as specified in Table 6.2 . . . . .	127
6.13	Normalized network throughput vs. normalized offered load for continuous state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 100. All other parameters are as specified in Table 6.2 . . . . .	127
6.14	Normalized network throughput vs. normalized offered load for continuous state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 200. All other parameters are as specified in Table 6.2 . . . . .	127
6.15	Normalized network throughput vs. codeset length $N$ for different transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is $(N, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2 . . . . .	129
6.16	Normalized network throughput vs. codeset weight $w$ for different transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is $(100, w, w)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2 . . . . .	130

6.17	Normalized network throughput at normalized offered load of 1 for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length $N = 42$ and weight $w = 6$ . The cross correlation parameter $\kappa$ is varied from 1 to 6. For the threshold scheduling algorithm, the threshold parameter was set to 0.5 . . . . .	132
6.18	Normalized network throughput vs. number of wavelength for different transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (10,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2 . . . . .	132
6.19	Normalized network throughput vs. threshold for the threshold scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset lengths are 10, 100 and 200. All other parameters are specified in Table 6.2 . . . . .	133
6.20	Normalized network throughput vs. threshold vs. offered load for the threshold scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length is 100. All other parameters are specified in Table 6.2 . . . . .	134
6.21	Normalized network throughput vs. normalized offered load for on demand state estimation, normalized propagation delay $a > 0$ (average distance from the coupler is 1000m). The codeset length is $N = 200$ . The results are based on simulation. All other parameters are specified in Table 6.2 . . . . .	135
6.22	Normalized network throughput vs. state estimation window (in bits) for the transmission scheduling algorithms and continuous and on-demand state estimation. The results are based on simulation. All other parameters are specified in Table 6.2 . . . . .	136
6.23	Normalized network throughput vs. average distance from coupler (uniform distribution) for the transmission scheduling algorithms and continuous state estimation. The results are based on simulation. The codeset length is 10. All other parameters are specified in Table 6.2 . . . . .	137

6.24	Normalized throughput vs. average distance from coupler (uniform distribution) for the transmission scheduling algorithms and continuous state estimation. The results are based on simulation. The codeset length is 100. All other parameters are specified in Table 6.2 . . . . .	137
6.25	Normalized throughput vs. average distance from coupler (uniform distribution) for the transmission scheduling algorithms and continuous state estimation. The results are based on simulation. The codeset length is 200. All other parameters are specified in Table 6.2 . . . . .	137
6.26	Normalized network throughput vs. average packet length (in bytes) for the transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (100,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2 . . . . .	138
6.27	Normalized network throughput vs. normalized offered load for the transmission scheduling algorithms and continuous state estimation with realistic network traffic. The results are based on simulation. All other parameters are specified in Table 6.2 . . . . .	139
6.28	Normalized network throughput vs. normalized offered load for the threshold scheduling transmission scheduling algorithms and continuous state estimation with Poisson arrivals, exponentially distributed packet sizes. The codeset length is 10. The results are based on simulation. All other parameters are specified in Table 6.2 . . . . .	140
6.29	Normalized network throughput vs. normalized offered load for the threshold scheduling transmission scheduling algorithms and continuous state estimation with Poisson arrivals, exponentially distributed packet sizes. The codeset length is 100. The results are based on simulation. All other parameters are specified in Table 6.2 . . . . .	140
6.30	Normalized network throughput vs. normalized offered load for the threshold scheduling transmission scheduling algorithms and continuous state estimation with real traffic. The codeset length is 100. The results are based on simulation. All other parameters are specified in Table 6.2 . . . . .	141
7.1	A block diagram of the optical CDMA experimental testbed constructed at the USC Electrical Engineering department's Optical Communications Laboratory. . . . .	144

7.2	A block diagram of the modulator part of the Optical CDMA testbed. . . . .	145
7.3	A block diagram of the four encoders of the Optical CDMA testbed. . . . .	145
7.4	The optical CDMA encoders. The tunable delay lines (TDLs) of the encoder encode an optical CDMA codeword. . . . .	149
7.5	The receiver of the optical CDMA testbed. . . . .	151
7.6	The decoder of the optical CDMA receiver. The figure shows how the Tunable Delay Lines (TDLs) of the receiver decode the optical CDMA codeword.	151
7.7	Experimental results: The transmission scheduling algorithm results in the best case BER. It occurred under cases when there were zero overlaps. The worst case BER is the worst possible bit error rate that was measured. It occurred in cases when the number of overlaps was the maximum possible. Aloha-CDMA does not control the number of overlaps. Its BER is the average over all the measured cases. . . . .	160
7.8	A block diagram of an Interference Avoidance Network Interface Card. . . . .	163
7.9	State observation collection module. The input to the state observation collection module is a multivalued optical signal that arrives at the chipping rate. The output is an electronic digital value, a chip observation, which changes at the chipping rate. Each chip observation represents the magnitude of the multivalued signal in a chip time. The figure shows 4 threshold detectors, enough to detect up to 4 different chip magnitudes. The number of detectors can be modified based on the accuracy desired. Section 7.2.6 discusses why a minimum of 3 is needed. . . . .	165
7.10	State estimation module. The input is a stream of chip observations arriving at the chipping rate. The output is a state estimate that is output at the clock rate of the hardware. The figure also shows the width of the data lines in the hardware. . . . .	167
7.11	Transmission scheduling hardware. There are two inputs: the codeword that arrives on packet arrival and the state estimate, which changes every clock cycle. The output is the delay in terms of number of chips from the start of the hardware clock cycle. This delay is used to delay the transmission. The width of the data lines is also shown in the figure. . . . .	170

7.12	Optical CDMA transmitter hardware implementation. The input is a stream of bits. It modulates a stream of codewords being transmitted by the laser and encoder. . . . .	175
7.13	Optical CDMA receiver hardware. The input is the signal on the optical fiber at the chipping rate and the output is the received data at the bit rate. . . . .	176
7.14	Timing diagram (not to scale). The diagram shows the timing of events on a NIC for a 10Gc/s network using a codeset of length $N = 100$ , weight $w = 3$ . The hardware clock speed is 100MHz. The hardware design is as described in this section. A new state estimate is generated every 10ns. On packet arrival, the transmission scheduling hardware operates on its input and produces its result in 50ns. After tuning delays and the calculated delay ( $t_d$ ), the packet is transmitted. The residual chip time (the time between the delay line tuning and the start of the next hardware clock cycle) is also shown. . . . .	178
A.1	State can be represented mathematically by a pair $(n_0, n_1)$ where $n_0$ is the size of the zeros section and $n_1$ is the size of the ones section. . . . .	207
A.2	The initial few states of a state transition diagram. . . . .	209
F.1	The performance of Interference Avoidance vs. codeset length and codeset weight. The results are based on simulation. The graph shows the normalized network throughput at a normalized offered load of 1 and normalized propagation delay of $a = 0$ . The traffic model is Poisson arrivals, exponentially distributed packet sizes. . . . .	228

## List of Tables

3.1	Comparison of the state of the network in three shared medium networks: Single channel wired networks, single channel wireless networks, and optical CDMA networks. . . . .	51
3.2	Comparison of the properties of the state of the network in three shared medium networks: Single channel wired networks, single channel wireless networks, and optical CDMA networks. . . . .	52
5.1	The <i>Pure selfish</i> transmission scheduling algorithm. . . . .	74
5.2	The <i>Threshold</i> transmission scheduling algorithm. . . . .	76
5.3	The <i>Overlap section</i> transmission scheduling algorithm. . . . .	77
5.4	The <i>Centralized Perfect Scheduling</i> algorithm. . . . .	88
5.5	The <i>Distributed Perfect Scheduling</i> algorithm. . . . .	89
5.6	Parameter list and default values for the transmission scheduling sensitivity study. . . . .	93
6.1	The <i>Sample means</i> state estimation algorithm. . . . .	117
6.2	Parameter list and default values for the state estimation performance and sensitivity study. . . . .	123
7.1	Codewords for the 4 nodes in the experimental testbed. The codewords are from a 2 D codeset (more than one wavelengths is used) with parameters (11, 6, 6, 1). . . . .	146
7.2	Testbed bit error rate in the presence of 1 interfering user. . . . .	157

7.3	Testbed bit error rate in the presence of 2 interfering users. . . . .	158
7.4	Testbed bit error rate in the presence of 3 interfering users. . . . .	159
F.1	Reduction of Interference Avoidance to the Aloha, Aloha-CDMA, and CSMA protocols. . . . .	227

## Abstract

This work proposes *Interference Avoidance*, a contention media access control protocol for optical Code Division Multiple Access (CDMA) Local Area Networks (LANs). Optical CDMA is a direct sequence spread spectrum technology for multiplexing transmissions on an optical fiber. When used as the physical layer of a LAN, its throughput tends to zero at high offered load because of multiuser interference. Interference Avoidance prevents this throughput collapse of optical CDMA networks at high offered load.

An optical CDMA LAN exhibits two unique properties: *non-mutually destructive interference* and *correlation of state*. Interference Avoidance exploits these properties to reduce packet loss because of interference. It extends the Carrier Sensing mechanism to high speed, shared medium, optical networks. Interference Avoidance consists of *state estimation* and *transmission scheduling*. This work proposes algorithms for transmission scheduling and state estimation. The transmission scheduling algorithms stabilize the throughput at around 30% of the maximum at high offered load, under the assumption of perfect state estimation. Under identical conditions, throughput without transmission scheduling tends to zero. The algorithms achieve throughput within 10% of optimal codeword scheduling algorithms. The complexity of the algorithms is linear in codeset length, enabling low complexity hardware

implementation. The performance of the state estimation algorithms approaches that of perfect state estimation as the codeset length increases and weight decreases, with a low number ( $< 100$ ) of estimation samples. The complexity of the algorithms is linear in the number of estimation samples, allowing for low complexity hardware implementation. This work also demonstrates the feasibility of transmission scheduling using a laboratory testbed. Measurements from the testbed show that transmission scheduling reduces the bit error rate in the presence of severe physical layer noise and potentially improves throughput performance as load is increased.

# Chapter 1

## Introduction

Optical CDMA is a multiplexing technology for optical fiber. When used as the physical layer of a Local Area Network (LAN), the throughput without media access control collapses at high offered load. *Interference Avoidance* is a contention media access control protocol for optical CDMA LANs that prevents throughput collapse. This dissertation proposes Interference Avoidance and shows that it prevents throughput collapse in optical CDMA LANs.

This chapter introduces the concepts of optical CDMA, interference, and Interference Avoidance. Section 1.1 provides a brief description of optical CDMA. Section 1.2 discusses the problem of interference. Section 1.3 discusses the contribution of this thesis: the solution of the throughput collapse problem at high offered load through Interference Avoidance. Section 1.4 discusses the role that optical CDMA may play in future optical access networks. Section 1.5 describes the organization of the thesis.

The discussion in this chapter is brief and informal. Subsequent chapters provide formal definitions and in-depth discussions of topics covered here.

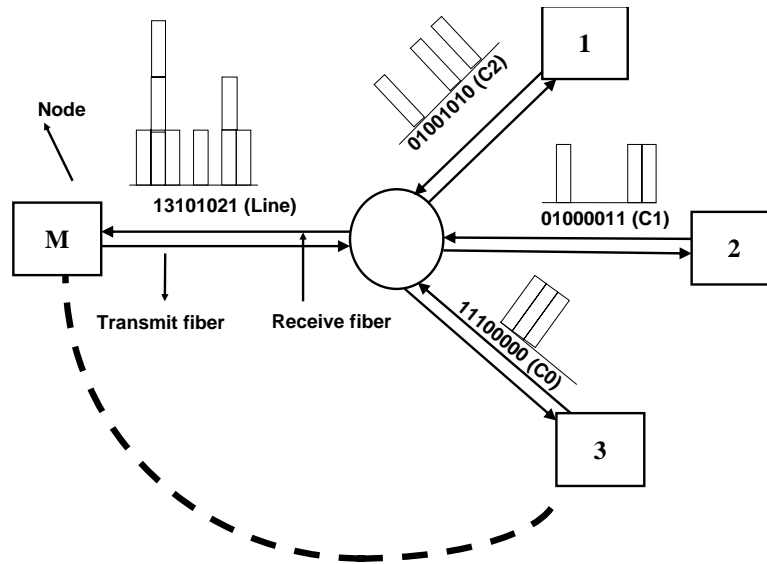


Figure 1.1: A typical optical CDMA network. The network consists of  $M$  nodes. The nodes are connected by transmit and receive fibers to a passive star coupler forming a shared medium LAN.

## 1.1 Optical CDMA

Optical CDMA is a multiplexing technology for optical fiber. This work considers a *packet switched, shared medium, multiple access* Local Area Network that uses optical CDMA as the physical layer.

This work considers optical CDMA based on *intensity modulation* and *unipolar encoding*. Data is encoded using optical CDMA codewords. An optical CDMA codeword is a distinct sequence of 0s and 1s (the 0s and the 1s are called chips)<sup>1</sup> A node transmits data by ON-OFF keying of the codeword, *i.e.*, to send a 1 bit, the nodes sends the codeword and to send a 0 bit, an all zeroes codeword is sent. In intensity modulation an optical pulse is a ‘1

<sup>1</sup>This work uses single quotes to distinguish between the magnitude and quantity of chips. The term ‘1 chip’ refers to a chip of magnitude 1.

chip’, while the absence of a pulse is a ‘0 chip’. Unipolar encoding refers to the encoding of data using sequences of ‘0 chips’ and ‘1 chips’. Figure 1.1 shows three nodes transmitting 1 bits using three codewords  $C0$ ,  $C1$  and  $C2$ . Multiple nodes transmit simultaneously using different codewords.

The network consists of several nodes connected by optical fiber to a passive star coupler as shown in Figure 1.1. The fibers and the coupler form a *low noise, guided medium*. Signals transmitted on the inputs enter the coupler, merge, and are transmitted on all outputs. The star coupler is passive, *i.e.*, the input power is split equally among the receive fibers and is transmitted to all nodes on the receive fibers. The signal at the output of the coupler on any receive fiber at time  $t$  is given by:

$$r(t) = (1/K) \sum_{i=1}^K s_i(t)$$

where  $K$  is the number of input (or output) ports of coupler and  $s_i(t)$  is the signal entering on the  $i^{th}$  transmit fiber at time  $t$ . The signals on the transmit fibers  $s_i(t)$  are binary optical signals and the signal on a receive fiber  $r(t)$  is a multilevel optical signal. The signal on the receive fiber may be amplified or attenuated after the coupler. In Figure 1.1 the merged signal is [1 3 1 0 1 0 2 1]<sup>1</sup>.

The network is based on a *Tunable Transmitter-Fixed Receiver (TT-FR)* architecture. A receiver chooses a codeword to receive on and a transmitter that needs to communicate with a

---

<sup>1</sup>The figure indicates that transmission is chip synchronous, *i.e.*, chip boundaries are aligned in time. In reality this need not be true. However, it is assumed for simplicity of explanation. Chapter 3 justifies this as a reasonable assumption for analysis. This work also assumes that there are amplifiers on every receive fiber after the coupler which amplify the signal  $K$  times.

receiver tunes to the receiver's codeword. The network uses *codeword sharing*. If the number of nodes is greater than the codewords, the codewords are shared among receivers. Packets sharing a codeword are demultiplexed using a higher layer unique identifier such as a link layer address. Every node runs a *frame synchronization algorithm* [29] that allows the node to identify that a frame destined for it has arrived and where the first bit of the frame begins. Chapter 4 discusses these three architectural choices.

Every node has a *hard-limiting correlation* receiver [57]. The receiver is tuned to a particular codeword. It consists of an optical hard-limiter, a decoder, a photodetector, and a threshold detector. The signal on the optical fiber is a multilevel optical signal. It is first passed through an optical hard-limiter. The hard-limiter limits the energy in each chip position. The decoder consists of a splitter and delay lines. It collects energy in the chip positions where it expects to see '1 chips'. The energy is then sent to a photodetector that converts it to a multilevel electrical signal. The signal is then sent through a threshold detector whose threshold is set proportional to the weight  $w$  of the codeset, *i.e.*, to the number of '1 chips' in the codeword. If the energy is equal to or greater than the *threshold*, a 1 data bit is detected, else a 0 bit is detected. Let  $R$  be the received signal (a vector of length  $N$  whose components are nonnegative integers), and  $C$  the codeword being received (a vector of length  $N$  whose components are binary values). Let  $R = [r_0 r_1 r_2 \dots r_{N-1}]$  and  $C = [c_0 c_1 c_2 \dots c_{N-1}]$ . Then the received bit  $b$  is given by:

$$\begin{aligned}
 b &= 1 \quad \text{if } (C \cdot h(R) \geq w) \\
 &= 0 \quad \text{otherwise}
 \end{aligned}$$

where ‘ $\cdot$ ’ represents the dot product defined as

$$U \cdot V = [u_0 u_1 \dots u_{N-1}] \cdot [v_0 v_1 \dots v_{N-1}] = \sum_{i=0}^{N-1} u_i v_i$$

and  $h(\cdot)$  is the hard-limiting operation defined as

$$h(\mathbf{R}) = [s_0 s_1 \dots s_{N-1}]$$

$$\text{where } s_i = 0 \quad \text{if } 0 \leq r_i < 1$$

$$s_i = 1 \quad \text{if } r_i \geq 1$$

Chapter 4 discusses the architecture of the network in detail.

## 1.2 The problem: Interference

The throughput of an optical CDMA LAN at high offered load collapses because of interference between multiplexed codewords.

Figure 1.2 is a snapshot of data bits on the receive fiber of an optical CDMA LAN. It shows three codewords sent by three nodes (codewords  $C0$ ,  $C1$ , and  $C2$ ). The figure also shows the signal on the line (any receive fiber) which is also called the state of the line (state is defined formally in Chapter 3). A node on the network has a receiver tuned to codeword  $C0$  and is receiving transmissions sent using codeword  $C0$ . The time  $t = 0$  on the time scale shown in the figure marks the start of a bit of codeword  $C0$ . Figure 1.2(a) shows a case where a 0 bit is being received and Figure 1.2(b) shows a case where a 1 bit is being received.

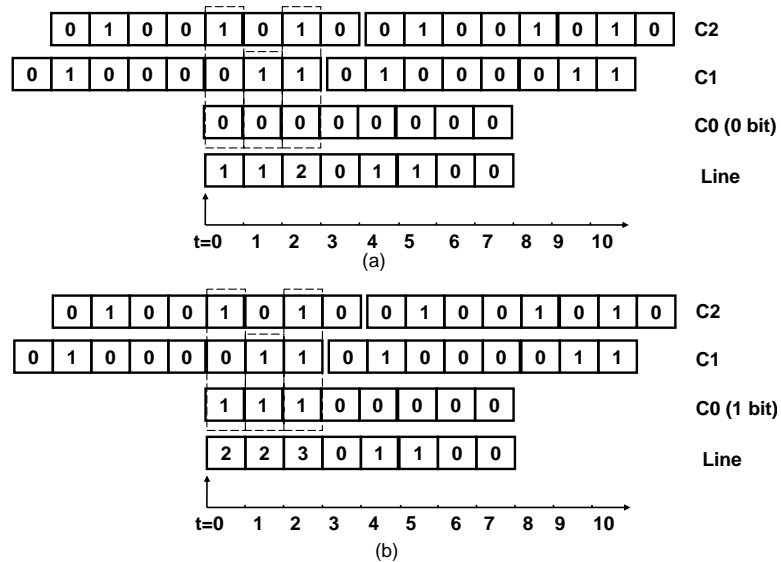


Figure 1.2: An example of multi-user interference showing three codewords  $C0$ ,  $C1$ , and  $C2$ . The two codewords ( $C1$  and  $C2$ ) interfere with another codeword ( $C0$ ). (a) shows the line (fiber) when  $C0$  is OFF (transmitting a 0 bit) and (b) shows the line when  $C0$  is ON (transmitting a 1 bit). In (a),  $C0$  experiences an interference error.

In Figure 1.2(a) the codewords  $C1$  and  $C2$  add up such that there is power in the 1st, 2nd, and 3rd chips, among others. Because of codewords  $C1$  and  $C2$ , the receiver tuned to  $C0$  detects a 1 bit whether the transmitter sends a 1 bit or a 0 bit. As shown in Figure 1.2(a), when a 0 bit is sent, the receiver detects a 1 bit. The packet would have an error and be lost. The overlap of chips is termed *multi-user interference* or *multi-access interference* or simply *interference*. The error is called an *interference error*. Note that interference does not necessarily cause interference errors, *i.e.*, there can be chip overlaps without any interference errors. Chapter 3 formalizes the conditions for an interference error. At high offered load interference leads to throughput collapse. Figure 1.3 shows the throughput of an optical CDMA

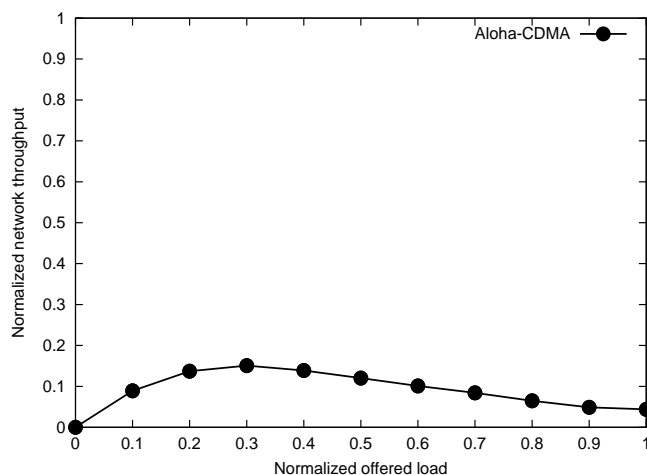


Figure 1.3: The normalized network throughput vs. normalized offered load for Aloha-CDMA. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is  $(10, 3, 3)$ . The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

network without media access control (Aloha-CDMA) as the offered load is increased. At high offered load (near or above 1), the throughput tends toward zero.

The problem of throughput collapse in optical CDMA LANs is conceptually similar to the problem of throughput collapse in single channel, shared medium networks. Single channel, shared medium networks without media access control (Aloha networks) suffer throughput collapse at high offered load because of collisions between packets. Carrier Sensing Multiple Access (CSMA) and its variants [68] have been proposed as solutions to this problem. They have been shown both analytically and experimentally to prevent throughput collapse by reducing packet collisions. However, the throughput of CSMA at high offered load decreases as the speed of the network increases for a network of fixed diameter and packet size. For typical packet sizes (1500 bytes) and network diameters (1000m), CSMA is limited to

network speeds of less than 100 Mb/s. At high speeds (in the range of 10 Gb/s) CSMA is ineffective.

The problem statement of this dissertation is: *Can Carrier Sensing-like protocols prevent throughput collapse in high speed optical CDMA LANs?*

This dissertation answers this question by proposing Interference Avoidance, a contention media access control protocol that extends Carrier Sensing to high speed networks.

### **1.3 The solution: Interference Avoidance**

*Interference Avoidance* is a contention media access control protocol that prevents throughput collapse at high offered load by estimating the state of the network and scheduling packet transmissions to reduce interference errors.

Consider the codewords shown in Figure 1.4. The codewords  $C0$ ,  $C1$ ,  $C2$  are the same as in Figure 1.2. The figure also shows the signal on the line (fiber) which is also called the state of the line. The state at a point on the line at a time is informally defined as the sum of the codewords on the line at that point, *e.g.*, the state of the line at  $t = 0$  in Figure 1.2(a) is [ 1 1 2 0 1 1 0 0 ]. Chapter 3 defines state formally and discusses its properties.

Codeword  $C0$  has been transmitted one chip time later compared to Figure 1.2. Now a receiver tuned to  $C0$  detects correctly whether the transmitter sends a 0 bit or a 1 bit. This is because when the transmitter transmits  $C0$  with a delay of one chip time,  $C0$  has at least one chip that does not interfere with codewords  $C1$  and  $C2$ . Therefore, a false positive error does not occur and the receiver receives the 0 or 1 bit correctly.

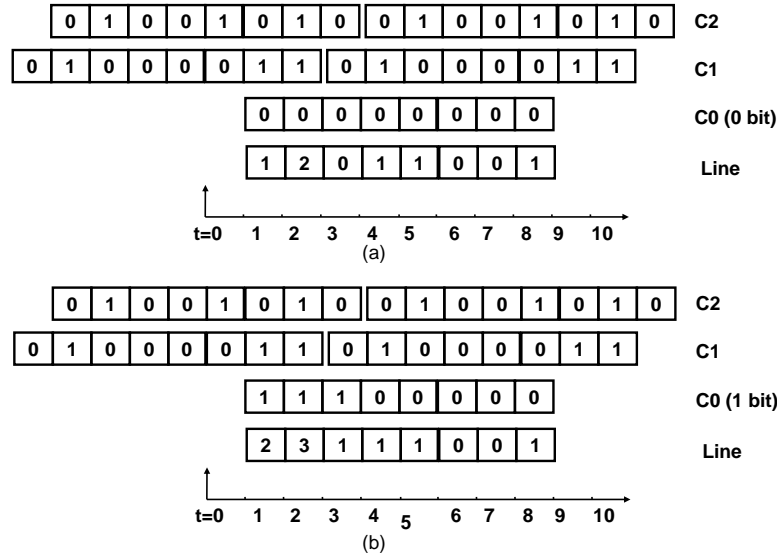


Figure 1.4: The effect of Interference Avoidance. The codewords are the same as in Figure 1.2.  $C0$  has been transmitted one chip time later compared to Figure 1.2. (a) shows the line (fiber) when  $C0$  is OFF and (b) shows the line when  $C0$  is ON. In either case,  $C0$  does not have an interference error.

This principle can be translated into a media access control protocol. The resulting protocol is called *Interference Avoidance*. Each node on the network contends for access to the medium using the Interference Avoidance protocol. This work shows that Interference Avoidance can prevent throughput collapse at high offered load. It may be viewed as an extension of carrier sensing to high speed optical CDMA LANs. The thesis statement of this dissertation is:

*Throughput collapse in high speed optical CDMA LANs can be prevented by Interference Avoidance, a Carrier Sensing-like protocol.*

The following sections give an overview of the properties of an optical CDMA network and how they make Interference Avoidance feasible.

### 1.3.1 State of an optical CDMA network

The state of an optical CDMA LAN is the sum of several overlapping codewords. The state is used to show that optical CDMA LAN possesses two properties:

- Non-mutually destructive interference: Interference can cause errors in a subset of the codewords on the line. The subset depends on the codewords and their delays. This property allows transmissions to be scheduled to reduce interference loss (*transmission scheduling*).
- Correlation of state: State at point (on a receive fiber) and time is correlated to the state at some other point and time. This property allows state to be estimated (*state estimation*).

There are two reasons for discussing these properties: First, these two properties make transmission scheduling and state estimation feasible. Secondly, these properties differentiate the problem of throughput collapse in optical CDMA LANs from the corresponding problem in other physical layers. Chapter 3 describes the state, its properties and these reasons in detail.

### 1.3.2 State estimation

*State estimation* is the process by which a node estimates the state of the line.

Chapter 6 formulates the state estimation problem. It demonstrates how correlation of state is a necessary condition for accurate state estimation. It discusses the distribution of the

state and proposes a state estimation procedure consisting of collecting observations and estimation. The performance of state estimation is analyzed under different topology conditions, codeset parameters and traffic conditions. In particular, this work shows that as the codeset length increases the performance of state estimation approaches optimal performance.

### **1.3.3 Transmission scheduling**

*Transmission scheduling* is the process by which a node schedules a packet transmission based on the estimated state and the codeword to be used for encoding. The objective is to improve throughput at high offered load by reducing packet loss because of interference.

Chapter 5 formulates the transmission scheduling problem. It discusses strategies for transmission scheduling. The strategies are used to define three scheduling algorithms: pure selfish, threshold and overlap section. The algorithms differ in how transmissions are scheduled to avoid interference loss. The algorithms are analyzed and simulated under different codeset parameters and traffic conditions under the assumption of perfect state estimation. Chapter 5 shows that the proposed transmission scheduling algorithms prevent throughput collapse when used with low codeset weights. They improve throughput to around 30% of the maximum at high offered load. Under identical conditions, systems with no transmission scheduling have close to zero throughput. The algorithms perform within 10% of an optimal codeword scheduling algorithm. This study demonstrates that when all the configurable parameters (codeset, transmission scheduling and state estimation) are chosen correctly, throughput collapse can be prevented at high offered load.

Finally, the performance of Interference Avoidance under laboratory conditions is measured. Chapter 7 describes a laboratory testbed that is used to measure the performance of transmission scheduling. Measurements from the testbed show that transmission scheduling improves the bit error rate (BER) by two orders of magnitude in the presence of physical layer noise.

## **1.4 Optical CDMA and the future of optical access networks**

This section discusses the role that optical CDMA may play in the evolution of access networks. The term *access network* refers to networks that connect residential users to a service provider. Shared medium access networks such as the cable network are currently being converted from hybrid fiber coaxial networks to all-fiber networks (Fiber To The Home (FTTH) networks) to take advantage of their large capacity and provide a diverse range of services.

There are limitations on how much capacity of an optical fiber may be utilized. This means that in the near future (5-10 years) optical access networks may evolve to use a mixture of switching, multiplexing and media access control technologies. Applications will use the most appropriate switching, multiplexing or media access control technology that satisfies their performance requirements. Optical CDMA with Interference Avoidance is a packet switched, CDMA, contention protocol with low latency, low packet error rates and moderate throughput at high offered load. It is particularly suited to the needs of delay sensitive, loss tolerant applications.

### **1.4.1 Utilization of the capacity of an optical fiber**

Optical fiber is well suited for the transmission of data because of its large bandwidth (up to 15THz) and low attenuation (0.2dB/km). Studies have shown that the transmission capacity of this bandwidth could be around 100 Tb/s [43]. Currently, only about 40 GHz (around 40 Gb/s) of this bandwidth can be used for communication. The maximum rate at which a single stream of data can be transmitted today is around 40 Gb/s (OC-768) although experiments have shown it can be increased to 160 Gb/s [37]. This is the maximum rate at which a laser can be modulated, *i.e.*, the electronic processing speed. So current hardware can utilize only a fraction of the capacity of an optical fiber. Slow progress in the development of laser technology means that this may be the status for the next 5-10 years. Therefore, in the near term, optical networks will need to share the capacity through some form of multiplexing and/or media access control.

Multiplexing in current optical access networks is by Wavelength Division Multiplexing (WDM), Time Division Multiplexing (TDM) or reservation media access control protocols. In WDM, the bandwidth of the fiber is divided and several nodes transmit in parallel by intensity modulation of different wavelengths. In TDM systems such as SONET, time is divided into slots, and different nodes transmit in different slots. In reservation protocols such as DOCSIS [19], nodes request transmission slots from a centralized controller. WDM is characterized by low access latency, high throughput and low packet error at the cost of expensive equipment such as tunable lasers. TDM is characterized by high access latency, high throughput and low hardware costs. Reservation protocols are characterized by moderate access latency, moderate throughput and low hardware costs.

### **1.4.2 Role of optical CDMA in optical access networks**

Different applications that share the optical access shared medium have different performance requirements. This means they may require different multiplexing/media access control technologies. Current Internet traffic is primarily composed of six applications: web, file sharing, file transfer, email, streaming media (video and audio) and multi-user games. Each of these has different performance requirements. Optical CDMA with Interference Avoidance is a multiplexing and media access control technology with characteristics of low access latency (compared to TDM/reservation protocols [50]), low packet error rate (Chapter 5 and 6), the ability to coexist with other multiplexing technologies [12] and the potential for low cost transmitters and receivers [60]. These characteristics are well suited for delay sensitive, loss tolerant applications such as voice and video. They make Optical CDMA a suitable access technology for specific applications on shared medium networks.

## **1.5 Organization**

The following chapters justify the assertion of this dissertation, *i.e.*, that a generalization of carrier sensing can exploit the properties of optical CDMA to prevent throughput collapse.

Chapter 2 is a primer on Optical CDMA. It discusses the characteristics of optical CDMA LANs, optical CDMA codeset design, transmitter and receiver design and network architecture. Chapter 3 is the problem statement of this work. It discusses the problem of interference, the concept of state and its properties and why they differ from problems studied before. It defines Interference Avoidance and shows that optical CDMA possesses properties that make

Interference Avoidance feasible. Finally, it shows that Interference Avoidance is a generalization of Carrier Sensing. The next three chapters discuss the functionality of Interference Avoidance. Chapter 4 provides an overview of the architecture of an optical CDMA Local Area Network. Chapter 5 discusses transmission scheduling strategies and algorithms. It analyzes the algorithms and compares their performance to optimal codeword scheduling algorithms. The sensitivity of the algorithms to network and traffic parameters is discussed. Chapter 6 defines state and state estimation algorithms. Performance of the system under different network and traffic parameters is discussed. Chapter 7 discusses the implementation of a prototype and presents measurements taken from a laboratory testbed. Chapter 8 is a survey of related work. Chapter 9 discusses conclusions and open areas of research.

## Chapter 2

### A primer on optical CDMA

This chapter is an introduction to optical CDMA. Optical CDMA is a direct sequence spread spectrum technology [55] used for multiplexing optical transmissions on an optical fiber. An optical CDMA network in this work refers to an all-optical, packet switched, shared medium, Local Area Network (LAN) [58] [57] that uses optical CDMA as the physical layer. The network consists of several nodes connected by optical fiber to a star coupler to form a low noise, guided medium. The optical signal is modulated using *intensity modulation* and encoding is *unipolar encoding*. The receiver is a *hard-limiting correlation receiver*.

Section 2.1 discusses the characteristics of the optical CDMA physical layer (intensity modulation and unipolar encoding) and the physical medium (low noise, guided medium). Section 2.2 discusses optical CDMA codesets, their design parameters and construction methods. Section 2.3 discusses the design of optical CDMA transmitters and receivers. Other architectural and design issues at the physical and link layer (topology, codeword allocation, addressing, frame format and link layer error correction/detection) are discussed in Chapter 4.

## 2.1 Optical CDMA channel characteristics

This section describes the characteristics of the physical layer. The intensity modulation and unipolar encoding differentiate optical CDMA from other multichannel physical layers such as TDM, WDM or wireless orthogonal CDMA. The low noise, guided medium differentiates optical CDMA from the other physical medium where CDMA has been applied before, *i.e.*, the electromagnetic field on the wireless medium.

### 2.1.1 Intensity modulation and unipolar encoding

Optical CDMA is based on *unipolar encoding* (also called time domain encoding) [54] [57] [58] [36]. Data is encoded using a codeword sequence. An optical CDMA codeword is a sequence of 0s and 1s. The 0s and 1s are transmitted by binary *intensity modulation*, *i.e.*, a pulse of light is a 1 and no pulse of light is a 0. Each 0 or 1 of a codeword is called a *chip*, and the codeword represents a data *bit*. Bits are transmitted by ON-OFF keying of the codeword, *i.e.*, to transmit a 1 bit, the codeword is sent and to transmit a zero bit, a codeword of all zero chips is sent. The term ON refers to the transmission of a 1 bit and OFF refers to the transmission of a 0 bit. The term *intensity modulation* refers to the transmission of binary signals using pulses of optical light. The term *unipolar encoding* refers to the use of ON-OFF keying of codeword sequences to encode bits. Intensity modulation and unipolar encoding are illustrated in Figure 2.1. The data is 10010110. The codeword is 0100011.

Unipolar encoding has been widely studied due to simplicity in construction of the transmitters and receivers. Other encoding techniques such as spectral phase encoding [59] and Code Cyclic Modulation (CCM) [35] have been proposed. In spectral phase encoding, an

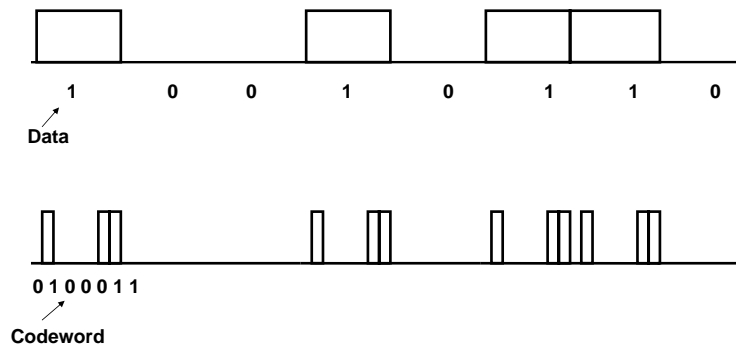


Figure 2.1: The ON-OFF keying of codewords. The lower line shows the signal on the optical fiber when a single codeword (0100011) is being transmitted. The upper line shows the corresponding data.

optical pulse is decomposed in the spectral domain. Data is transmitted by encoding each spectral component using a phase shift. In Code Cyclic Modulation, symbols are encoded using rotations of code sequences. Both forms of encoding require complex transmitter and receiver design. As a result, unipolar encoding has been more widely studied and is better understood. For the remainder of this work, it is assumed that unipolar encoding is used.

Unipolar encoding in optical CDMA differs from conventional CDMA encoding in wireless networks. Wireless networks use different modulation and encoding formats. Coherent modulation (phase, amplitude, or frequency) and bipolar encoding are most frequently used. In bipolar encoding codewords use +1 and -1 signals, thereby allowing the construction of orthogonal codewords. In contrast, optical CDMA codewords use 0 and +1, transmitted

asynchronously, and therefore cannot be designed to be orthogonal. They are called *pseudo-orthogonal*.

### 2.1.2 Guided, low noise medium

The optical CDMA LAN consists of several nodes connected by optical fiber to a passive star coupler. The optical fiber is single mode dispersion compensated optical fiber. The passive star coupler multiplexes the optical signals from all the nodes. It consists of several inputs and output ports. Each node is connected to one input and one output port by an optical fiber. The nodes and the passive star coupler form a shared medium network. The fiber connected to the input port is called the transmit fiber and the other is the receive fiber. Signals transmitted on the inputs enter the coupler, merge, and are transmitted on all outputs. This work assumes that the power transmitted by each node on the network is the same. The star coupler is passive, *i.e.*, input signals pass through without any electronic processing. The coupler is a divide-by- $N$  coupler [28]. When data enters the coupler on several transmit fibers, the optical power from all transmit fibers is merged. The power is then split equally among the receive fibers. The same signal emerges on all the receiver fibers of the coupler and is transmitted to all nodes. Figure 1.1 shows a typical optical CDMA network topology. Three codewords are shown entering the coupler and the merged signal emerges. The signal at an output port of the coupler is given by the equation:

$$r(t) = (1/K) \sum_{i=1}^K s_i(t)$$

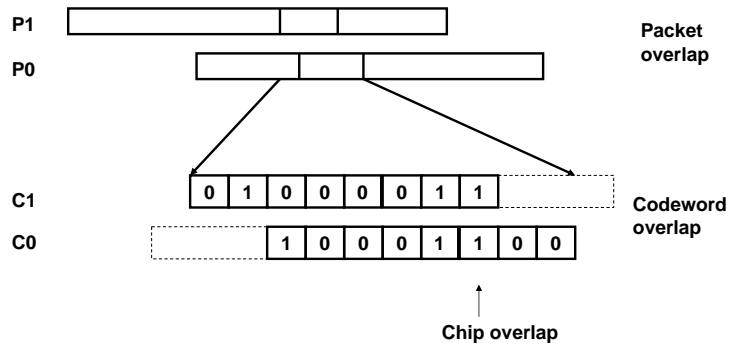


Figure 2.2: Packet, codeword and chip overlaps. This figure depicts chip synchronous transmission. The effect of chip synchronous and asynchronous transmission is discussed in Chapter 3.

where  $K$  is the number of input (or output) ports of the coupler and  $s_i(t)$  is the signal entering on the  $i^{th}$  transmit fiber. The signal on the receive fiber can be amplified or attenuated after the coupler to obtain a required power level. Therefore, for ease of explanation, diagrams and explanations in this work may indicate that the signal on a receive fiber is:

$$r(t) = \sum_{i=1}^K s_i(t)$$

The fiber and the coupler form a guided medium. A guided medium is a medium where all nodes on the network see the same signal (perhaps after different propagation delays) [28].

A *packet overlap* occurs when two or more packets are at the same point in space on the optic fiber (line) at the same time. Figure 2.2 shows a packet overlap on the line. When

a packet overlap occurs, the codewords of the packets overlap. When a codeword overlap occurs, two or more ‘1 chips’ of different codewords may overlap. This is termed a *chip overlap*. Codeword overlaps and chip overlaps are also shown in Figure 2.2. When a chip overlap occurs, the power of the optical pulses are added in the coupler. The *chip length*  $l$  is the length of a chip on the fiber. It is given by  $l = c/B$  where  $c$  is the speed of light in fiber and  $B$  is the chipping rate in chips per second ( $c/s$ ). The *chip magnitude* at a point on a fiber at a time is the number of chips at that point at that time. Any point on the fiber where the chip magnitude is greater than 1 is chip overlap.

The medium is a *low noise* medium, *i.e.*, it exhibits the following properties:

- Low noise: Physical layer noise such as thermal, ASE and shot noise can cause bit errors. However, these effects are negligible compared to the effects of multi-user interference at high loads. Other work [57] has explored these effects in detail. Therefore, this work neglects these effects.
- Low attenuation: The optical fiber has relatively low attenuation (0.2 dB/km) [28] compared to the other media such as copper or wireless. The effects of attenuation can be eliminated by placement of amplifiers at suitable points in the network. Therefore, this work assumes that there is no attenuation in the optical fiber.
- Low dispersion of the fiber: Dispersion in single mode optical fiber is low (1 ps/nm km). This work neglects dispersion because it can be reduced to a negligible effect by using dispersion compensated fiber.

The low noise, guided medium characteristic distinguishes the optical CDMA medium from the wireless medium where CDMA has been traditionally applied. The wireless medium is an unguided noisy medium. Transmissions from a node follow different paths to different receivers. As a result, different nodes on the network see different signals. This contrasts with optical CDMA where every node sees the same signal. Further, the wireless medium is noisy. The power of signals attenuate as the square of the distance and the signal experiences random fluctuations because of slow and fast fading [55]. The next few chapters will show how this low noise, guided medium property of optical CDMA may be exploited.

## 2.2 Optical CDMA codeset design

An Optical Orthogonal Codeset (OOC) [10] is a set of (0,1) sequences of length  $N$  that satisfies certain autocorrelation and crosscorrelation constraints. The term *codeset* refers to the set of sequences, and a *codeword* is a member of the set. The number  $w$  of ‘1 chips’ of a codeword of the codeset is called its Hamming weight or just *weight*. Codesets with all codewords having the same weight are called *constant weight* codesets.

The rate at which individual chips are transmitted is called the *chipping rate*. The rate at which the data bits are transmitted is called the *data rate*. The chipping rate is  $N$  times the data rate. The *chip time*  $t_c$  is the time required to transmit a chip. It is given by  $t_c = 1/B$ . The *bit time*  $t_b$  is the time required to transmit a bit. It is given by  $t_b = Nt_c = N/B$ .

OOCs are constructed such that the interference between pairs of codewords is constrained (interference is discussed in detail in Chapter 3). This is done by constructing the codeset such that the codewords obey auto and crosscorrelation constraints. These constraints

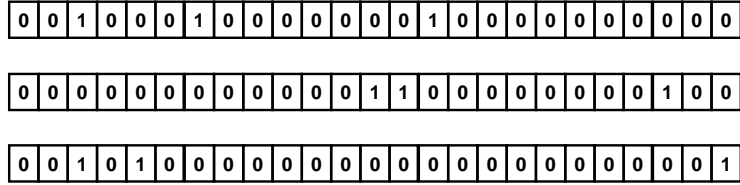


Figure 2.3: Three codewords of an optical CDMA codeset whose parameters are codeset length  $N = 25$ , weight  $w = 3$  and maximum crosscorrelation parameter  $\kappa = 1$ .

limit the number of overlapping chips between pairs of codewords. For any two codewords in the codeset, the correlation constraints are:

$$\sum_{n=0}^{N-1} s(i, n + \tau) s(j, n) \begin{cases} = w & \text{when } i = j, \tau = 0 \\ \leq \kappa & \text{otherwise} \end{cases}$$

where  $s(i, n)$  is the  $n^{th}$  chip of the  $i^{th}$  codeword,  $0 \leq \tau \leq N - 1$  and addition is modulo  $N$ . A codeset designed under these constraints is said to be *pseudo-orthogonal*.  $\kappa$  is called the *Maximum Collision Parameter*. A particular codeset is specified by the parameters  $(N, w, \kappa)$ .

Figure 2.3 shows 3 codewords of a  $(25, 3, 1)$  codeset.

The *size* of a codeset is the number of codewords in the codeset. The size  $S$  of a constant weight codeset constructed under the pseudo-orthogonality constraints can be shown to satisfy the Johnson bound [23]:

$$S(N, w, \kappa) \leq \left\lfloor \frac{1}{w} \left\lfloor \frac{N-1}{w-1} \left\lfloor \frac{N-2}{w-2} \left[ \dots \left\lfloor \frac{N-\kappa}{w-\kappa} \right\rfloor \right] \dots \right\rfloor \right\rfloor \right\rfloor$$

Codeset design for optical CDMA has been studied extensively [10] [9] [11]. Algorithms for codeset design methods include projective geometry, greedy and iterative constructions [10]. These methods differ in how closely they approach the Johnson bound. The choice of the algorithm to generate a codeset depends on the codeset parameters. Different algorithms may be closer to the bound for different codeset parameters. Codesets described in this work have been designed using the greedy algorithm.

Codesets can be extended by using other orthogonal dimensions such as wavelength or polarization [67] for encoding. A codeset that uses multiple dimensions is called a multidimensional codeset. A multidimensional codeset with multiple wavelengths is specified by the parameters  $(N, \Lambda, w, \kappa)$ . In this work, when three parameters are specified as part of codeset description, it may be assumed to be a one dimensional codeset. When four or more parameters are specified as part of the codeset description, it may be assumed to be a multidimensional codeset. Construction methods for one dimensional codesets can be extended to multidimensional codesets [52].

The use of unipolar encoding (0/1 signals) makes it impossible to construct asynchronous optical CDMA codewords that are orthogonal (having a cross correlation parameter equal to

zero). As a result, optical CDMA codewords are *pseudo-orthogonal*. Note that the cross-correlation constraint limits interference between pairs of codewords. Therefore, when more than two codewords are multiplexed on a fiber there may still be interference between the codewords. This problem is much more severe than wireless CDMA which can use bipolar orthogonal codesets [38]. Chapter 3 discusses the problem of interference in detail.

## **2.3 Optical CDMA transmitter and receiver design**

The transmitter and receiver design determines how packet transmissions, interference, packet errors, and packet receptions happen. Optical CDMA systems are Intensity Modulation-Direct Detection (IM-DD) systems [28]. This means that data is transmitted by using power to represent a chip. Data is received by conversion of the optical power to electrical energy using a photo detector.

### **2.3.1 Optical CDMA transmitters**

Optical CDMA transmitters (also called encoders) are intensity modulation based transmitters. The transmitter modulates the data using the optical CDMA codeword and transmits the modulated data on the fiber. An optical CDMA transmitter design is described in [57]. It consists of a laser source that is externally modulated at electronic speeds by the data. The encoder is built from passive optical components (delay lines, splitters, and couplers).

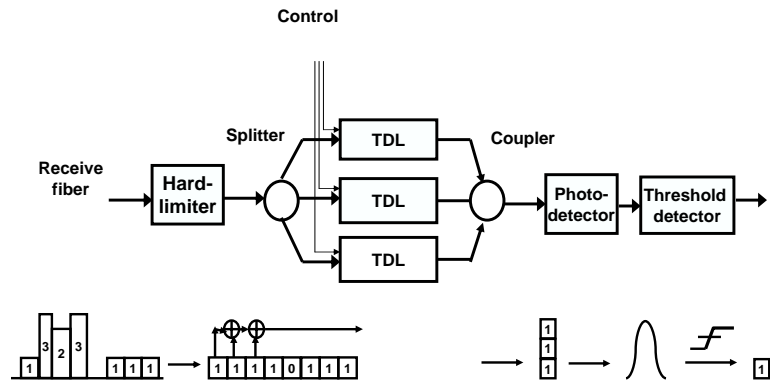


Figure 2.4: An optical CDMA receiver. It is a hard-limiting correlation detector that consists of a hard-limiter, decoder, a photodetector, and a threshold detector. The receiver is tuned to the codeword 1110000. The power in the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> chip positions is summed by the decoder. The photodetector converts the signal to current and the threshold detector detects a 1 bit.

### 2.3.2 Optical CDMA receivers

The optical CDMA receiver (also called a decoder or threshold detector) is a hard-limiting *correlation* receiver [57]. The receiver decodes the codeword in the received signal and regenerates the transmitted data. Figure 2.4 depicts the operation of a receiver. The input signal from the coupler is a multilevel optical signal. The receiver converts it to a digital optical signal by hard-limiting the power in each chip of the received signal. It then decodes the signal to detect a 1 or 0 bit.

The receiver consists of an optical hard-limiter, a decoder, a photodetector, and a threshold detector. The multilevel signal is first passed through the optical hard-limiter. The hard-limiter limits the energy in each chip position. The decoder consists of a splitter and delay lines. It collects energy in the chip positions where it expects to see 1 chips. The energy is then sent to a photodetector that converts it to a multilevel electrical signal. The signal is then sent through a threshold detector whose threshold is set proportional to the weight of the codeset, *i.e.*, in the figure this is an electrical power proportional to 3 chips. If the energy is equal to or exceeds the *threshold*, a ‘1’ data bit is detected, else a ‘0’ bit is detected.

In Figure 2.4, the receiver is tuned to codeword  $C0 = 11100000$ . The power in the 2nd chip is 3 units before the hard-limiter. After the hard-limiter, it is reduced to 1 unit. The decoder collects the energy in the 1st, 2nd, and 3rd chip. The threshold detector detects a 1 bit.

Let  $R$  be the received signal (a vector of length  $N$  whose components are nonnegative integers), and  $C$  be the codeword being received (a vector of length  $N$  whose components are binary values). Let  $R = [r_0 r_1 r_2 \dots r_{N-1}]$  and  $C = [c_0 c_1 c_2 \dots c_{N-1}]$ . Then the received bit  $b$  is given by the equation:

$$b = 1 \quad \text{if } (C \cdot h(R) \geq w)$$

$$= 0 \quad \text{otherwise}$$

where  $\cdot$  represents the dot product defined as

$$U \cdot V = [u_0 u_1 \dots u_{N-1}] \cdot [v_0 v_1 \dots v_{N-1}] = \sum_{i=0}^{N-1} u_i v_i$$

and  $h(\cdot)$  is the hard-limiting operation defined as

$$h(R) = [s_0 s_1 \dots s_i \dots s_{N-1}]$$

$$\text{where } s_i = 0 \quad \text{if } 0 \leq r_i < 1$$

$$s_i = 1 \quad \text{if } r_i \geq 1$$

The discussion so far has assumed that the receiver has acquired both chip and bit synchronization. In practice, before a frame is received, there needs to be a *frame synchronization phase* where the receiver detects where the frame begins. The frame synchronization sequence and phase is discussed in more detail in Chapter 4.

Other optical CDMA receiver designs such as the multi-user detector [49] and the balanced detector [14] have been proposed. The complexity (number of components) of the multi-user detector increases exponentially as the length of the codeset. The hard-limiting correlation based receiver is widely accepted as the receiver with the lowest implementation complexity and this work assumes that it is the receiver design.

## 2.4 Optical CDMA errors

This section defines all possible optical CDMA errors and specifies which of those errors this work deals with. An optical CDMA error is an error that occurs in a packet due to a property of optical CDMA. The following are a list of optical CDMA errors:

- **Physical layer errors:** Errors can occur due to characteristics of the optical fiber medium (such as thermal noise, shot noise, Amplifier Spontaneous Emission (ASE), beat noise). Other work [28] discusses these effects in detail. Moreover, at high offered load, these effects are negligible compared to the effect of multi-user interference. Therefore, these errors are not discussed in this work.
- **Interference errors:** An interference error occurs when the sum of the codewords on the line adds up to another codeword. An interference error causes a false positive detection at a receiver. If this error occurs when the receiver is receiving a frame, it causes an error in the frame and loss of the packet. This work deals exclusively with interference errors. Interference errors have been handled through FEC [47] or centralized control of codeword allocation [63]. This work does not assume the use of any forward error correction. This work studies the use of media access control to handle these errors.
- **Synchronization errors:** Synchronization [29] is the process of detecting the start of a frame. It is usually done by detection of a synchronization sequence in the preamble of the frame. A synchronization error occurs when the sum of the codewords on the line adds up to the initial bit of the synchronization sequence of some receiver on

the network. Because of this, the receiver inadvertently enters the frame synchronization phase. Synchronization and synchronization errors in an optical CDMA network have been studied and quantified in [29]. That work proposes a serial search synchronization algorithm and quantifies the synchronization errors. This work assumes that synchronization is instantaneous and that there are no synchronization errors to simplify the analysis of the network under heavy load. To determine overall throughput of the network, the effect of synchronization errors must be added to the interference losses.

- Receiver contention errors: A receiver contention error occurs when two packets are simultaneously sent to the same node. If the receiving node has only one receiver then it can receive only one of the packets. It synchronizes to the first packet that reaches it and the other packet is lost. Receiver contention errors have been studied and quantified in [74]. This work assumes that there is no receiver contention loss to simplify analysis. This work assumes that if receiver contention occurs, then all the packets destined to the receiver are received without error (provided they are not subject to interference error). To determine the overall throughput of the network, the effect of receiver contention errors must be added to the interference losses.

## **Chapter 3**

### **Interference and Interference Avoidance**

Interference between multiplexed optical CDMA codewords can cause errors in reception of packets. This chapter defines the conditions for a bit error and packet error due to interference. It shows that optical CDMA networks exhibit two properties: non-mutually destructive interference and state correlation. The properties distinguish the optical CDMA medium from other media. The properties can be exploited to design a media access control protocol, Interference Avoidance.

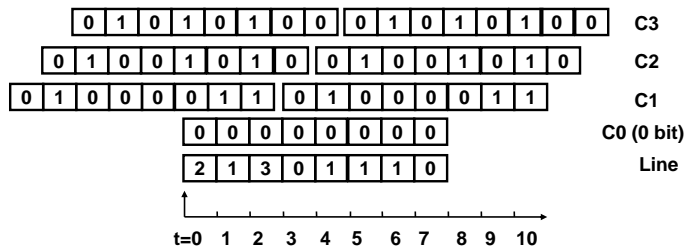
Section 3.1 defines interference and the conditions for a bit error and a packet error. Section 3.2 defines the state of the line. Section 3.3 uses the state definition to demonstrate that optical CDMA LANs possess the properties of non-mutually destructive interference and state correlation. Section 3.4 compares the properties to those in other media. Section 3.5 defines Interference Avoidance.

## 3.1 Interference

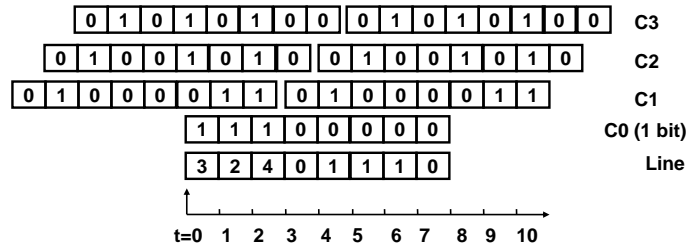
*Interference, multi-user interference or multi-access interference* in an optical CDMA network is the overlapping of chips when codewords are multiplexed on an optical fiber. Interference can result in packet errors and a consequent degradation in network throughput at high offered load. This section defines the conditions for a bit error and packet error due to interference<sup>1</sup>.

### 3.1.1 Assumptions

This work assumes that the codewords on the receive fiber are chip synchronized, *i.e.*, the chip boundaries are aligned in time. In reality that may not be true. Salehi [57] studied the effect of both chip synchronous and chip asynchronous transmission on a correlation receiver (both non-hard-limiting and hard-limiting). Using an argument based on varying the offset between two codewords continuously from zero to one chip time, it was shown that the chip synchronous case is a reasonable approximation for the Bit Error Rate (BER) of the system. This work uses that result. Analysis, simulations, and explanations assume that the codewords are chip synchronized.



(a)



(b)

Figure 3.1: A bit error. The figure shows four codewords on the optical fiber. In (a) codeword  $C0$  is OFF. A false positive error occurs at the receiver due to interference from  $C1$  and  $C2$  or  $C3$ .

### 3.1.2 Condition for a bit error

A *bit error* is an error in the reception of a codeword that is caused by interference. It occurs because a receiver falsely detects its codeword due to interference caused by other codewords on the line.

When several codewords are multiplexed on the line, a codeword on the line that has a chip overlap with the codeword being received is called an *interfering codeword* of the codeword being received. Figure 3.1 shows how a bit error occurs due to interfering codewords.

<sup>1</sup>Interference is not a measurable quantity nor is it a metric that measures network performance. Interference results in packet errors which results in a fall in network throughput. This work uses normalized network throughput (defined in Chapter 5) as a measure of network performance.

The figure is a snapshot of data bits sent by four nodes (codewords  $C0$ ,  $C1$ ,  $C2$ ,  $C3$ ) multiplexed on an optical fiber. The codewords are from an (8, 3, 3) codeset. A receiver tuned to codeword  $C0$  expects to receive ‘1 chips’ in the 1st, 2nd and 3rd chip slot. The time scale shown in the figure indicates the start of a bit of codeword  $C0$ . Figure 3.1(a) shows a case where a 0 bit is being received on  $C0$  and Figure 3.1(b) shows a case where a 1 bit is being received. The codewords  $C1$  and  $C2$  add up such that there is power in the 1st, 2nd and 3rd chips. The receiver tuned to  $C0$  detects a 1 bit whether a 0 bit is sent or a 1 bit is sent. So if a 0 bit were sent, it would be detected as a 1 bit. The packet would have an error and be lost. The errors are false positive errors, *i.e.*, a one bit is detected when a zero bit was transmitted. This work assumes that an error refers to a false positive error. The necessary and sufficient condition for a bit error is:

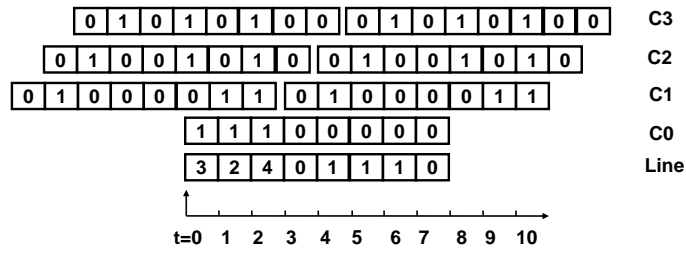
*A bit is received with error if every ‘1 chip’ of its codeword has at least one chip overlap with other codewords multiplexed on the line.*

*Conversely, a bit is received without error if at least one of the ‘1 chips’ of its codeword does not have a chip overlap with any other codeword multiplexed on the line.*

An Error Generating Codeset (EGC) of a codeword being received is a minimal subset<sup>1</sup> of the codewords on the line that have chip overlaps with all the ‘1 chips’ of the codeword being received. Its members are called *error generating codewords*. The *transmission delays* of the codewords are the delays (expressed in number of chips) of codewords from the start of the codeword being received (the reference time). The transmission delays of the error generating codewords are called *error generating delays*. Transmission delays are always

---

<sup>1</sup>A minimal subset for a given set and property is the subset with the minimum number of members that satisfies the property, *i.e.*, if any one member of the subset is removed, the subset no longer satisfies the property.



Receiving codeword: C0  
 Interfering codewords: {C1, C2, C3}  
 An error generating codeset: {C1, C2}  
 and its error generating delays: {3, 4}

Figure 3.2: Four codewords multiplexed on an optical fiber. The figure shows the receiving codeword, interfering codewords, the error generating codeset and the error generating delays.

specified as positive quantities, *i.e.*, time after the reference time in the interval  $(0, N]$ . Figure 3.2 shows the same codewords on the line. In this example, codeword  $C0$  is the receiving codeword. Codewords  $\{C1, C2, C3\}$  are interfering codewords. Codewords  $\{C1, C2\}$  form an error generating codeset. Codeword  $\{C3\}$  in the figure is an interfering codeword, but is not a member of this error generating codeset. However, it could be the member of another error generating codeset. With respect to the codeword  $C0$ , the transmission delay of codeword  $C1$  is 3 and the transmission delay of codeword  $C2$  is 4. The error generating delay set for the error generating codeset  $\{C1, C2\}$  is  $\{3, 4\}$ . The condition for a bit error may be restated as follows: *A bit error occurs in a codeword being received if there exists an error generating codeset with error generating delays.*

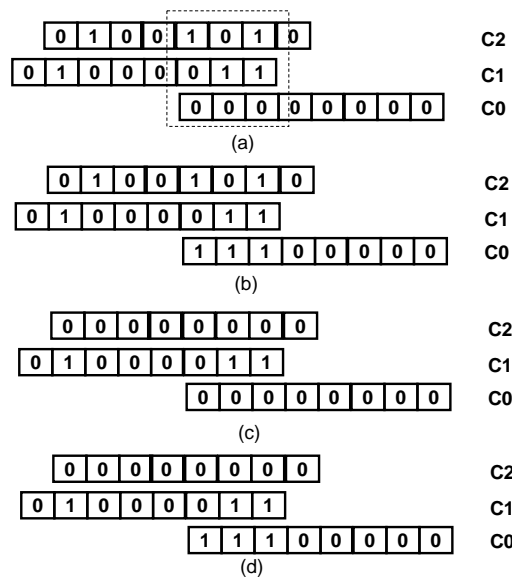


Figure 3.3: The different cases that can occur when codewords overlap. (a) shows the case when all members of the EGC are ON and the receiving codeword is OFF (causes a bit error). (b) shows the case when all members of the EGC are ON and the receiving codeword is ON (does not cause a bit error). (c) shows the case when one member of the EGC is OFF and the receiving codeword is OFF (does not cause a bit error). (d) shows the case when one member of the EGC is OFF and the receiving codeword is ON (does not cause a bit error).

The codeword being received may span two bits of an error generating codeword. For example, in Figure 3.1(a) two bits of codeword *C1* span the receiving codeword *C0*. From the figure, it can be seen that only the first bit needs to be ON for it to contribute to the false positive error. Similarly, only the first bit of codeword *C2* needs to be ON. However, that may not be the case in general. Four possible cases can occur when a receiving codeword has an error generating codeset on the line:

- The error generating codeset has all its codewords ON and the receiving codeword is OFF. This causes a false positive error.

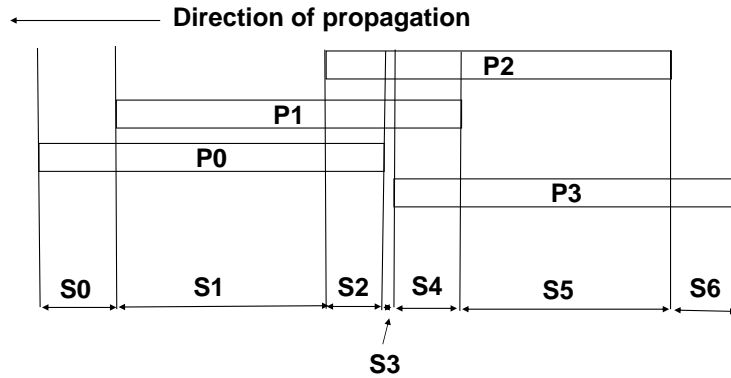


Figure 3.4: Four overlapping packets on an optical fiber. Each packet may be divided into sections at instants where a packet has arrived or departed.

- The error generating codeset has all codewords ON and the receiving codeword is ON.  
This does not cause an error.
- The error generating codeset has at least one codeword OFF and the receiving codeword is OFF. This does not cause an error
- The error generating codeset has at least one codeword OFF and receiving codeword is ON. This does not cause an error.

The cases are shown in Figure 3.3.

### 3.1.3 Condition for a packet error

A packet is received with an error if at least one of its codewords has been received with a bit error. This work assumes that if a packet is received with at least one error then the packet is lost. This loss is called an *interference error* packet loss.

Figure 3.4 shows packets on a receive fiber. The packet ( $P0$ ) is being received by a node. During the time when  $P0$  is on the line, there are arrivals and departures of other packets. The figure shows the overlapping of several other packets ( $P1, P2, P3$ ). The packet being received  $P0$  can be divided into *sections*. A section is defined as a portion of the received packet when a unique set of packets overlap with the packet being received. In Figure 3.4, packet  $P0$  has three sections:  $S0, S1$ , and  $S2$ . The new section may contain an error generating codeset for the codeword being received. The occurrence of an error in a packet being received depends on both the interfering codewords and delays in every section of the packet. This work assumes that if a packet has a section that has interfering codewords that form an error generating codeset, then the packet being received is lost. A section is generated when a packet arrives on the line or when a packet departs from the line.

## 3.2 State of the line

The *state of the line* (also called state of the network) at a point in time and space on a shared medium is a variable that can be used to predict the result (success or failure) of transmitting a packet at that point at that time. The prediction assumes that the state does not change during the transmission of the packet. The state at a point on a CSMA/CD based network

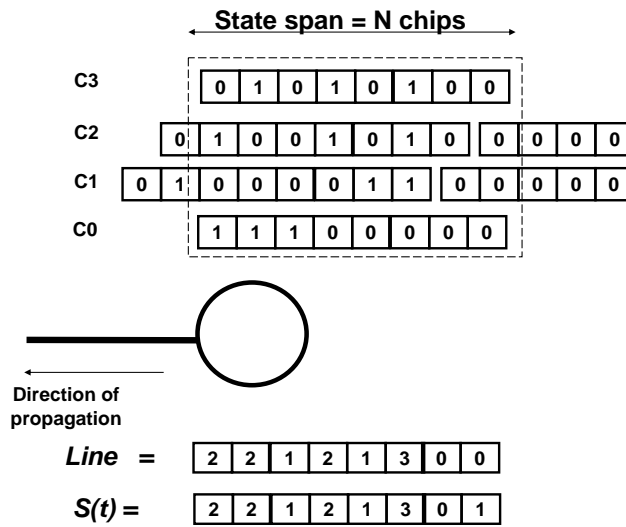


Figure 3.5: The state of the line at a point on a receiver fiber. The state of the line at time  $t$  at distance  $d$  is [2 2 1 2 2 3 0 0]. The second bit of codeword  $C1$  is OFF. To calculate the state,  $C0$  is assumed to be ON and the codewords are added.

such as 10 Mb/s Ethernet could be idle, busy or collision. The state can be used to predict the result of transmitting a packet at that point. If the state is busy, then the result of transmitting a packet is a failure. In the case of an optical CDMA LAN, the state is a variable that can be used to predict the result of transmitting a *codeword* at that point at that time. The following sections define state of an optical CDMA network. The conditions for a state transition are also discussed.

### 3.2.1 State of the line

The *state of the line* at a time  $t$ , at a distance  $d$  from the coupler on the receive fiber is a vector of length  $N$  whose elements are the chip magnitudes at distance  $d$  at times  $t, t + t_c, t + 2t_c, \dots, t + (N - 1)t_c$  from the coupler assuming that all codewords at  $d$  are transmitting 1 bits.

$$S(d, t) = [s_0 s_1 s_2 \dots s_{N-1}]$$

where  $s_n$  = chip magnitude at distance  $d$  at time  $t + nt_c$  ( $t_c$  is the chip time) assuming all codewords at  $d$  are transmitting 1 bits.

The *transmission delays* of the codewords with respect to the state are the delays (in chip times) of codewords from the start of the state (the reference time). The state is the sum of codewords each shifted by different transmission delays due to the different packet arrival times. In Figure 3.5, the second bit of codeword  $CI$  is OFF. To calculate the state, assume that it is ON and add the codewords. The state in Figure 3.5 is  $S(d, t) = [2 \ 2 \ 1 \ 2 \ 1 \ 3 \ 0 \ 1]$ . The state at the same distance from the coupler on different receive fibers is the same because all receive fibers carry the same signal. This is a consequence of the low noise, guided medium characteristic of the physical medium.

### 3.2.2 State observations

The signal that reaches a node on the receive fiber is a multilevel optical signal. A node receive this signal and can determine the chip magnitude at each chip time. Chapter 7 discusses the hardware design for a module that can perform this function. Using these chip observations, it can construct an observation of the state.

A *state observation* at time  $t$ , and distance  $d$  from the coupler on the receive fiber is a vector of length  $N$  whose elements are the chip magnitudes at distance  $d$  at times  $t, t + t_c, t + 2t_c, \dots, t + (N - 1)t_c$  from the coupler.

$$S_{observation}(d, t) = [s_0 s_1 s_2 \dots s_{N-1}]$$

where  $s_n$  = chip magnitude at distance  $d$  at time  $t + nt_c$  ( $t_c$  is the chip time).

### 3.2.3 State transitions

The state consists of  $N$  chips. It follows that state at a point is measured over an interval of  $Nt_c$  where  $t_c$  is the time to transmit a chip. Therefore, state at a point is measured at intervals of  $Nt_c$  (or integral multiples of  $Nt_c$ ) to avoid overlapping two consecutive state measurements. Similarly, state at a time is measured at points that are separated by distance of  $Nt_d$  (or integral multiples of  $Nt_d$ ). The times when state is measured at a point are called the *state measurement instants*. The points where state is measured are called the *state measurement points*.

A *state transition* is defined as the change of state at a point on the line from one state measurement instant to the next. A state transition occurs due to a packet arrival or departure. State transitions are significant because a state transition can trigger an interference error in a packet on the line. The state measurement instants between which the state transition occurs depend on the codeword that arrives/departs and its arrival/departure time with respect to the state measurement instants.

State observations at a point change between state measurement instants due to ON-OFF keying of codewords. However, the state does not change due to ON-OFF keying.

### **3.3 Properties of an optical CDMA network**

An optical CDMA network has two properties that distinguish it from other physical layers: the non-mutually destructive nature of interference and the correlation of state. The non-mutually destructive nature of interference makes transmission scheduling feasible. The correlation of state makes state estimation feasible.

#### **3.3.1 Non mutually destructive nature of interference**

The *non-mutually destructive* nature of interference is the property by which interference can cause errors in a subset of the codewords on the line. The subset can be controlled by varying the transmission delays of the codewords.

The non-mutually destructive nature can be illustrated by considering a packet arrival at the output of the coupler. A packet arrival triggers a state transition at the coupler. The state transition may trigger an interference error in the arriving packet or in any packet that overlaps



Figure 3.6: The four possible packet transmission events that are a consequence of the non-mutually destructive nature of interference.

with the arriving packet. As a result, one of four possible events (*transmission events*) could occur:

- The arriving packet may not have an interference error, (*i.e.*, it preserves itself), but may cause interference errors in one or more other packets on the line (it destroys others). This case is called *Preserves self/Destroys others (PS/DO)*.
- The arriving packet may preserve itself and also preserve all other packets on the line. This case is called *Preserves self/Preserves others (PS/PO)*.
- The arriving packet may destroy itself but preserve all other packets on the line. This case is called *Destroys self/Preserves others (DS/PO)*.

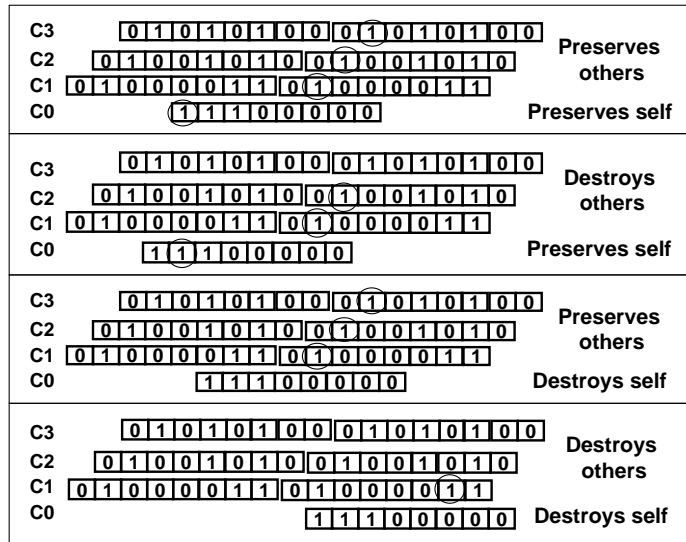


Figure 3.7: Examples of the four packet transmission events. In each example, the different transmission time of codeword  $C0$  has caused a different transmission event. The circled chips mark the ‘1 chips’ that do not have overlaps and the corresponding codewords are preserved.

- The arriving packet may destroy itself and destroy one or more other packets on the line. This case is called *Destroys self/Destroys others (DS/DO)*.

The condition for a bit error can be applied to every codeword on the line to determine which event occurs. Figure 3.6 shows the four different transmission events. Figure 3.7 shows examples of all four events. The codewords are the same as in Figure 3.2.  $C0$  is the arriving codeword and  $C1$ ,  $C2$ ,  $C3$  are codewords on the line. The offset of codeword  $C0$  is different in each case. The codewords with circled ‘1 chips’ are the codewords that are preserved (the circled ‘1 chips’ do not have overlaps and hence the codewords satisfy the condition for correct reception). All other codewords are destroyed.

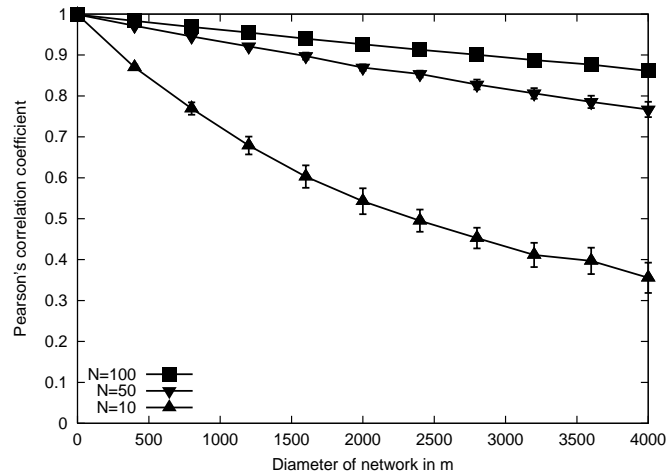


Figure 3.8: The correlation between the state at the estimation point at estimation time and the merging point at merging time for an optical CDMA network as the diameter of the network increases. The network has 100 nodes distributed uniformly over the length of the network. The network uses Aloha-CDMA. Packet arrivals are Poisson with offered load 1 and packet lengths are exponentially distributed with an average length of 1000 bytes.

The codewords that are destroyed by interference are a subset of the codewords on the line. The events do not specify *how many* other packets are destroyed. When a packet arrives, the state transition could result in the destruction of up to  $w$  other codewords. Each ‘1 chip’ of the arriving codeword could contribute to an interference error in one of the other packets on the line. If the arriving codeword preserves itself, it can destroy up to  $w - 1$  other codewords.

### 3.3.2 Correlation of state

*Correlation of state* is a property by which state at a state measurement point and instant is correlated to the state at other state measurement points and instants.

The correlation between two states is measured using the Pearson correlation coefficient [2]. The Pearson correlation coefficient between two vector samples  $X$  and  $Y$  is defined below:

$$r = \frac{\sum_{i=1}^N (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{X})^2 \sum_{i=1}^N (y_i - \bar{Y})^2}}$$

where  $x_i$  are the elements of a vector  $X$ , and  $\bar{X}$  is the mean of the elements. It is a number between -1 and 1 and indicates the extent of a linear relationship between two measured quantities. A value of 0.8 and above generally indicates a strong degree of linear correlation.

A simple experiment shows that an optical CDMA network exhibits correlation. Consider an Aloha-CDMA optical CDMA network, *i.e.*, an optical CDMA network without any media access control. Packet arrivals are Poisson and the offered load is 1. The packet size is exponentially distributed with average size 1000 bytes. The chipping rate is 10 Gc/s. The offered load is 1. The number of nodes on the network is 100 and the nodes are at uniformly distributed distances from the coupler. The distance between the nodes and the coupler are integral multiples of the length of the state on the line. Figure 3.8 shows the average correlation between the state at a node and the state at the output of the coupler as the diameter of the network is increased. The graphs are plotted for different codeset lengths. The graph shows that as the diameter of the network increases, the correlation between the state at the node and at the coupler decreases. As the codeset length increases the correlation increases. For diameters of up to 4000m, a codeset length of  $N = 100$ , shows a reasonable amount of correlation. This indicates that under certain conditions the state of the optical CDMA network exhibits correlation.

Several factors affect the correlation including the diameter of the network, packet lengths, chipping rate and the codeset length. Consider the states  $(S(d_1, t))$  and  $(S(d_2, t))$  measured at two different points  $(d_1, d_2)$  from the coupler) at the same time  $t$ . There are two reasons why the correlation increases as  $N$  increases and  $w$  decreases.

- The probability of a chip remaining the same increases: As  $N$  increases, the physical length of the packet on the line increases. To illustrate this, consider the  $k^{th}$  chips  $c_{k1}, c_{k2}$  in the two states  $S_0$  and  $S_1$ . If the chip is a 1, then the probability that it remains a 1 increases as  $N$  increases. This is because the length of packets are longer and therefore it is likely that the same codeword is present in both  $S_1$  and  $S_2$ .
- The probability of a chip changing decreases: A chip can change only if a packet arrival or departure occurs between the two state measurement points. As  $N$  increases, the probability that a change occurs in any particular chip of the state reduces, *e.g.*, if  $N = 10$ , and a change occurs in the state, the probability of a change in any particular chip is  $1/10$ . However if  $N = 100$ , this probability reduces to  $1/100$ .

The consequence of both of these is that there is higher correlation between the states.

### **3.4 Comparison of state and its properties to other physical layers**

This section compares the properties of an optical CDMA LAN with the properties of two other physical layers that use contention media access control: the single channel, shared medium, wired network (such as CSMA/CD based 10 Mb/s Ethernet) and the single channel, shared medium, wireless network (such as CSMA/CA based 802.11 wireless networks). This

section focuses on the differences in three areas: the state definition, the destructive/non-destructive nature of interference and the correlation/non-correlation of state. Tables 3.1 and 3.2 summarize these differences.

### **3.4.1 Wired networks**

Early work on state in single channel shared medium wired networks considered state at a point as a scalar that could take on one of three values: *Idle*, *Busy*, *Collision*. *Idle* is the state when there is no packet transmission at that point. *Busy* is the state when there is exactly one packet transmission at that point. *Collision* is the state when more than one packet is being transmitted at that point.

#### **3.4.1.1 Interference**

Collisions on wired networks are mutually destructive, *i.e.*, a collision destroys all packets involved in the collision. In terms of transmission events, the only possible events are *Preserves self/Preserves others* and *Destroys self/Destroys others*. This model has been used to analyze shared medium, wired networks with Aloha [32] and carrier sensing [68] media access control.

#### **3.4.1.2 Correlation of state**

The state in single channel, shared medium, wired networks exhibits the property of correlation. [32] showed that the mechanisms of Carrier Sensing and Collision Detection depend on this property (though the term correlation was not used in that work). It showed that the

performance of carrier sensing depended on the *normalized propagation delay*  $a$ , i.e., the ratio of the propagation delay to the packet transmission time. As  $a$  increases, the correlation of the state decreases and the efficiency of carrier sensing as a predictor of the state of the network decreased.  $a$  depends on three parameters: the length of the packets, the diameters of the network and the transmission rate.

$$a = t_p/t_x = (d/c)/(L/B) = dB/cL$$

where  $c$  is the speed of light,  $d$  is the diameter of the network,  $L$  is the packet length and  $B$  is the transmission rate. For certain values of  $a$  ( $0.01 < a < 0.2$ ) it was shown that carrier sensing and collision detection can significantly improve throughput in single channel shared medium networks. For a network of speed  $100Mb/s$ , packet length 1000 bytes, diameter 4000m the value of  $a = 0.1$ . However for a high speed network, such as a  $10Gb/s$  network, the value of  $a = 10$ . In comparison, for an optical CDMA network the normalized propagation delay reduces by a factor of  $N$ , because the codeword length increases the length of the packet on the line. For a network of diameter 4000m, chipping rate 10 Gc/s, packet length 1000 bytes and codeset length  $N = 100$ , the value of  $a = 0.25$ . This low value of  $a$ , is responsible for the high correlation of state in an optical CDMA network.

### **3.4.2 Wireless networks**

In a single channel, shared medium, wireless network the state depends on the channel conditions. The wireless medium is much noisier than the wired medium due to fading (slow and

fast), multipath effects, and frequency selective fading [55]. These conditions vary greatly depending on the location of a node within a network. Early work in wireless MAC protocols [6] uses a scalar state that can take on three values: idle, busy or collision. Recent work in the area of Channel State Estimation [56] uses a scalar value with a limited number of levels to describe the channel state.

#### **3.4.2.1 Interference**

Initial work on contention protocols such as packet radio and 802.11 considered packet collisions to be mutually destructive. The effect of carrier sensing in this model has been studied using both theoretical [6] and experimental [22] techniques. The model has been extended to consider the effect of *capture* [72]. Capture is the ability of a receiver to receive a packet despite interference from another transmitter (even if it overlaps with other packets). It is not just a property of the medium but also depends on the implementation of the radio, modulation scheme, and decoding scheme. In a network that exhibits capture, destruction may be controlled by varying the power levels of the transmitter [65]. A node can preserve itself by raising its transmission power, while possibly destroying other transmissions. A multi-packet reception model [69] has been proposed for modeling the reception of a subset of packets. The model has been used for cross layer design of signal processing and MAC protocols to improve throughput.

Although capture is similar to the non-mutually destructive optical CDMA interference, the difference is that it is controlled by power rather than delay. Moreover, the subset of packets being destroyed cannot be controlled.

Property	Single channel wired medium	Single channel wireless medium	Optical CDMA
State representation	Scalar	Scalar	Vector
State values	Idle, Busy Collision	Good/Bad	Vector of length $N$ Components are non-negative integers
State transitions	Packet arrival, departure	Attenuation, noise	Packet arrival, departure ON-OFF keying

Table 3.1: Comparison of the state of the network in three shared medium networks: Single channel wired networks, single channel wireless networks, and optical CDMA networks.

### 3.4.2.2 Correlation of state

The wireless medium exhibits low correlation of state. In a wireless medium, transmissions follow different paths from transmitters to a receiver. Every node on the network sees a different state because the propagation distances between different nodes are different and transmissions follow a straight line propagation path. In addition, multipath and mobility of nodes can make propagation delays dynamic. The limited range of transmitters can cause hidden and exposed terminals [6]. As a result, the wireless medium has low correlation of state. These four conditions mean that the state of the network appears different at different points on the network and correlation is low. Studies of carrier sensing in wireless networks [6] confirm this.

Property	Single channel wired medium	Single channel wireless medium	Optical CDMA
Correlation of state	Yes Depends on: Packet length Network diameter Transmission rate	No Depends on: Noise Attenuation Mobility	Yes Depends on: Packet length Network diameter Chipping rate Codeset length
Nature of interference	Mutually destructive	Non-mutually destructive (Capture)	Non-mutually destructive (PS/PO, PS/DO, DS/DO, DS/PO)

Table 3.2: Comparison of the properties of the state of the network in three shared medium networks: Single channel wired networks, single channel wireless networks, and optical CDMA networks.

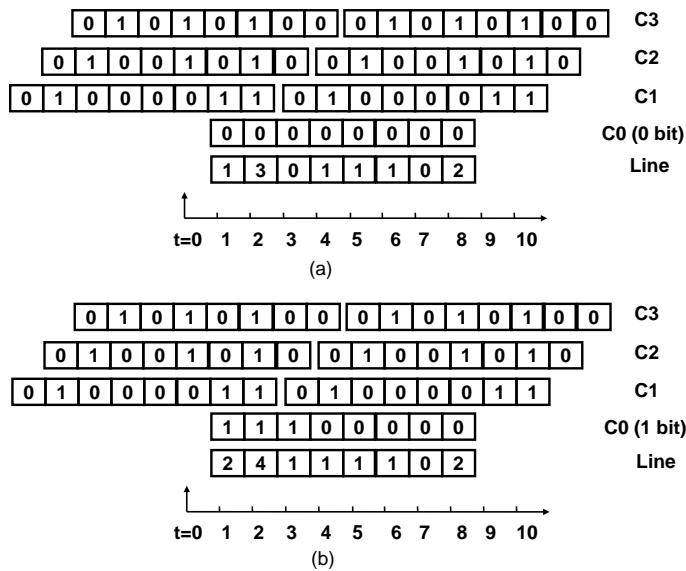


Figure 3.9: Interference Avoidance prevents occurrence of a bit error. No error is caused when the same codewords in Figure 3.1 are sent with a different set of transmission delays. The figure shows the same codewords with codeword C0 delayed by 1 chip time.

### 3.5 Interference Avoidance

*Interference Avoidance is a distributed, contention based Media Access Control (MAC) protocol for broadcast shared medium optical Code Division Multiple Access (CDMA) Local Area Networks (LANs). It improves network throughput at high offered load by estimating the state of the line and scheduling transmissions to reduce packet loss due to interference errors.*

From Sections 3.1 and 3.3:

- A packet is received correctly if at least one ‘1 chip’ of its codeword does not overlap with any other ‘1 chip’ on the line over the entire packet length (Condition for a packet error).
- Varying the transmission time of a packet can cause or prevent errors in any codeword on the line (The non-mutually destructive nature of interference).
- The state exhibits correlation, *i.e.*, the state at points on the fiber at the estimation time is correlated to the state at the output of the coupler (The state correlation).

Interference Avoidance exploits these properties. It consists of two procedures: *State estimation* and *Transmission scheduling*. State estimation is the process by which a node estimates the state of the line. Transmission scheduling is the process by which a node, given an estimate of the line state and the codeword for encoding, calculates when to transmit the codeword such that packet loss due to interference errors is reduced. Interference Avoidance is a distributed, contention based protocol, *i.e.*, nodes perform state estimation and transmission scheduling independent of each other. Each node on the network contends for access

to the medium using the Interference Avoidance protocol. There is no centralized scheduler arbitrating access to the network.

### 3.5.1 Transmission scheduling

Transmission scheduling is defined as the process by which a node, given an estimate of the state and the codeword to transmit, calculates when to transmit the codeword such that packet loss due to interference errors is reduced. The objective of transmission scheduling is to schedule the transmission time of codewords to reduce false positive errors. Figures 3.1 and 3.9 show this effect. In the second figure, codeword  $C0$  has been delayed by one chip time. The first case has an interference error and the second case does not. If Interference Avoidance was used, before transmitting  $C0$  the node would estimate the state and see that it was  $[ 2 1 3 0 1 1 1 0 ]$ . By applying the condition for correct reception of its own codeword, the transmission scheduling algorithm would calculate that by delaying the transmission by 1 chip time, an interference error in its own packet could be eliminated. As a result, the node would transmit as indicated in Figure 3.9. By delaying the transmission the event has been changed from a *Destroys self/Destroys others* to a *Preserve self/Preserve others* event. There are several ways to schedule a transmission such that any one of the four transmission events could occur. Chapter 5 formulates the transmission scheduling problem and discusses transmission scheduling algorithms and their performance.

### 3.5.2 State estimation

State estimation is the process by which a node estimates the state of the line at the point/time where packets merge. The objective of state estimation is to use state samples at the estimation point and estimation time to calculate a value as close as possible to the state at the output of the coupler. Chapter 6 formulates the state estimation problem and discusses state estimation algorithms and their performance in detail.

The performance of state estimation algorithms is compared to *Perfect state estimation*. *Perfect state estimation* is an ideal, unrealizable state estimation with the following properties:

- The nodes can estimate the state at any point on the line and time without error.
- All nodes see the same state at the same time.
- State estimation is instantaneous.
- There is no delay between state estimation and transmission scheduling.

Perfect state estimation is not possible in a real network. However, it simplifies the study of transmission scheduling. The study of transmission scheduling in Chapter 5 assumes a perfect state estimation process. When state is estimated using a state estimation algorithm that collects state observations it is called *Realistic state estimation*.

## Chapter 4

### Network architecture

This chapter discusses the architecture of an optical CDMA Local Area Network. The architectural designs discussed in this chapter neither are a limitation nor are enforced by Interference Avoidance. Interference Avoidance may also be used with alternate designs. This section discusses these designs to provide the context within which Interference Avoidance may be discussed.

The base features of the optical CDMA Local Area Network proposed in this work are:

- Shared medium with multiple access: The network consists of several nodes connected by optical fiber to a passive optical coupler. The fiber and the coupler form a shared medium. The multiple access mechanism is optical CDMA using one or more wavelengths. Its typical use is in a shared medium, optical access network such as a Fiber To The Home (FTTH) Passive Optical Network (PON).
- Packet switching: The unit of data transmission is a packet. Nodes on the network do not maintain connection state (for any protocol layer below the transport layer).

- Contention media access control: The media access control is Interference Avoidance. It is a distributed media access control protocol based on contention [5].

This chapter specifically looks at the following issues:

- Network topology: The physical layout of the network.
- Physical layer design: Codeword assignment.
- Link layer design: Addressing, address to codeword mapping, frame format and frame synchronization.
- Node design: How the physical, link and MAC layer are integrated in a node.

## **4.1 Network topology**

The topology of the network is a star topology. The optical CDMA LAN is constructed using single mode fiber and a passive optic coupler. Single mode fiber is used for most FTTH deployments today [31] because of low attenuation and dispersion. The passive optic coupler [28] links all the nodes on the network. The passive star coupler is an integrated star coupler based on free space optics. The coupler has low loss ( $< 3\text{dB}$ ) and can have up to 128 ports. The network has amplifiers placed along the fiber at points where signal strength degrades below receiver sensitivity. The placement of amplifiers is significant because of their contribution to noise. Erbium Doped Fiber Amplifiers (EDFAs) [28] are the most suitable amplifiers for this purpose. The effect of EDFAs on signal propagation in a long haul fiber carrying optical CDMA has been studied [42]. The results indicate that for long networks

noise due to EDFA can affect the signal to interference ratio. They also show that the effects can be reduced for by using dispersion compensation and EDFA gain equalization. Based on this conclusion, this work assumes that signal distortions can be compensated for in an FTTH network.

Chapter 6 shows that the performance of Interference Avoidance is dependent on the distance of nodes from the point where transmissions are multiplexed, *i.e.*, the coupler. Therefore the performance analysis presented here this applies only to a star topology <sup>1</sup>. This work assumes that the maximum diameter of the network (the distance between the two furthest nodes) is 4000m. This is a limitation of Interference Avoidance. It is discussed in Chapter 6.

The choice of a star topology and a passive optic coupler was influenced by the architecture of Fiber to the Home networks. Most FTTH networks are Passive Optical Networks, which are based on a star topology. The head end of the network and the residential nodes are connected by fiber to the coupler. Typical FTTH networks generally have much larger diameters than 2000m. The head end is usually 10-15 km away from the end users. However, most of the home users are within close proximity of each other (1000 users within 5 square km). The end users contend for access to the upstream link to the head end. Optical CDMA with Interference Avoidance can be used as the media access technology in such a topology. End users can contend for the upstream link using Interference Avoidance. The downstream link between the head end and the end users can use other access technology (it does not need multiple access). Figure 4.1 shows a diagram of the future FTTH network.

---

<sup>1</sup>Some topologies such as a bus topology can be reduced to a star topology. However there are several topologies for which such a mapping may not exist.

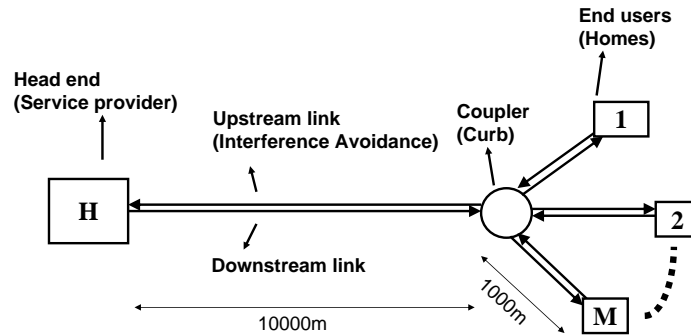


Figure 4.1: A typical Fiber to the Home (FTTH) network topology. It shows the upstream link where Interference Avoidance may be used as a media access mechanism.

## 4.2 Node architecture

Interference Avoidance is a contention media access control (MAC) protocol. Each node on the network contends for access to the medium using the Interference Avoidance protocol. Figure 4.2 shows a block diagram of an Interference Avoidance Network Interface Card. It consists of an optical CDMA transmitter, optical CDMA receiver, state estimation module, and transmission scheduling module.

The state estimation module performs two functions: receiving and estimation. It receives the multilevel optical signal on the receive fiber and collects observations of the state of the line. It uses a series of state observations to calculate a state estimate. The state estimation algorithm is always active. It is run continuously in a loop, collecting state observations

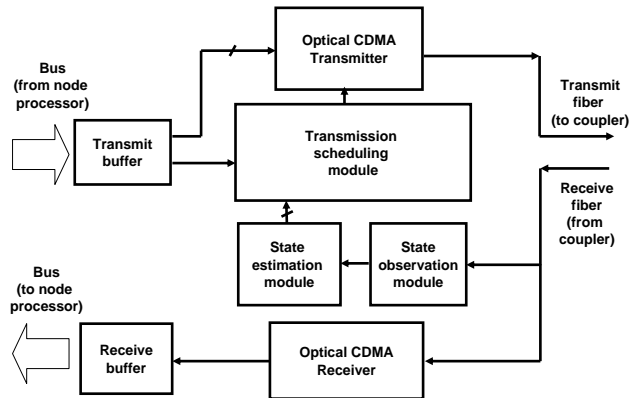


Figure 4.2: Block diagram of an Interference Avoidance Network Interface Card.

and calculating a state estimate. The state estimation module consists of both optical and electronic components.

The transmission scheduling module uses the state estimate and the codeword to be used for encoding to calculate an appropriate delay to transmit the packet. Transmission scheduling is invoked on arrival of a packet from the node processor. It reads the current state estimate and the codeword for encoding and calculates the delay. The optical CDMA transmitter encodes the data and begins transmission after the delay.

Chapter 5 defines the transmission scheduling algorithms. Chapter 6 defines different state estimation algorithms. Chapter 7 describes the hardware implementation of the individual blocks of this NIC.

## 4.3 Physical layer design

This section deals with the issue of codeword assignment, *i.e.*, how a node chooses a codeword to transmit or receive on.

### 4.3.1 Codeword assignment

Codeword assignment is the scheme by which nodes decide which codewords to receive on and which to transmit on. Every node on the network is equipped with at least one transmitter and one receiver.

This work assumes a Tunable Transmitter-Fixed Receiver (TT-FR) architecture. This means that a node receives on a fixed codeword. A node that wants to transmit tunes its transmitter to the receiver's codeword and transmits. Networks based on the (TT-FR) architecture as well as other architectures (TT-TR, FR-TT, FT-FR) have been studied before in the context of single hop WDM networks [48]. Each architecture has its advantages and limitations. The TT-FR architecture is chosen for the optical CDMA LAN because it requires only one transmitter/receiver pair per node and it eliminates the need for pre-transmission coordination. The TT-FR architecture works well when every node has a unique codeword to receive on. However, if the number of nodes is greater than the number of codewords in a codeset, then codewords are shared.

### 4.3.2 Codeset design

Chapter 2 shows that for a given set of optical CDMA codeset parameters  $(N, w, \kappa)$ , the number of codewords in the codeset is constrained to a value below the Johnson bound.

Any one of the parameters  $N$ ,  $w$  or  $\kappa$  can be varied to increase the number of codewords in a codeset. However Chapter 5 shows that varying  $N$  affects the data rate of the individual nodes and varying  $w$  affects the aggregate network throughput. Therefore to increase the number of codewords in the codeset,  $\kappa$  is the only parameter that can be varied. Chapter 5 shows that varying  $\kappa$  does not affect the throughput.

If  $\kappa = 1$ , then the number of codewords in the codeset is low. If the number of nodes on the network is greater than the number of codewords, then the network shares codewords among nodes. This is called *codeword sharing* or *codeword reuse*. Multiple nodes tune their receivers to the same codeword and receive their packets on the same codeword. They demultiplex their packets based on a unique higher layer address such as a link layer address. The number of codewords generated can be increased by increasing  $\kappa$  and each node on the network can be allocated a unique codeword to receive on. If  $\kappa = w$ , a large number of codewords may be generated. However, some codewords may be rotations of other codewords.

The codesets in this work are generated using  $\kappa = w$  with a greedy algorithm [10]. Codewords that are rotations of each other were removed from the codeset. To understand the effect of  $\kappa$  on performance, some experiments use codesets with  $\kappa < w$ , generated by other methods such as prime sequences, BCH [10]. In such cases, the number of codewords may be less than the number of nodes on the network. Such experiments use codeword sharing. For a given  $N$ ,  $w$  and  $\kappa$ , the different codeset methods generate a different codeset size. The performance at high offered load is not affected by the choice of codeset design method. This will be demonstrated in the next chapter by varying all three codeset parameters:  $N$ ,  $w$  and  $\kappa$  and the codeset size.

The mechanism that a node uses to choose the codeword to receive on is out of the scope of this work. Alternatives for the allocation mechanism include permanent allocation at manufacture, a centralized server based allocation (such as DHCP) or selection from a codeset based on the hash function of a higher layer address.

## **4.4 Link layer design**

### **4.4.1 Addressing**

Every node has a unique *node address* that is unrelated to the codewords in the codeset. The node address is a higher layer (link or network) address. It is an identifier that uniquely identifies the node on the LAN. A network layer address (IP address) or a link layer address (Ethernet address) may be used.

### **4.4.2 Address to codeword mapping**

It is assumed that there is a mapping function from the destination node address to the codeword that the destination node is receiving on. The mapping function allows any node on the network to determine the codeword that any other node is receiving on. When a node wants to transmit to another node, it uses the mapping function to determine the codeword to transmit on.

The mechanism for mapping is out of the scope of this work. Alternatives for the mapping function include a centralized server (a DNS like mechanism), a broadcast codeword (an ARP

like mechanism), or a hash function that maps the higher layer address to a codeword. Traffic due to the mapping function is not included in analysis of the system.

#### **4.4.3 Frame synchronization**

Frame synchronization is the process by which a receiving node determines the exact time when data transmission of the packet has begun. The objective of synchronization is to allow the receiver to determine where the first chip of the header of the frame starts.

Several frame synchronization algorithms for optical CDMA have been proposed [29] [30]. The serial search synchronization mechanism uses a synchronization sequence in the header of the packet. The receiving node continuously listens for the synchronization sequence. When a receiver detects that the frame synchronization sequence has started, the receiver enters a synchronization phase where it receives the remainder of the frame synchronization sequence (usually a series of 1 data bits) and attempts to acquire synchronization with the start of the packet. This is called the synchronization phase. When it detects the sequence, it knows that data will arrive after the sequence. After acquiring synchronization, it starts detection of the bits of the packet.

This work assumes that the frame synchronization is instantaneous, *i.e.*, the receiver knows where the frame starts.

#### **4.4.4 Frame format**

The frame has the following fields:

- Synchronization sequence: The frame synchronization sequence.

Synchronization sequence	Length of frame	Payload	Checksum
--------------------------	-----------------	---------	----------

Figure 4.3: The frame format for Interference Avoidance frames. The synchronization sequence is used for the frame synchronization operation, the length field is used to determine the size of the payload, and the checksum field is used for error detection.

- Length: The length of the link layer payload.
- Link layer payload
- Error detection: A checksum or other error detection mechanism.

Details of the synchronization sequence, synchronization algorithm, error detection and correction mechanism, and link layer error control protocols are out of the scope of this work.

## 4.5 Upper layer protocol design issues

The main concern about optical CDMA is its high bit error rate, which causes a high packet error rate. Interference Avoidance prevents throughput collapse at loads up to 100%. However, the packet error rates can still be fairly high. At such high error rates, transport protocols such as TCP have been shown to provide very low throughput. Moreover, at offered loads higher than 100%, Interference Avoidance shows a degradation in throughput. Therefore, some mechanism is required to reduce packet error at high load and beyond. There are four possible methods that can be used. The interaction of these mechanisms with Interference Avoidance is an open area of research.

- Forward Error Correction: Muckenheim [47] studied the performance of block FEC codes with optical CDMA and suggested the use of suitable codes. They showed that error codes can reduce error rates at high load.
- Link layer Automatic Repeat Request (ARQ): A link layer ARQ may be used to initiate retransmission of packets in which errors are detected.
- Randomized delay before transmission: Nodes can delay the execution of the scheduling algorithm by a random interval after the packet arrival. In single channel networks persistence algorithms (non,  $p$  and 1 persistent) are used to generate a random delay before transmission. This can improve throughput by reducing collisions [68].
- Interference detection: If a node aborts transmission of a packet when it sees that it has been destroyed due to interference, then packet throughput may improve. However,

this is at the cost of additional optical components needed to perform the error detection during transmission.

## Chapter 5

### Transmission scheduling

*Transmission scheduling* is the process by which a node, given an estimate of the state of the line and a codeword to transmit, calculates *when* to transmit the codeword such that packet losses due to interference errors are reduced. This chapter formulates the transmission scheduling problem and discusses strategies for transmission scheduling. The chapter assumes that all nodes can perform perfect state estimation and the network has a propagation delay of zero. It proposes three algorithms for transmission scheduling. The algorithms stabilize throughput at around 30% of the maximum throughput at high offered load. Under similar conditions, non-scheduled systems have almost zero throughput.

Section 5.1 formulates the transmission scheduling problem. Section 5.2 discusses strategies for transmission scheduling. Section 5.3 defines and explains the algorithms. Section 5.4 presents a performance study of transmission scheduling. It analyzes performance using mathematical modeling and simulation. Section 5.5 shows that the performance of the algorithms is within 15% of the performance of an optimal codeword scheduling algorithm.

Section 5.6 presents a sensitivity study that shows how transmission scheduling algorithms perform over a wide range of codeset parameters and traffic conditions.

## 5.1 The transmission scheduling problem

The transmission scheduling problem is:

*Given a state estimate  $S_e$  and a codeword to be used for encoding  $c_{tx}$ , determine a transmission delay  $t_d$  in chips such that the aggregate network throughput  $Th_{norm}$  is maximized.*

The input to the transmission scheduling algorithm is the state estimate and the codeword to be used for encoding<sup>1</sup>. If the packet can be scheduled, the algorithm returns a transmission delay  $t_d$  ( $0 \leq t_d < N$ ) and the node transmits its codeword after  $t_d$  chips<sup>2</sup>. If not the packet is returned to a higher layer for a retransmission attempt.

The transmission scheduling problem is an online scheduling problem. Nodes on the network schedule their packet transmissions taking local decisions with the objective of maximizing the global aggregate network throughput.

## 5.2 Transmission scheduling strategies

A *transmission scheduling strategy* is a subset of the transmission events that a node tries to achieve when it schedules its transmissions. Chapter 3 defined the four possible transmission events that could occur when a packet is transmitted (PS/PO, PS/DO, DS/PO, DS/DO).

---

<sup>1</sup>This chapter assumes *Perfect state estimation*, as defined in Chapter 3. Chapter 6 discusses the state estimation problem.

<sup>2</sup>The transmission delay  $t_d$  is measured from the first chip of the estimated state. Chapter 6 discusses the effect of timing differences between the state estimate and the state of the line.

Transmission scheduling controls which of the four events occurs when a packet is transmitted.

A *transmission scheduling algorithm* is an implementation of a strategy, *e.g.*, a node's strategy may be to transmit only if the resulting event is *Preserves self/Preserves others*. There may be several possible algorithms that achieve this goal.

The objective of transmission scheduling is maximization of the network throughput. The throughput is maximized when interference errors are minimized. Therefore, the transmission scheduling strategy should prevent transmissions where a node *Destroys self* or *Destroys others*. In other words, a node must schedule such that it *Preserves self/Preserves others*. Achieving this in a distributed algorithm is not straightforward. Given the state and a codeword and its transmission delay, it is possible to apply the condition for correct reception to determine whether it *Preserves self*. However, as the next three sections demonstrate, it is not possible to determine whether the transmission will destroy any codewords already on the line or how many codewords it will destroy. Therefore, it is not possible to apply the condition for correct reception and determine whether it *Preserves others*. The following sections discuss *selfish*, *cooperative*, and *pseudo-cooperative* strategies in detail.

### **5.2.1 Selfish strategies**

An algorithm follows a *selfish strategy* if the node schedules its packet transmission such that the resulting event is either *Preserves self/Preserves others* or *Preserves self/Destroys others*. A selfish strategy guarantees that the packet that the node transmits is preserved *if the state does not change after the packet is transmitted*. It can be implemented using the

condition for correction reception to choose a delay that guarantees that the line state does not cause an error in the codeword being transmitted. The delay should be chosen such that the codeword being transmitted has at least one '1 chip' that does not have a chip overlap. Note that this strategy does not guarantee that packets will be received correctly. Errors in state estimation or subsequent scheduling by other nodes may cause an error in the packet. The selfish strategy defines a class of algorithms that are significant because there is no incentive for a node to attempt non-selfish scheduling. If an algorithm is non-selfish, a node schedules a transmission even if it knows its packet may be lost due to interference. Therefore, there is no motivation for a non-selfish algorithm.

### **5.2.2 Cooperative strategies**

An algorithm follows a *cooperative strategy* if the node schedules its transmission such that the resulting event is either *Preserves self/Preserves others* or *Destroys self/Preserves others*. A cooperative strategy guarantees that the packet that the node transmits does not destroy other packets multiplexed on the line if the state does not change after the packet is transmitted.

It is not easy to design a cooperative strategy. The previous chapter discussed the condition for a packet error. A decision on whether all the codewords are preserved can be made only if all the codewords on the line are known. However, a node only knows the state of the

line and not the individual codewords multiplexed on the line. It is not possible to decompose the line state into its constituent codewords in general for all codesets.<sup>1</sup>

Another way of implementing a cooperative strategy is to schedule transmissions such that the number of chip overlaps in the resulting state is below the weight of the codeset. This guarantees that no codeword on the line can be destroyed (the condition for error is not satisfied). However, this strategy limits the number of codewords multiplexed on the line to a very low number. Section 5.6 shows that this strategy results in low network throughput. The difficulty of implementing a cooperative strategy is the motivation for the pseudo-cooperative strategy, discussed in the next section<sup>2</sup>.

### 5.2.3 Pseudo-cooperative strategies

An algorithm follows a *pseudo-cooperative* strategy if it schedules transmission such that it increases the probability that the resulting event is either *Preserves self/Preserves others* or *Destroys self/Preserves others*. It is a best effort strategy that reduces the probability of destroying other codewords on the line. A node speculatively schedules its transmission because of insufficient information about the codewords on the line.

Figure 5.1 shows the strategies along with the transmission events that they cause. Intuitively a good transmission scheduling algorithm should incorporate both selfish and pseudo-cooperative strategies, so that both its own and other packets are preserved. This strategy is

---

<sup>1</sup>It may be possible to decompose the line state into its constituent codewords under certain conditions. However, to accomplish this decomposition, the codewords in the codeset and all their rotations must be linearly independent. It can be shown that this property is not satisfied by all codeset designs.

<sup>2</sup>*Codeword estimation* is a scheduling algorithms where the codewords multiplexed on the line are estimated [26] and transmissions are scheduled. Estimation gives a probabilistic estimate of the codewords and therefore it cannot guarantee that other codewords are not destroyed. Therefore, it cannot be regarded as truly cooperative.

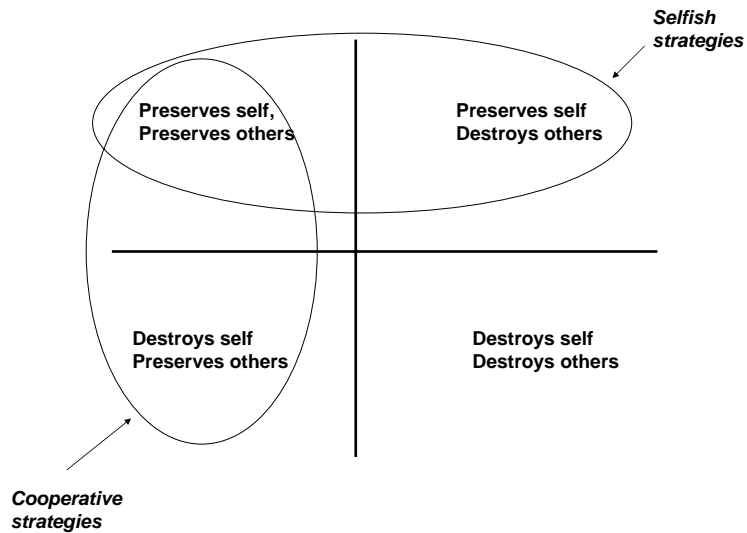


Figure 5.1: The transmission scheduling strategies and the events that they attempt to achieve.

not guaranteed to maximize the network throughput. However, intuitively, this policy should improve the network throughput when compared to systems that are not scheduled. Performance analysis confirms this.

### 5.3 Transmission scheduling algorithms

The following sections define three transmission scheduling algorithms based on the selfish and pseudo-cooperative transmission scheduling strategies. The algorithms are *Pure selfish scheduling*, *Threshold scheduling* and *Overlap section scheduling*. The algorithms are compared to Aloha CDMA, which is optical CDMA without any media access control.

$c_{tx} \leftarrow$ Codeword to be transmitted $state \leftarrow$ State estimate $hstate \leftarrow hardlimit(state)$ $t_d \leftarrow 0$ for $delay = 0$ to $N$ if $(hstate \& c_{tx} \neq c_{tx})$ then mark $delay$ as a feasible delay rotate $c_{tx}$ to the right by one chip $t_d \leftarrow$ any feasible delay
--

Table 5.1: The *Pure selfish* transmission scheduling algorithm.

### 5.3.1 Pure selfish scheduling

The Pure Selfish scheduling algorithm is based on the selfish strategy. The algorithm schedules a packet transmission (chooses a transmission delay) only if the state of the line permits transmission without loss of its own packet. It uses the condition for packet error as described in Chapter 3. It schedules the codeword such that a least one ‘1 chip’ does not overlap with any other ‘1 chip’ in the line state thus ‘selfishly’ attempting to preserve the codeword. No effort is made to control the impact on (preservation or destruction of) other packets. Note that the selfish algorithm does not guarantee that the packet transmitted selfishly is received correctly because subsequent packet arrivals could destroy the codeword.

The algorithm is shown in Table 5.1. The *hardlimit()* function is a function which takes a state vector and reduces any element of the vector which is greater than 1 to 1. The algorithm searches for transmission delays that satisfy the condition. It chooses one of these delays at random. The computational complexity of the algorithm is  $O(N)$ .

### 5.3.2 Threshold scheduling

Threshold scheduling is an algorithm based on both the selfish and the pseudo-cooperative strategies. The algorithm schedules a packet transmission (chooses a transmission delay) only if the state of the line permits transmission without loss of its own packet and the number of chip overlaps in the resulting state is below a threshold. The threshold is expressed as a fraction of the codeword length  $N$ , called the *threshold parameter*  $\alpha$ . It schedules the codeword such that a least one of its ‘1 chip’ does not overlap with any other ‘1 chip’ in the line state and the number of chip overlaps is below a threshold. This reduces the probability of destroying other codewords. The algorithm does not guarantee that other packets are not destroyed, but it reduces the probability of it happening. In addition, this algorithm does not guarantee that the transmitted packet is received correctly because subsequent packet arrivals could destroy the codeword.

The algorithm is shown in Table 5.2. The function *overlaps()* takes a state vector as input and returns the number of elements that have chip overlaps. The algorithm selects delays as described above. It chooses one of the delays at random. The computational complexity of the algorithm is  $O(N)$ .

Note that if the threshold is chosen such that  $\alpha N < w$ , then the threshold algorithm is both selfish and cooperative because no codewords can be destroyed. The sensitivity analysis (Section 5.6) of the threshold parameter on throughput shows that such a low value of the threshold results in low throughput.

```

 $c_{tx} \leftarrow$  Codeword to be transmitted
 $state \leftarrow$  State estimate
 $hstate \leftarrow \text{hardlimit}(state)$ 
 $t_d \leftarrow 0$ 
for  $delay = 0$  to  $N$ 
  if ( $hstate \& c_{tx} \neq c_{tx}$ ) then
     $newstate = state + c_{tx}$ 
     $numoverlaps = \text{overlaps}(newstate)$ 
    if ( $numoverlaps < threshold$ )
      mark  $delay$  as a feasible delay
      rotate  $c_{tx}$  to the right by one chip
 $t_d \leftarrow$  any feasible delay

```

Table 5.2: The *Threshold* transmission scheduling algorithm.

### 5.3.3 Overlap section scheduling

The overlap section scheduling algorithm is based on the selfish and pseudo-cooperative strategies. The algorithm schedules a packet transmission (chooses a transmission delay) only if the state of the line permits transmission without loss of its own packet and the number of chip overlaps in the resulting state is below the number of ones in the state. It reduces the probability of destroying other packets by constraining the number of overlapping chips below the number of non-overlapping chips. The algorithm does not guarantee that other packets are not destroyed, but it reduces the probability of it happening. In addition, this algorithm does not guarantee that the transmitted packet is received correctly because subsequent packet arrivals could destroy the codeword.

The algorithm is shown in Table 5.3. The function *ones()* takes a state vector as input and returns the number of elements that have 1 chips. The algorithm selects delays as described

```

 $c_{ix} \leftarrow$  Codeword to be transmitted
 $state \leftarrow$  State estimate
 $hstate \leftarrow \text{hardlimit}(state)$ 
 $t_d \leftarrow 0$ 
for  $delay = 0$  to  $N$ 
  if ( $hstate \& c_{ix} \neq c_{ix}$ ) then
     $newstate = state + c_{ix}$ 
     $numoverlaps = \text{overlaps}(newstate)$ 
     $numones = \text{ones}(newstate)$ 
    if ( $numoverlaps < numones$ )
      mark  $delay$  as a feasible delay
    rotate  $c_{ix}$  to the right by one chip
 $t_d \leftarrow$  any feasible delay

```

Table 5.3: The *Overlap section* transmission scheduling algorithm.

above. It chooses one of these delays at random. The computational complexity of the algorithm is  $O(N)$ .

## 5.4 Performance study

This section analyzes the transmission scheduling algorithms. First, the metrics used for evaluation are defined. Then, an analysis is described which shows that the algorithms prevent throughput degradation. It also shows that the threshold and overlap section algorithms have lower packet errors compared to the pure selfish algorithm. The analysis is validated through simulation.

### 5.4.1 Assumptions

The transmission scheduling algorithm requires an estimate of the state as input. This chapter assumes *Perfect state estimation* as defined in Chapter 3. The network is homogeneous, *i.e.*,

all nodes use the same transmission scheduling algorithm. In addition it is assumed that the network has a propagation delay equal to zero. These assumptions are unrealistic but serve as an aid to understanding and allow easy analysis. Chapter 6 removes these assumptions.

## 5.4.2 Performance metrics

The main performance metric used in this work is the normalized network throughput. It is measured at different values of the normalized offered load.

### 5.4.2.1 Normalized offered load

The *normalized offered load* is the average arrival rate (in packets/s) expressed as a fraction of the maximum possible arrival rate (in packets/s) of the network when it is used as a single channel network<sup>1</sup>. The arrival rate is defined as the aggregate rate at which packets arrive to all the nodes for transmission on the network. Let the arrival rate be  $\lambda$  packets/s. If the average packet size is  $L$  bytes, the chipping rate is  $B$  c/s, then the maximum possible packet arrival rate on a single channel network is  $B/8L$  packets/s. Therefore, the normalized offered load  $\rho$  is,

$$\rho = 8L\lambda/B$$

Consider an optical CDMA network where the average arrival rate of packets is  $\lambda$  packets/s. The service time of a packet is the average time needed to transmit the packet. Its

---

<sup>1</sup>For an optical CDMA network of chipping rate  $B$  c/s, the maximum possible data rate of the network when used as a single channel network is  $B$  c/s (the chipping rate becomes the bit rate). The maximum possible arrival rate in packets/s is  $B$  divided by the average packet size in bits  $8L$ .

inverse, the service rate is  $\mu$  (in packets/s). For an optical CDMA network using a codeset  $(N, w, \kappa)$ , an average packet size of  $L$  bytes, and a chipping rate  $B$  c/s the service rate is  $\mu = B/(8LN)$  packets/s. Substituting for  $B$  in the expression above, the normalized offered load may be expressed as:

$$\rho = 8L\lambda/(8LN\mu) = \lambda/\mu N$$

#### 5.4.2.2 Normalized network throughput

The normalized network throughput is the ratio of the number of packets that are transmitted over the network without error to the total number of packets offered for transmission multiplied by the normalized offered load. It is a measure of the throughput of packets transmitted without error at a particular offered load.

For example, a normalized offered load of 0.1 means that the packet arrival rate is 10% of the maximum arrival rate of a corresponding single channel network. If the normalized network throughput is 0.09, it means that transmission rate of packets without error is 9% of the maximum arrival rate. So, at an offered load of 0.1, the fraction of packets transmitted without error is  $0.09/0.1 = 90\%$ . The remaining packets (10%) are either not transmitted or are transmitted with errors.

Consider an Aloha-CDMA system where packets are transmitted on arrival. Consider a point on the line at the output of the coupler ( $d = 0$ ). A packet that is passing by that point takes  $1/\mu$  seconds to cross that point. Packets are arriving at that point at a rate of  $\lambda$  packets/sec. They arrive in parallel with no waiting time at that point. The number of codewords

(packets) multiplexed on the line is defined as the number of codewords transmitted in parallel at the output of the coupler (at  $d = 0$ ). By Little's law, the average number of packets multiplexed on the line at that point at any time is  $\lambda/\mu$ .

Consider any other system based on a transmission scheduling algorithm where the same average number of packets is offered for transmission. The transmission scheduling algorithm does not allow all of these packets to be transmitted. Let the average number of packets admitted by the transmission scheduling algorithm and multiplexed on the line at the output of the coupler at any time be  $N_{online}$ , of which a fraction  $P_e$  are lost due to error.

The fraction of packets transmitted without error is  $N_{online}(1 - P_e)/(\lambda/\mu)$ . The normalized offered load is  $\rho = \lambda/(\mu N)$ . Therefore, the normalized throughput of the network is

$$\begin{aligned} Th_{norm} &= (N_{online}(1 - P_e)/(\lambda/\mu))(\lambda/(\mu N)) \\ &= N_{online}(1 - P_e)/N \end{aligned}$$

### 5.4.3 Analysis

This section describes a mathematical analysis of the transmission scheduling algorithms. The analysis models the system as a Markov chain. The state of the line at the output of the coupler is used as the system state. Packet arrivals and departures cause state transitions. The network throughput may be found by obtaining expressions for the average number of codewords multiplexed on the line at a point in each state, the packet error rate in each state and the equilibrium state probabilities. A broad outline of the analysis follows. Details may be found in the appendices. The analysis follows the steps below:

- First, a concise representation of line state that allows easy mathematical manipulation is defined (Appendix A).
- Using this state representation, expressions are derived for the number of codewords on the line and the probability of packet error when the system is in any state (Appendix B).
- Based on the transmission scheduling algorithm (Aloha-CDMA, Pure Selfish, Threshold, Overlap section) the state transition probabilities are calculated and a state transition diagram is constructed. Under the assumptions of Poisson arrivals and exponentially distributed packet sizes, the state transition diagram can be viewed as a Markov chain. The Markov chain is solved for equilibrium state probabilities at a particular offered load (Appendix C).

The analysis is used to determine the normalized throughput at any normalized offered load. A graph of the normalized throughput vs. normalized offered load for different scheduling algorithms is shown in Figure 5.2. The traffic model is Poisson arrivals and exponentially distributed packet sizes. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset. The graph indicates that a system with no transmission scheduling (Aloha-CDMA) suffers throughput degradation. Beyond an offered load of around 0.5, the network throughput decreases and tends to zero at high offered loads. In contrast, all the transmission scheduling algorithms prevent throughput degradation. Throughput is stabilized at around 30% of the maximum throughput and remains stable as offered load increases.

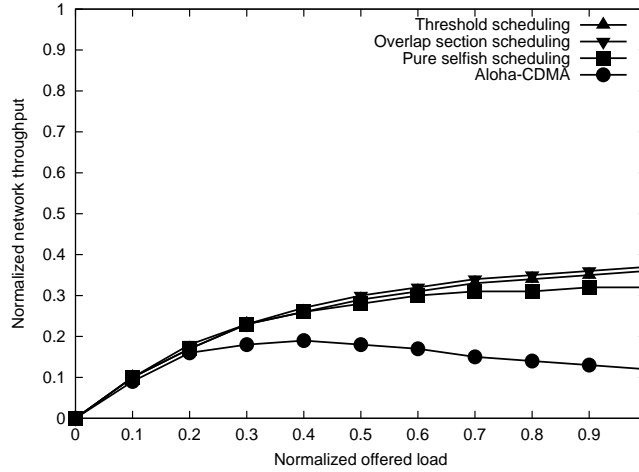


Figure 5.2: Normalized network throughput vs. normalized offered load for different transmission scheduling algorithms based on analysis. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is  $(10, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

#### 5.4.4 Simulation

A discrete event based packet simulator is used to validate the mathematical analysis. The objective of the simulator is to validate the analysis and perform a deeper study of transmission scheduling. The simulator models multiple nodes on a broadcast shared medium optical CDMA LAN. It supports different state estimation algorithms, transmission scheduling algorithms and a hard-limiting correlation receiver. Unless specified otherwise, the default parameters for the simulations are specified in Table 5.6. The optical orthogonal codeset construction method is the greedy construction method [10]. It is used to generate several codesets for a given set of codeset parameters. The results in this work did not change substantially with the particular codeset or the algorithm used to generate the specific optical

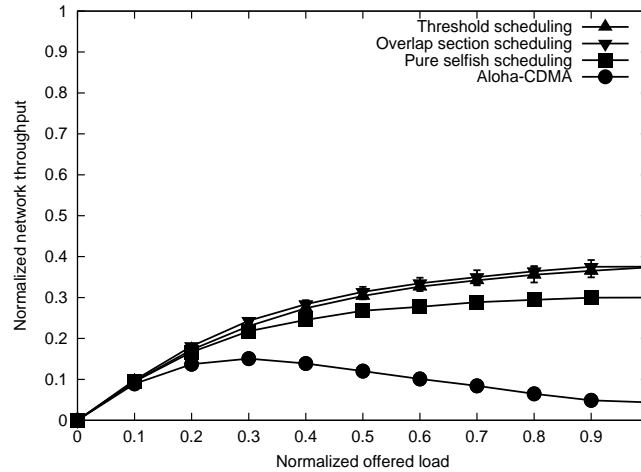


Figure 5.3: This graph shows the normalized network throughput vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (10,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

orthogonal codeset. The results from the simulations are the mean of around 10 runs of anywhere from 10,000 to 100,000 packets each and standard deviations are shown on the graphs.

Figure 5.3 shows the results of simulation for the same codeset as described in the analytical results. The results are quite similar to the analytical results. All transmission scheduling algorithms prevent throughput degradation. In addition, the overlap section and threshold scheduling show marginally higher throughput than pure selfish scheduling. The analytical model over predicts the throughput for Aloha-CDMA. This is because the analysis is based on a finite state model. A finite state model is suitable for transmission scheduling algorithms,

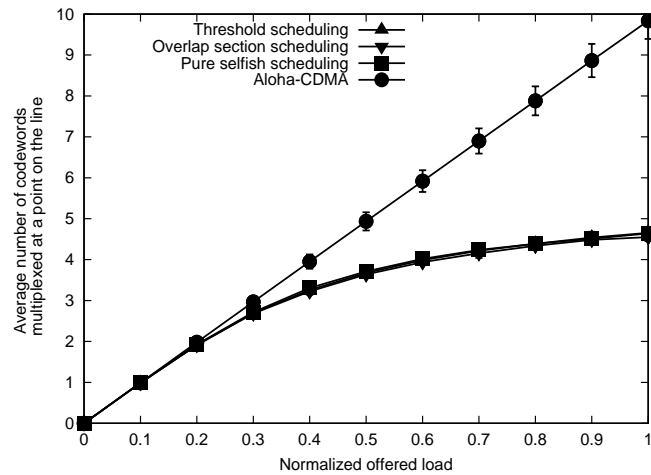


Figure 5.4: The average number of codewords multiplexed at a point on a receiver fiber vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is  $(10, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

which limit the traffic on the line. However for Aloha-CDMA, the finite state model over predicts the equilibrium state probabilities. As a result, the analytical results differ slightly from simulation.

### 5.4.5 Discussion

Though Figures 5.2 and 5.3 indicate that all three scheduling algorithms have approximately the same throughput, the algorithms differ when the number of codewords multiplexed on the line at a point and the packet error rate is considered. Figures 5.4 and 5.5 show the average number of codewords multiplexed on the line at a point on a receive fiber and the average packet error rate for the transmission scheduling algorithms.

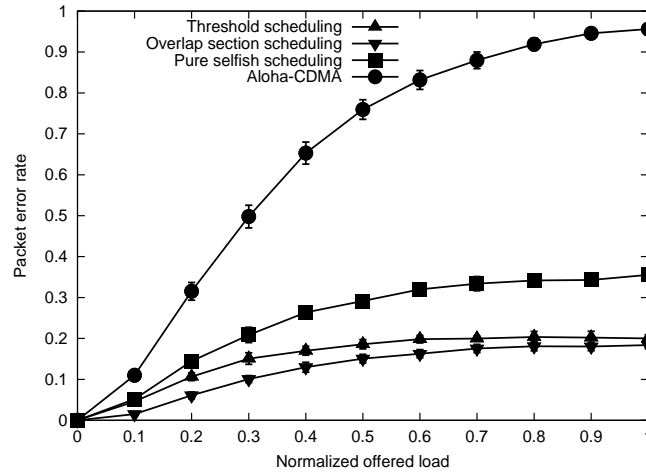


Figure 5.5: Packet error rate vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is  $(10, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

There is a trade-off between the number of codewords multiplexed on the line at a point and the packet error rate. The transmission scheduling algorithm tells a node *if* it can transmit. It also tells the node *when* to transmit. By doing this, the transmission scheduling algorithm controls two quantities:

- The number of codewords multiplexed on the line at a point.
- The transmission delays of the codewords multiplexed on the line, which affects the packet error rate.

The trade-off can be understood by considering the performance of Aloha-CDMA. As the offered load increases, the number of codewords on the line for Aloha-CDMA shows a linear increase with offered load. Initially as the number of codewords on the line increases and the

network throughput increases. However, as the number of codewords on the line increases further, interference errors increase and as a result the packet error rate increases. As a result the network throughput falls. Therefore, as the offered load increases, the network throughput attains a maximum and then decreases. To the left of the maximum, the throughput is low due to the low number of codewords on the line. To the right of the maximum, the throughput is low due to the higher packet error rate. The objective of the transmission scheduling algorithm is to keep the system operating point close to the maximum irrespective of the offered load. Therefore, if the transmission scheduling algorithm is aggressive and allows more codewords on the line, the packet error rate increases, lowering throughput. On the other hand, if the algorithm is conservative and does not allow enough codewords on the line, the throughput is low. So the transmission scheduling algorithm must carefully balance the number of codewords on the line and the transmission delays to maximize throughput.

The difference between the three scheduling algorithms is evident in Figure 5.5. There are significant differences in the packet error rate. The threshold and overlap section scheduling algorithms are conservative and constrain the number of overlapping chips. This results in a low number of errors due to interference. However, the selfish algorithm is aggressive and admits a larger number of codewords with a higher probability of packet error. When packets are lost due to interference errors, the higher layers of the protocol stack must recover through some form of Automatic Repeat Request (ARQ) or Forward Error Correction (FEC).

This leads to the first main conclusion of this study: *All three transmission scheduling algorithms prevent throughput degradation at high offered load. All have approximately the*

same network throughput (which is greater than that of Aloha-CDMA). However, the threshold scheduling and overlap section scheduling algorithms lose a lower number of packets due to interference errors than the pure selfish algorithm.

## 5.5 Performance limits of transmission scheduling algorithms

This section evaluates whether there is a limit to the performance of transmission scheduling. The previous section showed that transmission scheduling improves the performance of an optical CDMA system when compared to an Aloha-CDMA system. As the offered load increases, the throughput of transmission scheduling increases and then levels off. However, the analysis gives no intuition as to why the throughput levels off and why it is less than the maximum throughput. It also does not establish whether the throughput at which it levels off can be increased or whether there is a limit to the network throughput of optical CDMA when used with a media access control protocol. The existence of a limit would be useful as a performance metric with which to compare the performance of the transmission scheduling algorithms. To establish whether there is a limit, the performance of transmission scheduling is compared to the performance of two optimal codeword scheduling algorithms. The algorithms are called *Centralized perfect scheduling* and *Distributed perfect scheduling* and are described in the next two sections. These algorithms are codeword scheduling algorithms for optical CDMA that guarantee that the event is always *Preserves self/Preserves others*. Each algorithm requires a capability that transmission scheduling does not have. The algorithms are unrealistic and cannot be implemented in a real network. However, they are ideally

<pre> state ← State estimate c<sub>tx</sub> ← 0 if (numonline == 0)   c<sub>tx</sub> ← any codeword else   c<sub>tx</sub> ← any codeword where (exactly 1 ‘1 chip’ does not have a chip overlap)   AND (all w – 1 other chips have chip overlaps) t<sub>d</sub> ← 0 </pre>
--

Table 5.4: The *Centralized Perfect Scheduling* algorithm.

suites to evaluate the throughput performance of the transmission scheduling algorithms and to understand the causes for its low throughput.

### 5.5.1 Centralized perfect scheduling

*Centralized Perfect Scheduling (CPS)* is an online codeword scheduling algorithm. It takes as input the state of the line and returns as output a codeword and a transmission delay. The codeword and the delay are chosen such that zero interference errors occur and the network throughput is maximized. CPS assumes

- Perfect state estimation with  $a = 0$ .
- Tunable transmitters, tunable receivers with instantaneous tuning and pre-transmission coordination.

A network based on CPS operates as follows: When a node has a packet to transmit, it estimates the state of the line (perfect state estimation). It uses the CPS algorithm to calculate the codeword to use for encoding and its transmission delay. Note that the codeword for encoding is chosen, unlike transmission scheduling where the codeword for encoding

$c_{tx} \leftarrow$ Codeword to be transmitted $state \leftarrow$ State estimate $\{c_n, t_n\} \leftarrow decompose(state)$ for $delay = 0$ to $N$ $event \leftarrow calceventtype(\{c_n, t_n\}, c_{tx}, delay)$ if ( $event == Preserves\ self/Preserves\ others$ ) mark $delay$ as a feasible delay rotate $c_{tx}$ to the right by one chip $t_d \leftarrow$ any feasible delay
--

Table 5.5: The *Distributed Perfect Scheduling* algorithm.

depends on the node to which the packet is to be sent. The transmitter and the receiver to tune instantaneously to the codeword (the pre-transmission coordination may be through a common coordination channel or through a centralized server). The codeword and delay are chosen such that the event is always *Preserves self/Preserves others*, i.e., no codeword has an interference error. The codeword is chosen such that the number of codewords multiplexed on the line at a point is maximized. The scheduling algorithm is defined in the Table 5.4. Appendix E describes CPS in detail. It shows that centralized perfect scheduling is an upper bound on the throughput of any transmission scheduling algorithm that follows a selfish strategy.

### 5.5.2 Distributed perfect scheduling

*Distributed Perfect Scheduling (DPS)* is an online codeword scheduling algorithm. It takes as input an estimate of the state and the codeword to be used for encoding and returns as output the transmission delay. The delay is chosen such that zero interference errors occur and the network throughput is maximized. DPS assumes

- Perfect state estimation with  $a = 0$ : As defined previously.
- Perfect line state decomposition: There exists an algorithm that allows the line state to be decomposed uniquely into the codewords on the line.
- Tunable transmitter/Fixed receiver: The receiver it is allocated a fixed codeword and the transmitter has to tune to it. The codeword allocation to the receiver is assumed to be uniform random over the codeset.

A network based on DPS operates as follows: When a node has a packet to transmit, it estimates the state of the line. It uses the DPS algorithm to calculate the delay to transmit. The DPS node is capable of perfect state decomposition, *i.e.*, it can decompose the state vector into a set of codewords and delays. Perfect state decomposition is a capability that transmission scheduling does not possess. The transmission delay is chosen such that the node *Preserves self/Preserves others*, *i.e.*, no interference error occurs. The scheduling algorithm is defined in the Table 5.5.

The function *calceventtype()* takes as input the codeword for encoding, its transmission delay, the set of codewords multiplexed on the line and their delays, and returns the resulting event type (PS/PO, PS/DO, DS/PO or DS/DO).

### 5.5.3 Discussion

Figure 5.6 shows a graph of normalized network throughput vs. normalized offered load for CPS, DPS and the threshold scheduling algorithm. The threshold algorithm used a threshold of 0.5 and perfect state estimation. The codeset is (10,3,3). The traffic model is Poisson

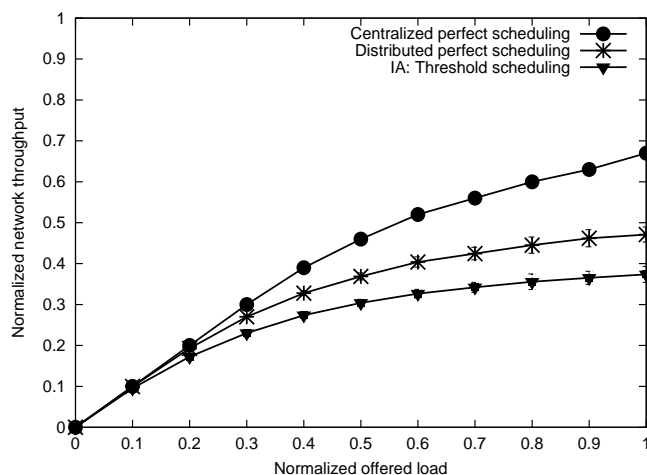


Figure 5.6: The normalized throughput vs. normalized offered load for the Centralized and Distributed Perfect Scheduling algorithms and threshold scheduling. The results are based on both simulation and analysis. The parameters are detailed in the text. The codeset is  $(10,3,3)$ . The threshold algorithm used a threshold of 0.5.

arrivals and exponentially distributed packet lengths. The results shown are the average of 10 runs.

The results show that the thresholding algorithm performs reasonably well. The difference between the throughput of Distributed Perfect Scheduling and Interference Avoidance is about 10% of the maximum throughput. CPS performs the best because the optimal codeword and transmission delay are chosen. The reason that it does not reach 100% throughput is because of the pseudo-orthogonal property of the optical CDMA codewords. There is a limit to the number of codewords that can be multiplexed without any interference errors. DPS differs from CPS in that it chooses only the optimal delay and not the codeword. The difference between CPS and DPS represents the penalty paid for choosing a distributed architecture with pre-allocated codewords and non-tunable receivers. The threshold scheduling algorithm has

a lower throughput than DPS. The difference between DPS and threshold scheduling represents the penalty paid because perfect line state decomposition is not possible. However, the difference between the two is not large. This leads to the following conclusion:

*Transmission scheduling algorithms can perform reasonably close (within 10% of the maximum throughput) to optimal distributed codeword scheduling algorithms.*

This means that given limited information, threshold scheduling algorithms performs quite well. An open area of research is to determine whether there exist other transmission scheduling algorithms whose performance approaches the bound of DPS.

## **5.6 Sensitivity analysis**

This section describes a deeper sensitivity analysis of the transmission scheduling algorithms. The motivation is to understand the effect of varying parameters at the physical layer (codeset parameters), the media access control layer (the scheduling algorithm parameters) and the traffic model (packet arrival and size distributions). The objective of this study is to quantify the impact of these factors on the transmission scheduling algorithms. The sensitivity analysis consists of quantifying the:

- Effect of varying the codeset length.
- Effect of varying the codeset weight.
- Effect of varying the cross correlation parameter.
- Effect of varying the number of wavelengths.

Parameter	Default value
Codeset parameters:	
Codeset length $N$	100
Number of wavelengths $\Lambda$	1
Codeset weight $w$	3
Maximum cross-correlation parameter $\kappa$	3
Number of codewords in codeset	100
Chipping rate:	10 $Gc/s$
Codeword allocation:	Uniform random
Interference Avoidance parameters:	
Transmission scheduling algorithm:	Threshold scheduling
Threshold:	0.5
State estimation algorithm:	Perfect state estimation
Traffic parameters:	
Inter-arrival time distribution	Exponential
Normalized offered load	1
Packet size distribution	Exponential
Average packet size	1000 <i>bytes</i>
Destination address distribution:	Uniform random
Topology parameters:	
Node to coupler distance distribution	Deterministic
Average node to coupler distance	0 <i>m</i>
Number of nodes	100

Table 5.6: Parameter list and default values for the transmission scheduling sensitivity study.

- Effect of varying the threshold transmission scheduling parameter.
- Effect of different packet size distributions.
- Performance under realistic network traffic.

### 5.6.1 Effect of varying the length of the codeset

Increasing the length of the codeset has two effects. As  $N$  increases, the number of codewords multiplexed on the line increases because the scheduling algorithm can multiplex a larger

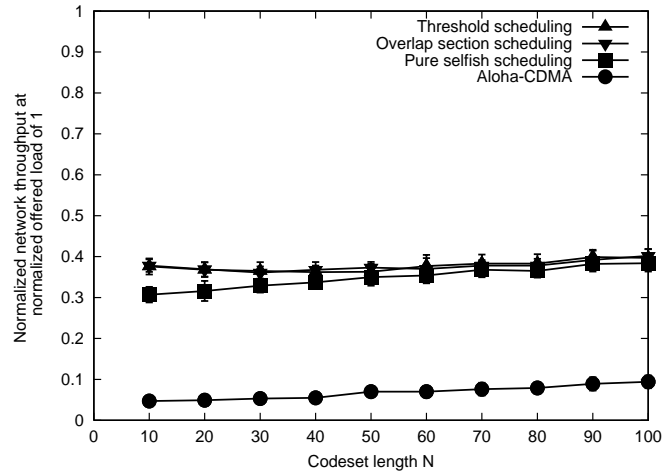


Figure 5.7: The normalized network throughput at normalized offered load of 1 vs. the codeset length  $N$  for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset weight is 3 and  $\kappa = 3$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

number of codewords simultaneously on the line. Therefore, more nodes can transmit in parallel without error. However, as  $N$  increases the data rate reduces. The results show that the two effects balance each other and the network throughput is constant when  $N$  is varied. Figure 5.7 shows a graph of network throughput at an offered load of 1 vs. the codeset length. Therefore, network throughput is independent of codeset length and transmission scheduling scales with increase in codeset length.

### 5.6.2 Effect of varying the weight of the codeset

Increasing the codeset weight  $w$  has two effects: As  $w$  increases, the threshold on the correlation receiver can be increased. Increasing the threshold makes the codewords more resistant

to interference, because it takes a larger number of chip overlaps to cause a false positive error. However, with the increase in weight, each codeword causes more interference with other codewords. An increase in  $w$  reduces the number of potential transmission delays that the transmission scheduling algorithm can use to schedule codewords on the line without causing interference errors. This reduces the number of codewords that can be multiplexed on the line. The results show that as the weight increases the throughput decreases rapidly. The reduction in the multiplexing of codewords on the line and the increased interference offset any gains in the resistance to interference. It has been shown [58] that codesets with higher weight have better bit error rate characteristics at low loads. At low loads, the increase in resistance to interference dominates resulting in higher throughput for high weight codesets. However as the load is increased, the effect of interference tends to dominate, resulting in lower throughput for high weight codesets. Figure 5.8 shows a graph of the network throughput at offered load of 1 vs. the weight. The length of the codeset was 100. The weight is increased from 2 to 10. The graph indicates that lower codeset weights improve the performance of transmission scheduling. The graph also shows that the effect of increasing the weight is more detrimental to the selfish algorithm than the threshold and overlap scheduling algorithms. This is because the threshold and overlap scheduling algorithms limit the number of overlaps while the selfish does not. As the codeset weight increases, for the same number of codewords multiplexed on the line, the probability that the selfish algorithm will result in much more overlaps than the threshold scheduling algorithm is high. This results in lower throughput for the selfish algorithm.

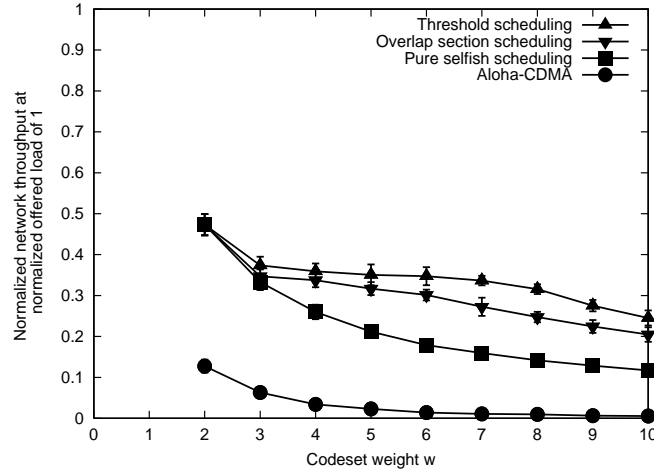


Figure 5.8: Normalized network throughput at normalized offered load of 1 for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length is 100 and  $\kappa = 3$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

### 5.6.3 Effect of varying the cross correlation parameter

The cross correlation parameter  $\kappa$  controls the interference between pairs of codewords. It has traditionally been used as a parameter in codeset design.

Although  $\kappa$  controls the interference between pairs of codewords, packet loss is usually caused by interference between several codewords. Therefore, a low cross correlation parameter does not necessarily result in low interference loss at high offered load.  $\kappa$  also controls the number of codewords in the codeset. Johnson [23] showed a bound on the number of codewords in a codeset for a given  $(N, w, \kappa)$  codeset. If  $\kappa$  is low ( $= 1$ ), the interference between pairs of codewords is minimized, but the number of codewords in the codeset is low<sup>1</sup>.

<sup>1</sup>In networks where the number of nodes is greater than the number of codewords, codewords are shared among the nodes, with packets demultiplexed using a higher layer unique address

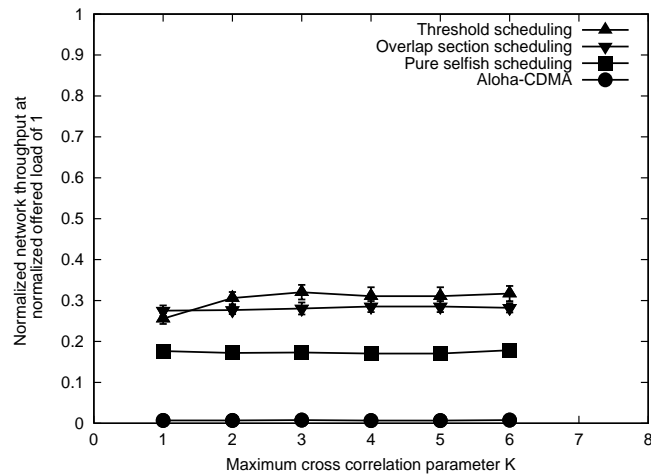


Figure 5.9: Normalized network throughput at normalized offered load of 1 for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length  $N = 42$  and weight  $w = 6$ . The cross correlation parameter  $\kappa$  is varied from 1 to 6. For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

If  $\kappa$  is high ( $= w$ ), the interference between pairs of codewords may be high, but the number of codewords in the codeset is also high.

Figure 5.9 shows the performance of the transmission scheduling algorithms for a codeset of length 42, weight 6 as  $\kappa$  is varied from 1 to 6. The throughput does not vary significantly with  $\kappa$ . At high offered load, the number of codewords multiplexed on the line is so large that interference loss is primarily due to interference among all codewords on the line. A low value for  $\kappa$  does not help prevent interference loss at high offered load. Note that the throughput in the figure is lower than the throughput shown in previous performance results. This is because of the high weight codeset ( $N = 42, w = 6$ ).

#### 5.6.4 Effect of varying the number of wavelengths

The use of other orthogonal dimensions such as wavelengths for codeset design has been proposed [67]. These are called multidimensional codesets. As the number of wavelengths is increased, a larger fraction of the optical fiber capacity is used for transmission. As the wavelengths increase, the scheduling algorithm should be able to schedule a larger number of codewords simultaneously onto the line. So, the increase in wavelengths increases overall system transmission capacity and increases the capacity utilized by each user. The results show that both effects balance each other: though more users can transmit simultaneously, each user uses more capacity to transmit. The extra capacity that is available through multiple wavelengths is utilized in the same proportion as the utilization of the capacity of a single wavelength. As a result, the overall network throughput, *i.e.*, the fraction of packets transmitted without error remains the same. Figure 5.10 shows a graph of network throughput at an offered load of 1 vs. the number of wavelengths. The results show the throughput curves different (10, 3, 3) codesets with variable number of wavelengths. The graph indicates that the network throughput is independent of the number of wavelengths. As the capacity of the fiber increases, the performance of transmission scheduling scales.

#### 5.6.5 Effect of varying the threshold scheduling algorithm parameter

The threshold scheduling algorithm was described in Section 5.3. The performance of this algorithm is sensitive to the threshold parameter  $\alpha$ . The threshold scheduling algorithm constrains the number of overlaps in the state to be less than  $\alpha N$ . At low values of the threshold,

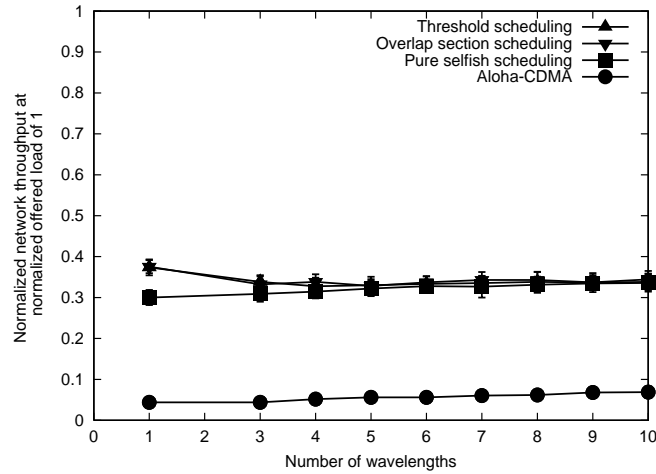


Figure 5.10: Normalized network throughput at normalized offered load of 1 vs. the number of wavelengths for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length is 10, weight is 3 and  $\kappa = 3$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

the algorithm constrains the number of codewords on the line and maintains a low probability of packet error. As the threshold increases, the algorithm allows more codewords on the line but with increased probability of error. When the threshold is equal to 1, the threshold scheduling algorithm reduces to the pure selfish scheduling algorithm. Figure 5.11 shows a graph of the network throughput at offered load of 1 vs. the threshold. The throughput is low for low values of the threshold. As the threshold increases, there is an increase in throughput. Beyond a certain value, the throughput degrades due to increased number of codewords interfering on the line. Above the value of 0.6, the degradation stops and throughput levels off. This is because beyond this value the threshold is longer the limiting factor that prevents degradation: the selfishness of the algorithm prevents further degradation. At low thresholds,

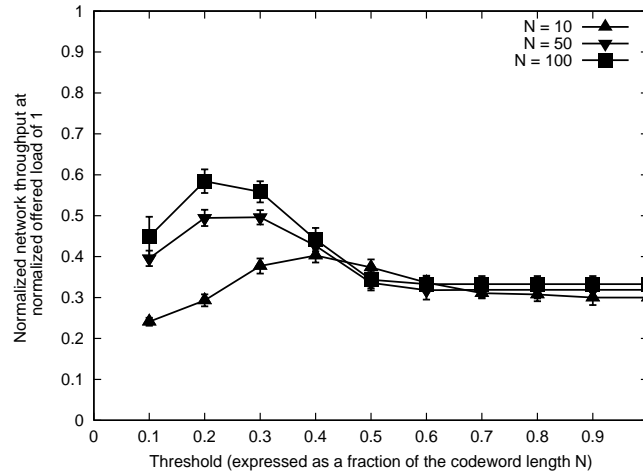


Figure 5.11: Normalized network throughput at normalized offered load of 1 vs. threshold parameter  $\alpha$  for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (100,3,3). The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

codesets with large lengths perform better because the transmission scheduling algorithm is able to schedule more codewords on the line. The graph indicates that cooperative algorithms for which the threshold parameter is below the weight have low throughput.

### 5.6.6 Effect of packet size distribution

Figure 5.12 shows the packet throughput as the average packet size is varied. The traffic model is Poisson arrivals and exponentially distributed packet sizes. The figure shows that the packet size has no effect on packet throughput. However, studies indicate that real network traffic packet size distributions may not be exponential [53]. Recent packet statistics obtained from a backbone network [66] exhibit a trimodal distribution. In one trace, about 70% of the packet sizes were 40 bytes, about 20% were 1500 bytes and the remaining were

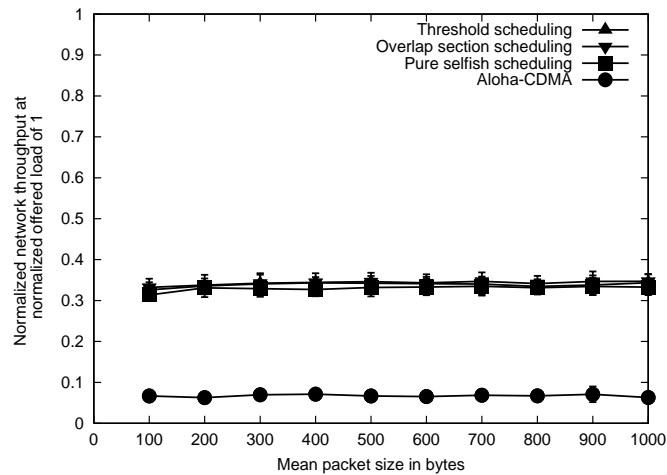


Figure 5.12: Normalized network throughput vs. average packet size at a normalized offered load of 1. The results are based on simulation. The traffic model is Poisson arrivals and exponentially distributed packet sizes. The codeset is (100, 3, 3). The algorithm was threshold scheduling; the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

around 500 bytes long. A traffic model with such a trimodal packet size distribution was used to drive a simulation that used Poisson arrivals, uniform random allocation of codewords and a (100, 3, 3) codeset. The results are shown in Figure 5.13. The graph shows interesting behavior. Aloha-CDMA does not degrade as much as in the case of Poisson traffic/exponential packet sizes. The other transmission scheduling algorithms have almost 25% higher throughput when compared to performance with exponentially distributed packet sizes. A majority of the packets are small size packets (40 bytes). Most of them get through without error. Therefore, the aggregate network packet throughput is higher. There are fewer longer packets transmitted on the line and a larger fraction is lost due to errors. This behavior occurs when the fraction of shorter packets is fairly high (60-70%). This is similar to behavior on

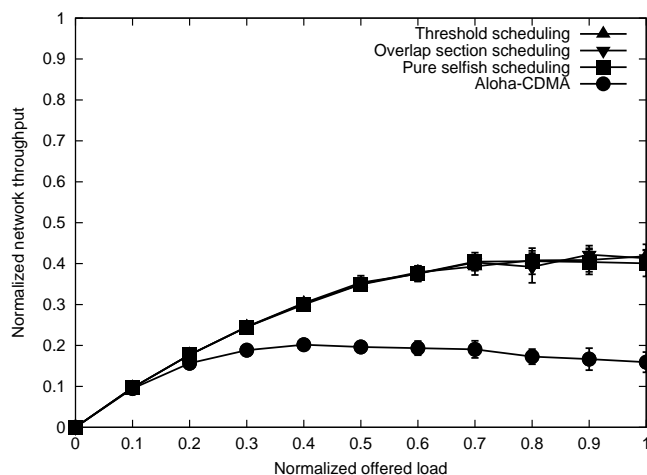


Figure 5.13: Normalized network throughput vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with a trimodal packet size distribution (70% 40 bytes, 20% 1500 bytes and 10% 500 bytes). The codeset is (100,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.5. The destination of a packet is chosen uniform randomly from all the nodes and codewords are allocated to nodes by choosing them randomly from the codeset.

wireless links where short packets tend to experience lower error rates than long packets [4].

The improvement in throughput is caused by a *squeeze through effect* where shorter packets experience lower packet error rates than longer packets.

The squeeze through effect can be demonstrated analytically and through simulation for a network with a bimodal distribution of packet sizes which uses a pure selfish transmission scheduling algorithm and a codeset with  $\kappa = w$ . Consider a network where the traffic has two packet types of sizes  $l_1$  and  $l_2$  where  $l_1 < l_2$ . Let the fraction of packets of size  $l_1$  be  $\gamma$ . Then the average packet size on the network is  $l_{avg} = (\gamma)l_1 + (1 - \gamma)l_2$ . The throughput of such a network can be calculated by finding the probability of packet error  $P_{error}$  and the number of codewords multiplexed at a point on the line  $N_l$ .

The average number of codewords on the line at any point on the receive fiber  $N_l$  depends on the transmission scheduling algorithm. If there are  $N_l$  codewords on the line, then  $N_l w$  ‘1 chips’ were added to the state. Of these,  $N_l + (w - 1)$  ‘1 chips’ were aligned with ‘0 chips’ of the state (because of selfish scheduling) and  $(N_l - 1)(w - 1)$  ‘1 chips’ were added randomly to any position ( $\kappa = w$ ).

The probability that the state vector has no ‘0 chips’ is

$$P_{full} = 1 - \left(1 - \frac{1}{(N - (N_l + (w - 1)))}\right)^{(N_l - 1)(w - 1)}$$

At an offered load of 1, packets arrive for transmission at a rate much higher than the rate at which packets are transmitted (the transmission scheduling does not allow all packets to be transmitted). When a packet departs from the line, a few chips of the state may change from 1 to 0. The next packet arrival results in a transmission of a packet such that the ‘0 chips’ are filled. Therefore, under equilibrium conditions, for the pure selfish scheduling algorithm,  $P_{full}$  is close to 1. Through simulation  $P_{full}$  is determined to be around 0.85 for the pure selfish algorithm under Poisson arrivals and exponentially distributed packet sizes. This analysis assumes that this is true for bimodal packet distributions too. Therefore, the number of codewords on the line can be calculated by finding  $N_l$  such that  $P_{full}$  is close to 1.

Interference errors in a packet on the line are caused by packets that arrive during the packet’s transmission. The transmission scheduling algorithm allows only a fraction of the arriving packets (called colliding packets) to be transmitted. The probability of packet error in a codeword on the line is the probability that at least one of its colliding packets causes

an interference error. If the number of colliding packets is  $n_c$ , then  $n_c$  ‘1 chips’ are added to the state selfishly (align with ‘0 chips’) and  $n_c(w-1)$  ‘1 chips’ are added to the state in random positions. These random positions are chosen from  $N-1$  possible choices (1 chip is chosen selfishly). The probability that one of the added ‘1 chips’ overlaps with the one of the ‘1 chips’ of the codeword on the line is  $p = w/(N-1)$ . The probability of packet error is the probability that more than  $w-1$  overlaps occur. Therefore,

$$P_{error} = 1 - \sum_{k=0}^{w-1} \binom{n_c(w-1)}{k} p^k (1-p)^{n_c(w-1)-k}$$

At a normalized offered load of 1, the average packet inter-arrival time is  $t_{arrival} = l_{avg}/B$  where  $B$  is the chipping rate of the network. The transmission time for packets of type 1 is  $t_1 = l_1/(B/N)$  where  $N$  is the codeset length. The average number of colliding packets for packet type 1 is,

$$n_{c1} = (t_1/t_{arrival}) * (N_l/N)$$

A similar expression can be derived for  $n_{c2}$ . This can be used to calculate the probability of packet error for each type of packet  $P_{error1}$  and  $P_{error2}$ .

The normalized network throughput is given by:

$$Th = (N_l/N)(\gamma(1 - P_{error1}) + (1 - \gamma)(1 - P_{error2}))$$

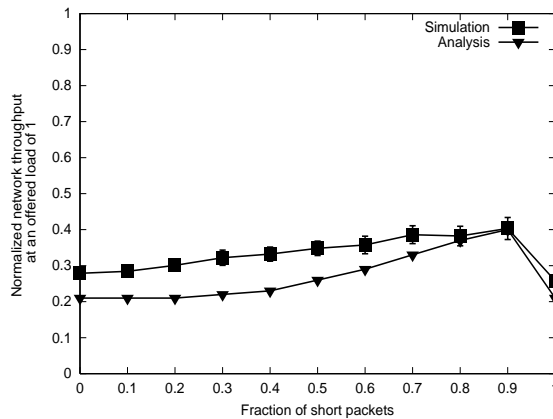


Figure 5.14: The squeeze through effect. The throughput is maximized when the fraction of short packets is 0.9. The graph shows both analytical and simulation results. The transmission scheduling algorithm is pure selfish. The packet sizes are 50 bytes and 1000 bytes. The codeset is (100,3,3).

A graph of normalized network throughput vs. the fraction of short packets is shown in Figure 5.14. The packet sizes were set to 50 bytes and 1000 bytes. The results (both analysis and simulation) show that the normalized network throughput peaks at a particular value of the fraction of small packets confirming the squeeze through effect. The long packets experience high packet error rates and the short packets experience low error rates. The throughput attains a maximum when the fraction of short packets reaches a particular value (around 0.9).

The higher throughput of shorter packets may not be a desirable characteristic, because it is unfair to longer packets. Future work will address the issue of providing a uniform packet error rate to all traffic. Possible alternatives include using constant packet sizes or using different codeset lengths for different size packets.

### 5.6.7 Performance with real network traffic

This section discusses the performance of transmission scheduling with realistic network traffic. The assumption of Poisson arrival and exponential distributed packets lengths is convenient for analysis. However, the inter-arrival and packet size distribution of real network traffic could be different depending on when and where it is observed.

Simulations were performed with traffic traces obtained from a real network link to understand the impact of real packet arrival times. Traffic traces from a single OC48 [51] link were used. Several of these traces were merged to generate traffic of different offered loads. The packet sizes, source addresses, and destination addresses were preserved during merging. The packet size distribution was trimodal (35% of the packets were 40 bytes, 30% were 500 bytes and 35% were 1500 bytes). The traffic file had approximately 6000 unique source addresses and 40000 unique destination addresses. In contrast to all the previously described results, in this case the destination addresses were mapped to codewords. Codeword reuse was used where codewords were insufficient.

The results of the simulation are shown in Figure 5.15. The results indicate that the performance is similar to that of the Poisson arrivals/exponentially distributed model, indicating that it was a reasonable choice for analysis. Note that the performance improvements of the previous section due to the squeeze through effect are not visible here. The proportion of short packets (40 bytes) in the traffic was insufficient to cause the squeeze through effect.

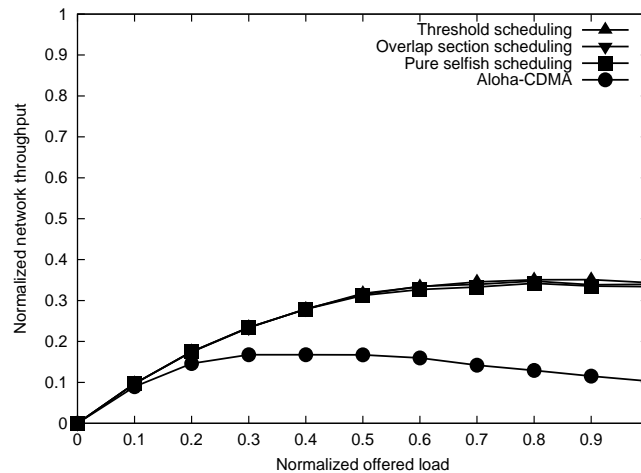


Figure 5.15: Normalized network throughput vs. normalized offered load for different transmission scheduling algorithms. The results are based on simulation. The traffic model was based on real network traffic traces (see description). The codeset is  $(100, 3, 3)$  and codewords are allocated to addresses. For the threshold scheduling algorithm, the threshold parameter was set to 0.5

## 5.7 Conclusions and future work

This work has presented an analysis of transmission scheduling algorithms. The transmission scheduling algorithms trade-off the number of codewords on the line with the packet error rate to keep the network throughput high. It was shown that several transmission scheduling algorithms perform comparably well (with 15% of max) compared to optimal codeword scheduling algorithms. Simulations also showed that transmission scheduling performs well when used with a realistic traffic model based on traffic obtained from a real network.

Open problems in this area include the design of transmission scheduling algorithms that approach the bound of optimal codeword scheduling algorithms such as CPS.

## Chapter 6

### State estimation

*State estimation* is the process by which a node calculates an estimate of the state at a point on the line at some time using state observations obtained at some (possibly different) point on the line at some (possibly different) time. The estimated state is used as input to a transmission scheduling algorithm. Chapter 5 analyzed the throughput of transmission scheduling under perfect state estimation with a propagation delay of zero. This chapter studies the performance of Interference Avoidance with *realistic state estimation*, *i.e.*, using a state estimation algorithm in a network with normalized propagation delay greater than zero. It shows that as the diameter of the network increases, errors occur in state estimation. Errors in state estimation result in inaccurate transmission scheduling and a degradation of throughput. The degradation can be mitigated by using the threshold scheduling algorithm with a low threshold and a long codeset length.

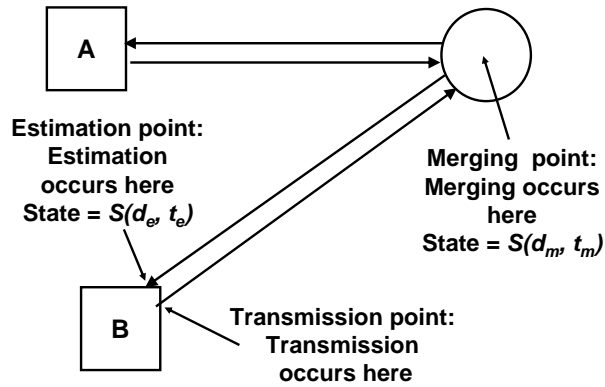


Figure 6.1: The points on the optical fiber where state is estimated and transmissions merge. The state is estimated by a node at its estimation point at the estimation time. The node transmits a packet at the transmission time and it merges with other packets at the merging point (coupler) at the merging time.

Section 6.1 formulates the state estimation problem and discusses its relationship to transmission scheduling. Section 6.2 discusses the state estimation algorithms. Section 6.3 discusses the causes for packet loss in a network with realistic state estimation and  $a > 0$ . Section 6.4 studies the performance of realistic state estimation. Section 6.5 presents results for the sensitivity analysis of state estimation algorithms.

## 6.1 The state estimation problem

The state estimation problem can be stated as follows: Given a series of observations of the state  $S_0, S_1, S_2, \dots, S_{K-1}$  at times  $t, t + t_b, t + 2t_b, \dots, t + (K - 1)t_b$ , (where  $t_b$  is a bit time)

calculate an estimate of the state such that the throughput of the transmission scheduling algorithm is maximized.

Figure 6.1 illustrates the process of state estimation. Node  $B$  is the node estimating the state. It collects  $K$  observations of the state from the signal at the estimation point. Chapter 7 describes hardware design using optical devices and electronic hardware that is capable of performing the observation function. The state estimation algorithm takes the  $K$  observations of the state as input. It calculates an estimate of the state called the *estimated state*. It uses this value as the input to the transmission scheduling<sup>1</sup>.

When the node transmits its packet, it merges with other transmitted packets at the coupler. The state of the line at the coupler at the merging time may differ from the estimated state. The reason for the difference is because of state transitions. A state transition could occur during the collection of observations or between the estimation and the merging. A state transmission could result in an inaccurate state estimate.

Note that the metric to measure state estimation is the aggregate network throughput when it is used with a particular transmission scheduling algorithm. Other metrics such as correlation or distance metrics ( $L_1, L_2, L_p$  norms) could have been chosen to measure the accuracy of state estimation. However, it is difficult to relate these metrics to potential improvements in the throughput of the network. Therefore, because state estimation is always used with transmission scheduling, the throughput of transmission scheduling is the most suitable metric.

---

<sup>1</sup>This work assumes that the estimation point and transmission point are separated by an integral number of state spans. If this is not true, a ranging procedure as described in Chapter 7 can determine the difference in chip times and rotate the estimated state by the appropriate offset.



### 6.2.1 Distribution of the state

This section shows that the probability distribution of the chip magnitude of the components of the state observation is a binomial distribution. Therefore, the sample means estimator is the optimal estimator for the state.

Consider  $K$  state observations collected at a point on the receive fiber between two state transitions, *i.e.*, between consecutive packet arrivals or departure. The state at the point at every state measurement instant is the same, say  $S$ . The state observations may be different from each other because of ON-OFF keying.

Consider the  $K$  observations of the first chip of each state observation. These *chip observations* form a time series of  $K$  samples. Consider the first chip of the corresponding state  $S$ . Assume that first chip of the state  $S$  has chip magnitude  $m$ . The chip observation is the sum of the  $m$  chips, each of which is turned ON or OFF with probability  $p = 0.5$ <sup>1</sup>. Therefore, the chip observations may be considered to be the number of successes in a series of  $m$  Bernoulli trials. Therefore, the distribution of the chip magnitude of the chip observations is a Binomial distribution. The mean number of successes of a binomial distribution of  $m$  trials where  $p$  is probability of success is  $\mu = mp$ . The distribution is stationary between state transitions. The mean of a stationary binomial distribution can be estimated by using a *sample means* estimator. The *sample means estimator* is the estimator for the mean that minimizes the estimation error for the binomial distribution. It is a minimum variance, unbiased maximum likelihood estimator. It can be shown to converge to the mean as the number of observations

---

<sup>1</sup>This argument assumes that the packet data has 0 and 1 bits with equal probability ( $p = 0.5$ ). If the probability is different (say, due to the use of a higher layer encoding such as 4B/5B), then the value of  $p$  must be changed accordingly. The higher layer encoding may be chosen to ensure a particular value of  $p$ .

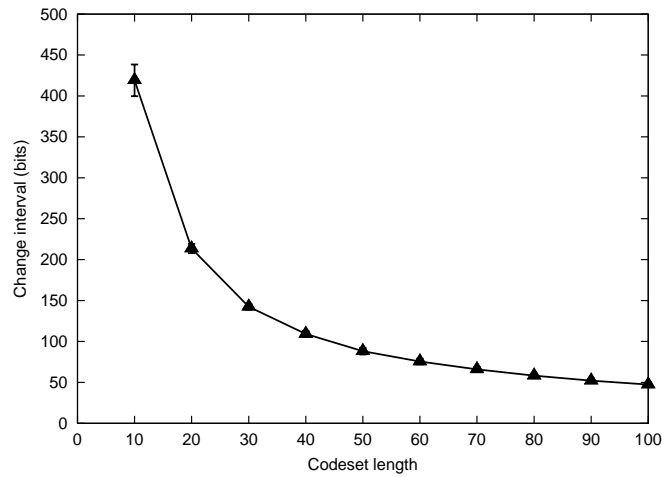


Figure 6.3: The mean number of bits between a state change for different codeword lengths at an offered load of 1. The network is an Aloha-CDMA network with Poisson arrivals and exponentially distributed packet lengths. The average packet length is 1000 bytes.

increases. If the estimated mean is  $\mu_e$ , then  $\mu_e = mp$ , where  $p = 0.5$ , *i.e.*,  $m = 2\mu_e$ . The procedure can be applied to the observations of each element and the estimated state vector can be calculated.

Figure 6.2 shows three state observations [11113001], [22202011] and [00112000]. The chip observations from the first chip of each of the state observations are 1, 2, 0. In Figure 6.2, the state is [22213011]. The value of the element in the first chip is 2. So in Figure 6.2, (1, 2, 0) are observations of the number of successes in a Bernoulli trials with  $m = 2$ . The mean value of the observations is  $(1 + 2 + 0)/3 = 1$ . Therefore, the estimated value is  $2\mu_e = 2 * 1 = 2$ . The same can be repeated for every element of the state vector. The estimated state vector in Figure 6.2 is [22315011].

The sample means estimator is the optimal estimator if the distribution is stationary, *i.e.*, the state does not transition during the collection of observations. A state transition (an arrival

or a departure of a packet) may change the mean of the distribution of the chip observations of the state observations. The sample means estimator needs enough observations before the estimates approaches the mean value. If the time needed to collect observations is comparable to the interval between state transitions then the estimation error increases because the distribution is no longer stationary <sup>1</sup>. Therefore, to justify the sample means estimator, it must be shown that the interval between state changes is larger than the time needed to collect sufficient observations.

The sample means estimator may be justified by a simple experiment that determines the interval between state changes. Consider an optical CDMA LAN using Aloha CDMA with Poisson arrivals and exponentially distributed packet lengths. The average packet length is 1000 bytes. The normalized offered load is 1. Figure 6.3 shows the average interval (in bits) between changes of the state of the line at a point. The interval between state changes is shown for different codeset lengths. Note that the graph shows the interval between state changes in terms of bit transmission times, and not in absolute time. This is intentional: the time to collect a single observation of the state depends on the codeset length (bit length). In one bit time, 1 observation of the state can be collected. Therefore, the time between state changes in bit times gives a measure of how slow the state is changing in comparison to the observation collection time. Even for codeset lengths of 100, state changes happen at most every 50 bits. For transmission scheduling algorithms, which allow around 1/3 as many

---

<sup>1</sup>A nonstationary distribution where the mean changes at a rate comparable to the rate at which observations are collected or which has a high variance is best estimated using a smoothed average rather than a sample means. For example, the Round Trip Time (RTT) estimator for the Transmission Control Protocol (TCP) uses an exponentially smoothed estimator because the rate at the which the RTT varies is comparable to the rate at which observations are collected.

packets on the line (for a codeset of weight 3) at 100% offered load, the interval between state changes is 2-3 times larger. The following sections show that a sample means estimator needs around 10-20 observations to obtain a reasonable estimate of the state. Therefore, there are very few state changes during collection of the set of observations. Under these conditions, the sample means is a reasonable estimator<sup>2</sup>.

## 6.2.2 Implementation

The state estimation hardware consists of an *observation module* and an *estimation module*. The observation module collects observations and the estimation module uses the observations to estimate the state. Chapter 7 describes the implementation. This section describes the observation and estimation parameters.

### 6.2.2.1 Observation parameters

The three parameters that control observations are

- Observation start time  $t_s$ : The time when the observation starts.
- Observation count  $n_s$ : The number of state observations collected.
- Buffer size  $b_s$ : The number of state observations that can be stored. The buffer is assumed to be a FIFO queue.

For example, if  $t_s = 0$  and  $n_s = \infty$ , collecting observations begins when the node is switched on and the node samples the state continuously. If  $t_s = t_a$  and  $n_s = b_s$ , sampling

---

<sup>2</sup>Note that these conditions may change depending on the traffic model. A different traffic model may change the interval between state transitions and may require a different estimator. Section 6.5 studies the performance of state estimation under realistic traffic and finds that the performance does not degrade noticeably.

begins when a packet arrives and ends when  $b_s$  samples have been collected. State samples are stored in a circular buffer of size  $b_s$  state samples. When the buffer is full, the oldest values are overwritten. It follows that the following relationship must hold:  $b_s \geq n_s$ .

### 6.2.2.2 Estimation parameters

The three parameters that control sampling are

- Estimation start time  $t_e$ : The time when the estimation algorithm is run.
- Estimation count  $n_e$ : The number of state samples used by the estimation algorithm.
- Inter-estimation time  $\delta_e$ : The time between successive runs of the state estimation algorithm.

Estimation can be started at any time when there are at least  $n_e$  state samples available.

The estimation algorithm takes as input the instantaneous samples in the buffer (at least  $n_s$  samples). It follows that the following relationship must hold:  $b_s \geq n_s \geq n_e$ . The estimation algorithm is shown in Table 6.1. The computation complexity of the estimation algorithm is  $O(n_e)$ . Therefore, the latency of the estimation algorithm is controlled by  $n_e$ .

### 6.2.3 State estimation algorithms

The sampling and estimation parameters control when the state estimate becomes available for use by the transmission scheduling algorithm. Transmission scheduling begins any time after a packet arrives and the state estimate is available

```

estimate()
{
  p ← 0.5
  eststate ← 0
  totalstate ← 0
  for count = 0 to ne
    totalstate += buffer[count]
  eststate ← totalstate * / (pne)
}

```

Table 6.1: The *Sample means* state estimation algorithm.

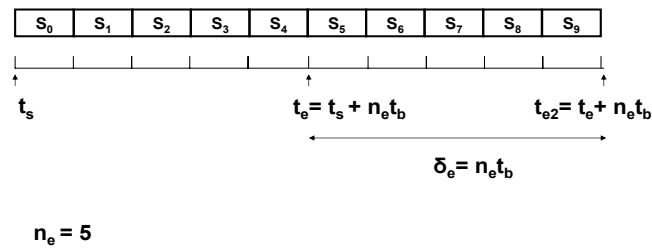


Figure 6.4: The state estimation parameters. The figure shows 10 state observations. The collection of state observations begins at  $t_s$ . Estimation is run after  $n_e = 5$  observations are collected at time  $t_e$ .

The parameters control a trade off between the access latency and the state estimation accuracy. The access latency can be minimized if, on packet arrival, transmission scheduling is started immediately using the last calculated state estimate. This may result in an inaccurate state estimate because of possible state transitions in the time between the last state estimation and transmission scheduling.

The estimation accuracy can be maximized by using the state estimate as soon as it is available. A way to do this is to defer transmission scheduling until a new state estimate is available. However, this results in an increase in access latency.

To evaluate this trade off this work considers two algorithms:

- Continuous estimation:  $t_s = 0$ ;  $n_s = \infty$ ;  $t_e > n_e t_b$ ;  $t_e < t_{arr}$ ;  $\delta_e = t_b$ ;  $n_e = 10$  to  $100$ :

The collection of observations begins when the node starts up and are collected every bit time. Estimation begins anytime before the first packet arrives. It is repeated after  $n_e$  new observations are collected (it is assumed that the state estimation hardware can be designed to complete the state estimation function before the next  $n_e$  observations are collected). When a packet arrives for transmission, transmission scheduling uses the last estimated state. Continuous estimation minimizes access latency but uses a potentially less accurate state estimate.

- On demand estimation:  $t_s = t_{arr}$ ;  $n_s = n_e$ ;  $t_e = t_a + n_e t_b$ ;  $\delta_e = \delta_a$ ;  $n_e = 10$  to  $100$ : Observations are collected on packet arrival and  $n_e$  observations are collected. Estimation begins after observations are collected. When a packet arrives for transmission, observations are collected, state is estimated, and transmission scheduling uses the latest estimated state. On demand estimation uses the most accurate state estimate but has the maximum access latency.

Other values of the sampling and estimation parameters are possible. Section 6.4 shows that the parameters of the state estimation algorithm do not significantly affect the performance as long as  $n_e$  is sufficiently large and  $t_e$  is within a few bit times of the start of the

transmission scheduling. This is because the correlation of the system is high (high correlation of state over distances of around 1000m), and the state remains constant for periods on the order of 100s of bit transmission times. Therefore, using a state estimate that is 10s of bit transmission times old does not result in noticeable degradation.

### 6.3 Causes for packet loss

This section discusses the causes for packet errors when realistic state estimation is used. Under the assumption of perfect state estimation and normalized propagation delay  $a = 0$  the sole cause of interference is erroneous transmission scheduling. When realistic state estimation is used with transmission scheduling there are three sources of interference. The three causes are listed below <sup>1</sup>.

- Erroneous transmission scheduling.
- Collisions.
- Erroneous state estimation.

#### 6.3.1 Interference because of erroneous transmission scheduling

When a transmission scheduling algorithm (threshold, selfish, overlap scheduling) is used, a small fraction of the packets are lost due to interference. This is because the algorithms

---

<sup>1</sup>It is important to note that all these factors increase interference, *i.e.*, chip overlaps. However, the condition for a bit error is  $w$  chip overlaps. The  $w$  chip overlaps may be due to any combination of the causes mentioned above. Therefore, while interference may be attributed to the causes above, a particular packet loss cannot be attributed to any one particular cause.

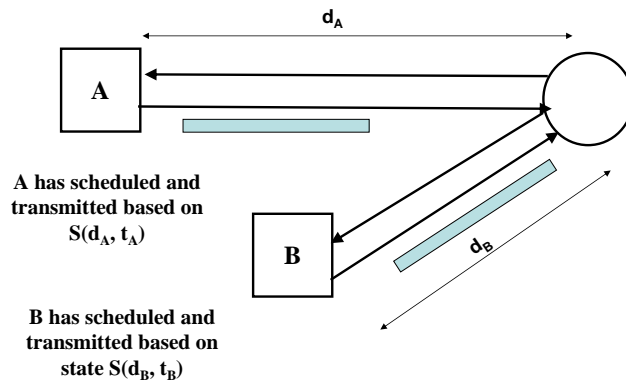


Figure 6.5: Collisions on an optical CDMA network. Transmissions from nodes A and B collide because neither can see the others' transmission.

are either *pseudo-cooperative* or *selfish*. These strategies do not guarantee that a node's transmission *Preserves self/Preserves others*. Chapter 5 showed a graph (Figure 5.5) of the probability of packet error. The graph shows how all the scheduling algorithms lose a fraction of their packets due to interference.

### 6.3.2 Interference because of collisions

A *collision* is an event where two or more nodes schedule their transmissions using state estimates that do not contain the codewords transmitted by each other. As a result, the transmissions of the nodes may interfere with each other. Another way of describing a collision is a state transition that occurs at the coupler in the interval between estimation of the state and

the merging of the transmitted packet at the coupler. A collision may or may not result in an interference error.

Consider two nodes located at different distances from the coupler, each estimating the state of the line. Consider two nodes A and B at distances  $d_A$  and  $d_B$  from the coupler where  $d_A > d_B$ . This is shown in Figure 6.5 Assume that node A estimates, schedules and transmits at time  $t_A$  and node B at  $t_B$  such that  $t_A < t_B$ . and  $t_B - t_A < d_A/c$ . Then both A and B have estimated and scheduled their transmissions without being aware of each other's transmissions.

As the average distance from the coupler increases the probability of collisions occurring increases. This is analogous to collisions on a single channel network and happens because of the separation of the nodes in space on the network.

### **6.3.3 Interference because of erroneous state estimation**

An error in state estimation can occur for two reasons:

- State transitions: A state transition during the collection of observations
- Low number of samples: The number of samples is not large enough. As a result, the estimate does not approach the mean.

Inaccurate state estimates can cause erroneous transmission scheduling.

## 6.4 Performance study

This section describes three experiments to quantify the effect of the three causes of error (erroneous transmission scheduling, collisions, and erroneous state estimation). Two parameters can be varied: the normalized propagation delay  $a$  ( $a = 0$  or  $a > 0$ ) and the state estimation algorithm (perfect or realistic). By varying them, the causes for interference can be isolated and the reduction in throughput due to the causes can be quantified.

The experiments and the causes that they isolate are:

- Perfect state estimation, normalized propagation delay  $a = 0$ : The nodes are at zero distance from the coupler and every node does perfect state estimation. Collisions or errors in state estimation are not possible. Therefore, interference is only due to erroneous transmission scheduling. This set of experiments is identical to the analysis in Chapter 5.
- Perfect state estimation,  $a > 0$ : Nodes do perfect state estimation at the estimation point. Therefore, there are no errors in state estimation due to a low number of samples. However, nodes are at different distances from the coupler. Therefore, collisions can occur. Therefore, interference is due to erroneous scheduling and collisions.
- Realistic state estimation,  $a > 0$ : Interference is due to erroneous scheduling, collisions and erroneous state estimation (both state transitions and low number of samples).

The results described in this section are simulation results. Unless explicitly stated the value of the parameters used in the simulation are the default values provided in Table 6.2

Parameter	Default value
Codeset parameters:	
Codeset length $N$	100
Number of wavelengths $A$	1
Codeset weight $w$	3
Maximum cross-correlation parameter $\kappa$	3
Number of codewords in codeset	100
Chipping rate:	10Gb/s
Codeword allocation:	Codewords are chosen uniform randomly from the codeset
Interference Avoidance parameters:	
Transmission scheduling algorithm:	Threshold scheduling
Threshold:	0.3
State estimation algorithm:	On demand state estimation
Window:	10 <i>bits</i>
Traffic parameters:	
Inter-arrival time distribution	Exponential
Normalized offered load	1
Packet size distribution	Exponential
Average packet size	1000 bytes
Destination address distribution:	Destinations are chosen uniform randomly from all the nodes
Topology parameters:	
Node to coupler distance distribution	Uniform
Average node to coupler distance	1000 m
Number of nodes	100

Table 6.2: Parameter list and default values for the state estimation performance and sensitivity study.

#### 6.4.1 Perfect state estimation, $a = 0$

Figures 6.6, 6.7 and 6.8 show the results for perfect state estimation, equidistant nodes for different codeset lengths ( $N = 10, 100, 200$ ). These results are identical to those in the previous chapter. The throughput of the threshold algorithm is slightly lower than indicated in

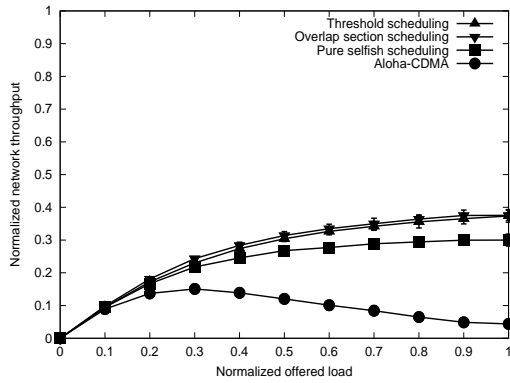


Figure 6.6: Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay  $a = 0$ . The results are based on simulation. The codeset length is 10. All other parameters are as specified in Table 6.2

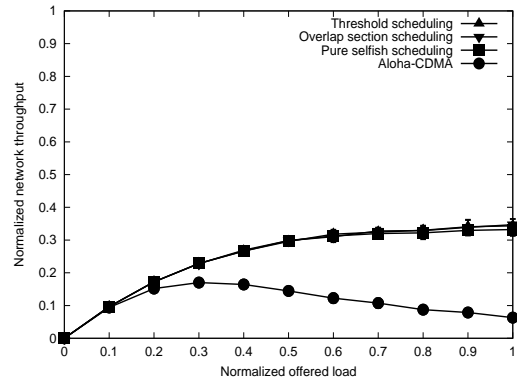


Figure 6.7: Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay  $a = 0$ . The results are based on simulation. The codeset length is 100. All other parameters are as specified in Table 6.2

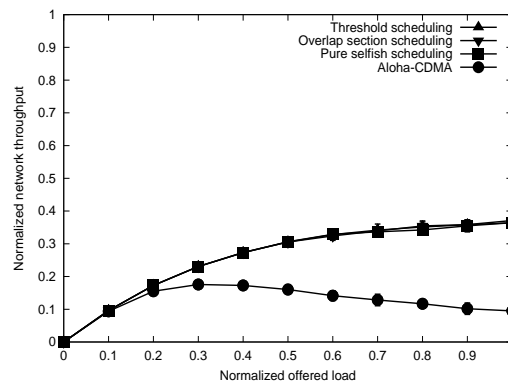


Figure 6.8: Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay  $a = 0$ . The results are based on simulation. The codeset length is 200. All other parameters are as specified in Table 6.2

the previous chapter because the threshold parameter is slightly lower (0.3). The important conclusion is that under perfect state estimation and equidistant nodes, all algorithms perform approximately the same and all prevent throughput degradation.

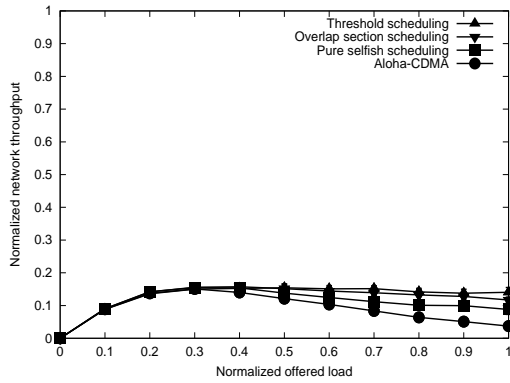


Figure 6.9: Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 10. All other parameters are as specified in Table 6.2

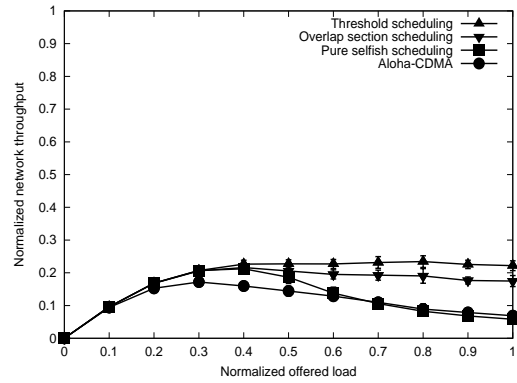


Figure 6.10: Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 100. All other parameters are as specified in Table 6.2

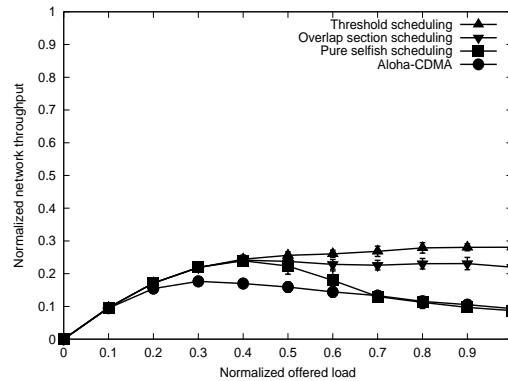


Figure 6.11: Normalized network throughput vs. normalized offered load for perfect state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 200. All other parameters are as specified in Table 6.2

#### 6.4.2 Perfect state estimation, $a > 0$

Figures 6.9, 6.10 and 6.11 show the throughput curves for perfect state estimation with codesets of length  $N = 10, 100$  and  $200$ . The nodes are located at an average distance of 1000m

from the coupler. The distances are uniformly distributed. The graphs show two clear differences from the results in the previous section: Firstly, selfish and overlap section scheduling have lower throughput at high offered load than threshold scheduling. Throughput degrades more rapidly for pure selfish and overlap length scheduling than for threshold scheduling. The threshold setting for these experiments is 0.3. Because of this, the number of code-words on the line for threshold scheduling is low compared to the other algorithms. Both the selfish and overlap scheduling aggressively transmit packets. When nodes are located at different distances from the coupler, collisions can occur. When collisions are possible, it is better to send less aggressively and lower the probability of collisions. This is the reason why threshold scheduling with a low threshold value performs better. The second difference is that the throughput at high offered load of threshold scheduling is best for long codeset lengths ( $N=200$ ). As mentioned in Chapter 3, this improves correlation between the estimated state and the state. In addition, as the codeset length increases, the probability that colliding packets affect the same chip reduces. This leads to the first three conclusions of this study:

*The throughput of the threshold transmission scheduling is greater than overlap and selfish scheduling. The throughput of all three transmission scheduling algorithms exceeds that of Aloha-CDMA.*

*When nodes are different distances from the coupler, collisions can occur. Collisions increase interference. To compensate, nodes must transmit conservatively. In particular, threshold transmission scheduling with a low threshold value is suitable.*

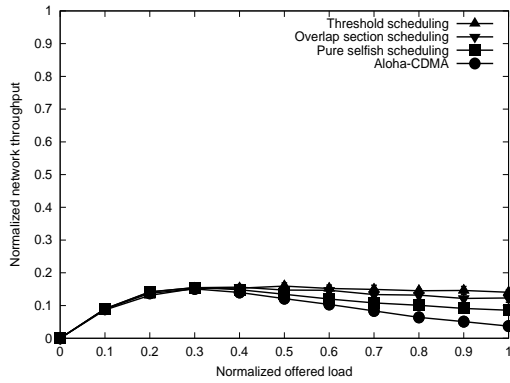


Figure 6.12: Normalized network throughput vs. normalized offered load for continuous state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 10. All other parameters are as specified in Table 6.2

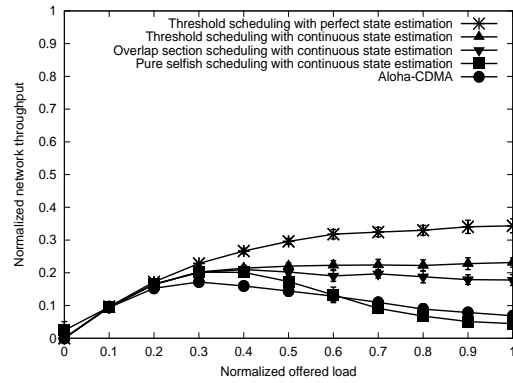


Figure 6.13: Normalized network throughput vs. normalized offered load for continuous state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 100. All other parameters are as specified in Table 6.2

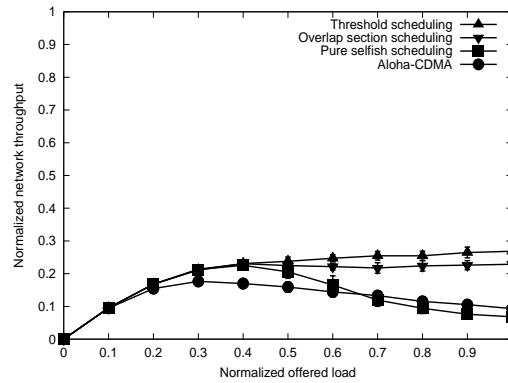


Figure 6.14: Normalized network throughput vs. normalized offered load for continuous state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The results are based on simulation. The codeset length is 200. All other parameters are as specified in Table 6.2

*Longer codeset lengths increase correlation and reduce the number of packets lost because of collisions, and therefore improve throughput.*

### 6.4.3 Realistic state estimation, $a > 0$

Figures 6.12, 6.13 and 6.14 show the throughput curves for continuous state estimation with nodes located at different distances from the coupler (uniformly distributed with an average distance of 1000m). The graphs show two differences from the results in the previous sections. Firstly, the normalized throughput at high offered load of all algorithms has decreased from the results in the previous section. The reason for this is the error in state estimation. Increasing the codeset length increases the correlation thereby reduces the error in state estimation. This is shown in Figure 6.15 where the throughput at normalized offered load of 1 increases with increase in  $N$ . This leads to the next two conclusions of this study:

*Errors in state estimation result in a degradation in throughput. Longer codeset lengths can help compensate for this effect.*

## 6.5 Sensitivity analysis

This section studies the performance of Interference Avoidance under realistic state estimation when codeset, MAC and traffic parameters are varied. All results are based on simulation. Unless explicitly specified the default parameters are as specified in Table 6.2. Each point is the average of 10 simulation runs. Standard deviations are indicated on the graphs. The sensitivity analysis consisted of quantifying the:

- Effect of varying the codeset length.
- Effect of varying the codeset weight.
- Effect of varying the cross correlation parameter.

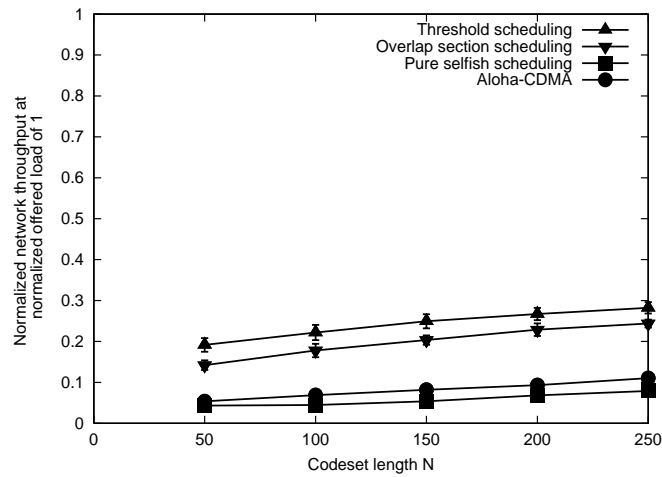


Figure 6.15: Normalized network throughput vs. codeset length  $N$  for different transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The code-set is  $(N, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2

- Effect of varying the number of wavelengths.
- Effect of varying the threshold transmission scheduling parameter.
- Effect of varying the number of state observations collected.
- Effect of different state estimation parameters
- Effect of different packet size distributions.
- Performance under realistic network traffic.
- Performance under very high loads.

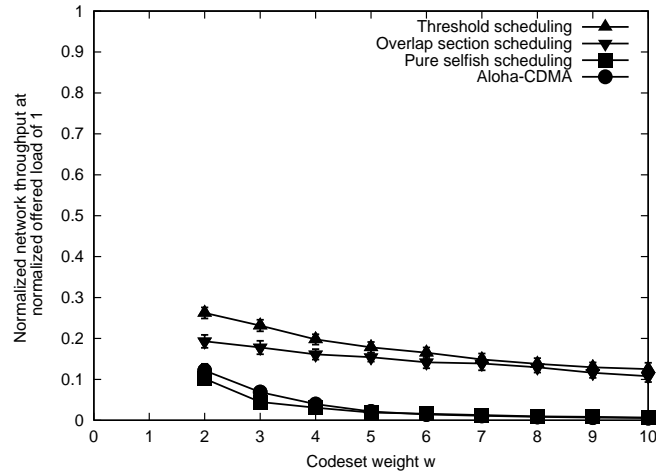


Figure 6.16: Normalized network throughput vs. codeset weight  $w$  for different transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The code-set is  $(100, w, w)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2

### 6.5.1 Effect of varying the codeset length

Figure 6.15 shows the effect of varying the codeset length  $N$ . Chapter 5 showed that with perfect state estimation and  $a = 0$ , the codeset length had no effect on transmission scheduling. An increase in codeset length allowed more codewords to be multiplexed on the line. However, it also reduced the data rate. The two effects balanced each other resulting in no change in throughput. When  $a > 0$ , the throughput falls due to collisions and erroneous state estimation. When  $N$  increases, throughput improves. As discussed in the previous section this effect is because of the improved correlation. The magnitude of this improvement can be seen in the graph.

### 6.5.2 Effect of varying the codeset weight

Figure 6.16 shows the effect of varying the codeset length weight  $w$ . Chapter 5 discussed the effect of varying  $w$  under perfect state estimation. As the weight of the codeset increased, the throughput degraded. This was due to the increase in interference between codewords and the difficulty in packing more codewords on the line. When  $a > 0$  and realistic state estimation is used, throughput degrades further due to the collisions and erroneous state estimation.

### 6.5.3 Effect of varying the cross correlation parameter

The cross correlation parameter  $\kappa$  controls the interference between pairs of codewords. Chapter 5 discussed the effect of  $\kappa$  on codeset design. Figure 6.17 shows the performance of the transmission scheduling algorithms under continuous state estimation for a codeset of length 42, weight 6 as  $\kappa$  is varied from 1 to 6. As may be seen the throughput does not vary significantly with  $\kappa$ . Note that the throughput in the figure is significantly lower than the throughput shown in previous performance results. This is due to the use of a codeset of  $N = 42, w = 6$ .

### 6.5.4 Effect of varying the number of wavelengths

Figure 6.18 shows the effect of varying the number of wavelengths on the network throughput. As the number of wavelengths is increased, the overall network throughput remains around the same. Under perfect state estimation and  $a = 0$ , increasing the number of wavelengths has no effect on throughput. The capacity of the system was increased, but the users

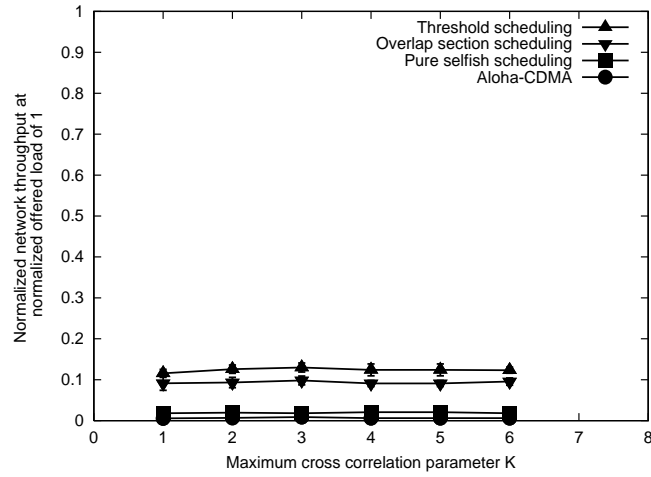


Figure 6.17: Normalized network throughput at normalized offered load of 1 for different transmission scheduling algorithms. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset length  $N = 42$  and weight  $w = 6$ . The cross correlation parameter  $\kappa$  is varied from 1 to 6. For the threshold scheduling algorithm, the threshold parameter was set to 0.5

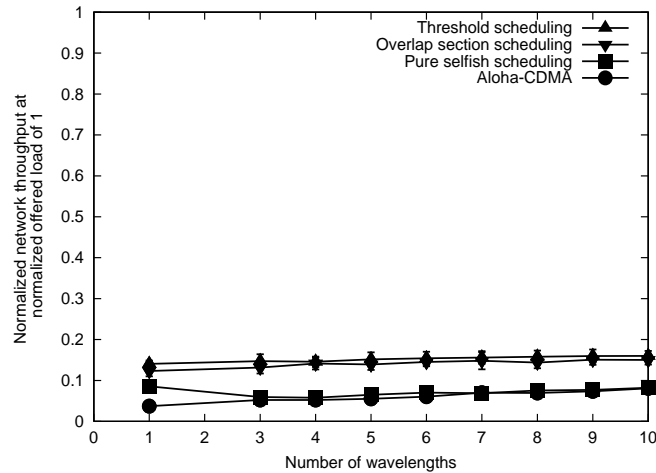


Figure 6.18: Normalized network throughput vs. number of wavelength for different transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is  $(10, 3, 3)$ . For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2

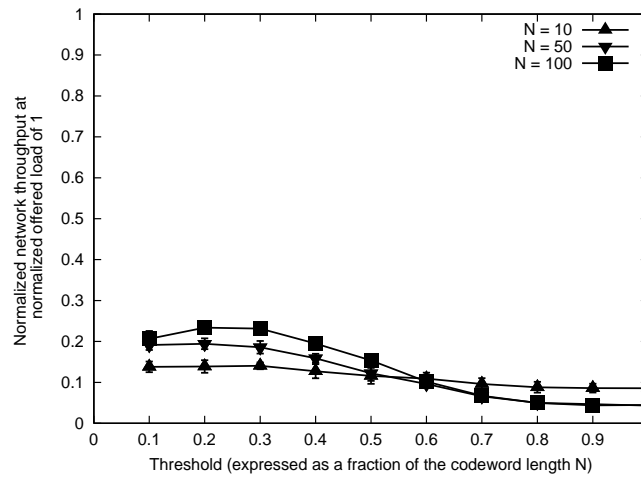


Figure 6.19: Normalized network throughput vs. threshold for the threshold scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset lengths are 10, 100 and 200. All other parameters are specified in Table 6.2

could use only the same fraction of the capacity, so overall network throughput was around the same. There is no change when  $a > 0$  and realistic state estimation is used.

### 6.5.5 Effect of varying threshold

Figure 6.19 shows the effect of varying the threshold on network throughput. The results in the previous chapter (for perfect state estimation,  $a = 0$ ) showed that there is an optimal threshold, although network throughput at the optimal threshold was not significantly higher than at other values. Two interesting differences can be seen in these results:

- The peak network throughput occurs at a lower threshold, *i.e.*, the optimal threshold is lower.
- At higher thresholds there is a degradation in throughput.

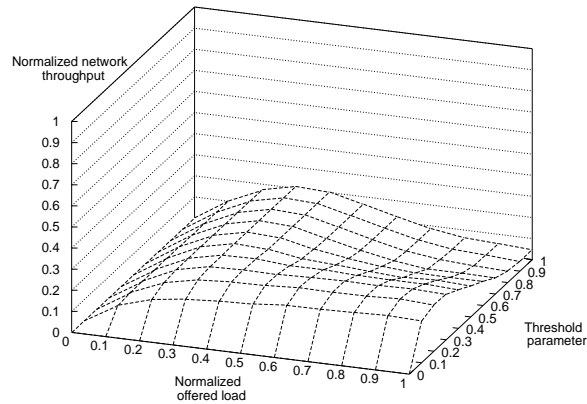


Figure 6.20: Normalized network throughput vs. threshold vs. offered load for the threshold scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The code-set length is 100. All other parameters are specified in Table 6.2

The reason for both these effects is the new sources of interference that are introduced: collisions and erroneous state estimation. In the presence of both these effects, it is better for a node to transmit conservatively, resulting in a lower number of codewords multiplexed on the line. This reduces the packets lost due to interference from collisions and erroneous state estimation and this improves the throughput. Figure 6.19 shows the normalized throughput for different threshold parameter values.

### 6.5.6 Effect of different estimation algorithms

Figure 6.21 shows the throughput curves for on-demand state estimation for  $N = 200$ . Comparing with Figure 6.13 indicates that there is no difference between the performance of on-demand and continuous state estimation. This is because small windows (10 bits) are

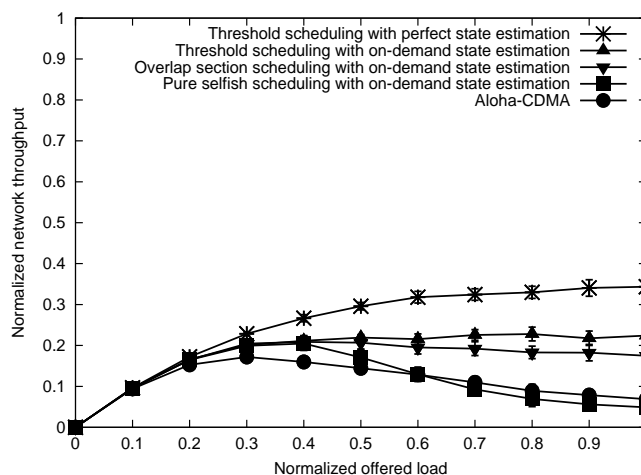


Figure 6.21: Normalized network throughput vs. normalized offered load for on demand state estimation, normalized propagation delay  $a > 0$  (average distance from the coupler is 1000m). The codeset length is  $N = 200$ . The results are based on simulation. All other parameters are specified in Table 6.2

sufficient for state estimation. Therefore the delay is negligible when compared to the state estimation time. Continuous state estimation performs the same as on demand state estimation.

### 6.5.7 Effect of varying the number of state observations collected

Figure 6.22 shows the effect of varying the number of state observations collected (*window*) on the throughput. When the window is above 10 bits, it does not have an appreciable difference in improving the throughput. However, the correct window size is dependent on the codeset length. As  $N$  increases, the transmission scheduling algorithm multiplexes a larger

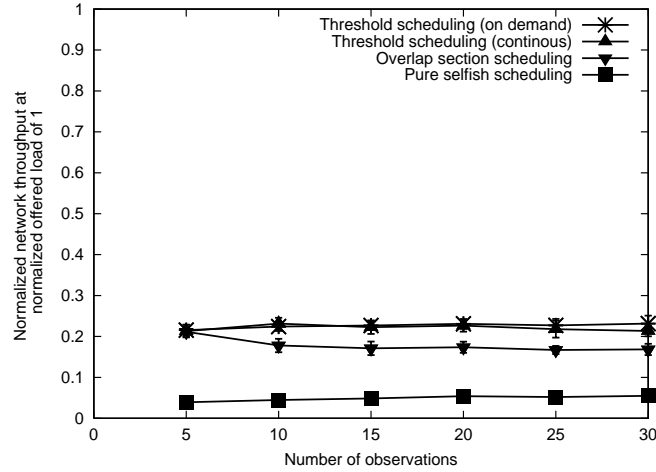


Figure 6.22: Normalized network throughput vs. state estimation window (in bits) for the transmission scheduling algorithms and continuous and on-demand state estimation. The results are based on simulation. All other parameters are specified in Table 6.2

number of codewords on the fiber. For larger  $N$ , there is a larger difference in the state observations because of ON-OFF keying. Therefore, the window needs to be chosen depending on the codeset length. For codeset lengths of up to 100, a window of 10 is sufficient.

### 6.5.8 Effect of varying the average distance from the coupler

Figures 6.23, 6.24 and 6.25 show the effect of increasing distance on the throughput. As explained earlier, the effect of increasing distance is to increase the probability of collisions.

The results show that this effect reduces for larger values of  $N$ . The graph also shows that at zero distance, the throughput of selfish and overlap scheduling is better for  $N = 10$ . The reason for this is that as  $N$  increases, the differences between state observations increases and more observations are needed for accurate state estimation. The selfish and overlap section allow a large number of codewords on the line. Therefore, the error in state estimation is

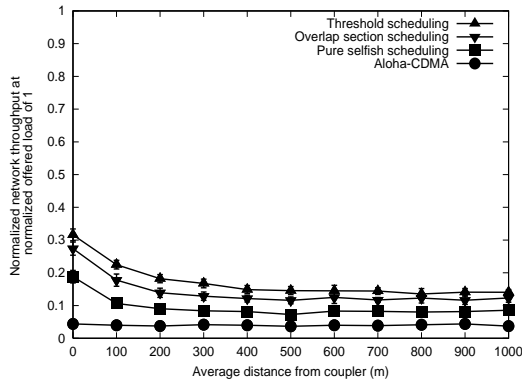


Figure 6.23: Normalized network throughput vs. average distance from coupler (uniform distribution) for the transmission scheduling algorithms and continuous state estimation. The results are based on simulation. The codeset length is 10. All other parameters are specified in Table 6.2

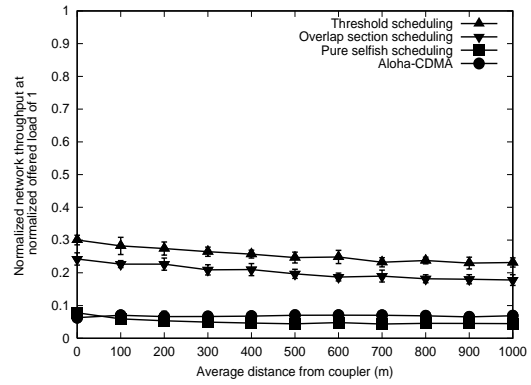


Figure 6.24: Normalized throughput vs. average distance from coupler (uniform distribution) for the transmission scheduling algorithms and continuous state estimation. The results are based on simulation. The codeset length is 100. All other parameters are specified in Table 6.2

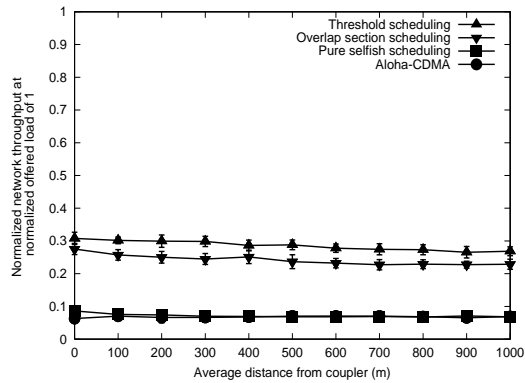


Figure 6.25: Normalized throughput vs. average distance from coupler (uniform distribution) for the transmission scheduling algorithms and continuous state estimation. The results are based on simulation. The codeset length is 200. All other parameters are specified in Table 6.2

large for these algorithms. The threshold scheduling algorithm allows a lower number of codewords on the line (due to the low threshold of 0.3). Hence this effect is less pronounced.

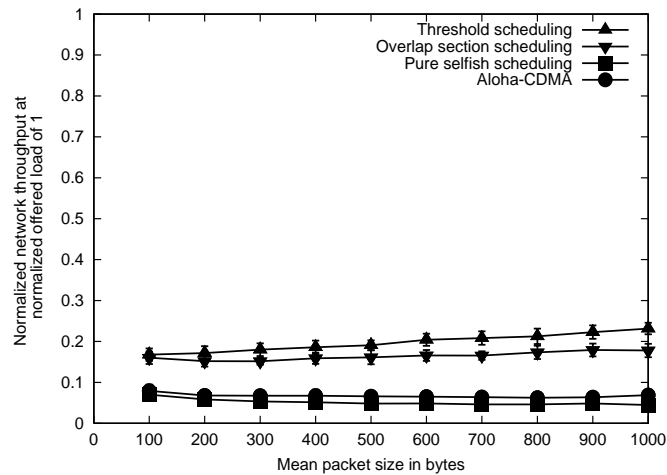


Figure 6.26: Normalized network throughput vs. average packet length (in bytes) for the transmission scheduling algorithms and on-demand state estimation. The results are based on simulation. The traffic model is Poisson arrivals with exponentially distributed packet lengths. The codeset is (100,3,3). For the threshold scheduling algorithm, the threshold parameter was set to 0.3. All other parameters are specified in Table 6.2

### 6.5.9 Effect of varying the packet length

Figure 6.26 shows the effect of different packet sizes on network throughput. When nodes are distributed on the line and they use realistic state estimation, the effect of varying packet length is similar to the effect of increasing  $N$ . With large packets sizes, the packets remain on the line much longer. This improves correlation, resulting in an improvement in throughput.

### 6.5.10 Performance with real network traffic

Figure 6.27 shows the performance with realistic network traffic. As before, a traffic trace from an OC48 link was used. To generate the required load several traces were merged. The packet sizes, source address, destination address were preserved during merging. The average packet size of the resulting traffic file was 500 bytes. The trace file had around

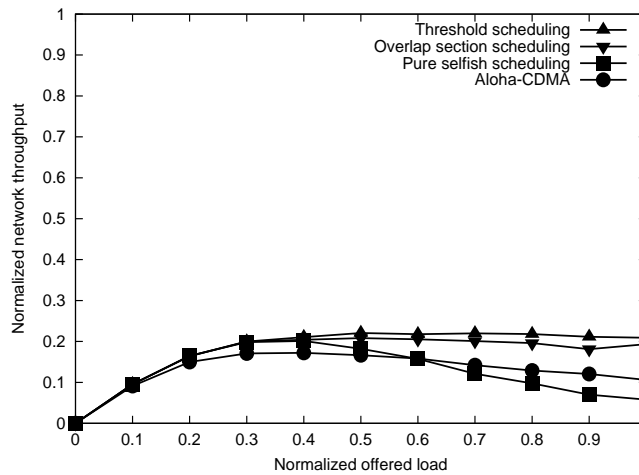


Figure 6.27: Normalized network throughput vs. normalized offered load for the transmission scheduling algorithms and continuous state estimation with realistic network traffic. The results are based on simulation. All other parameters are specified in Table 6.2

6000 unique source addresses and 6000 unique destination addresses. Care was taken during merging to ensure that the traffic of the appropriate load was generated. In contrast to all the previously described results, in this case destination addresses were mapped to codewords. Where codewords were insufficient, codeword sharing was used. The results show a slight improvement in network throughput. The packet squeeze through effect described in the previous chapter increases throughput.

### 6.5.11 Performance under very high load

Figures 6.28 and 6.29 show the performance of Interference Avoidance under very high load. The results show that eventually, at load between ten and a hundred times the transmission capacity, the throughput of Interference Avoidance will collapse. The collapse occurs because collisions increase as the load increases. However, the load at which Interference Avoidance

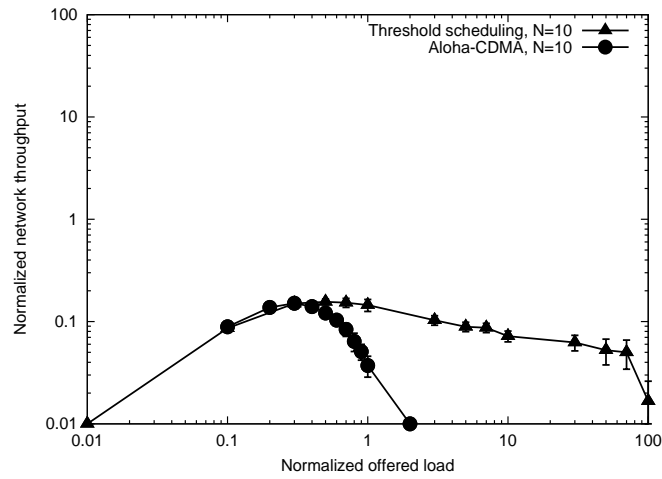


Figure 6.28: Normalized network throughput vs. normalized offered load for the threshold scheduling transmission scheduling algorithms and continuous state estimation with Poisson arrivals, exponentially distributed packet sizes. The codeset length is 10. The results are based on simulation. All other parameters are specified in Table 6.2

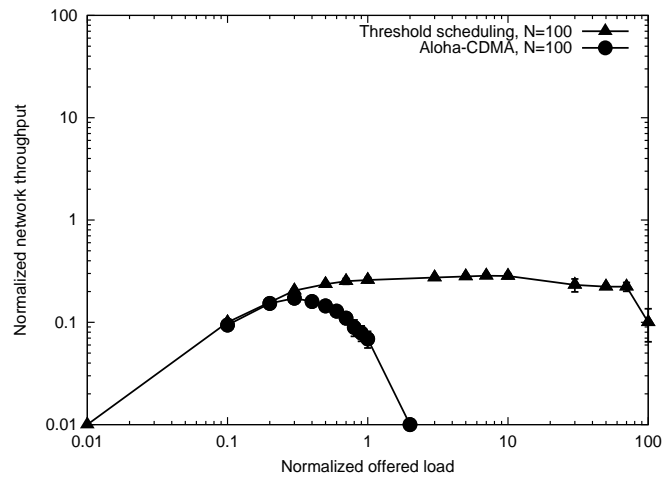


Figure 6.29: Normalized network throughput vs. normalized offered load for the threshold scheduling transmission scheduling algorithms and continuous state estimation with Poisson arrivals, exponentially distributed packet sizes. The codeset length is 100. The results are based on simulation. All other parameters are specified in Table 6.2

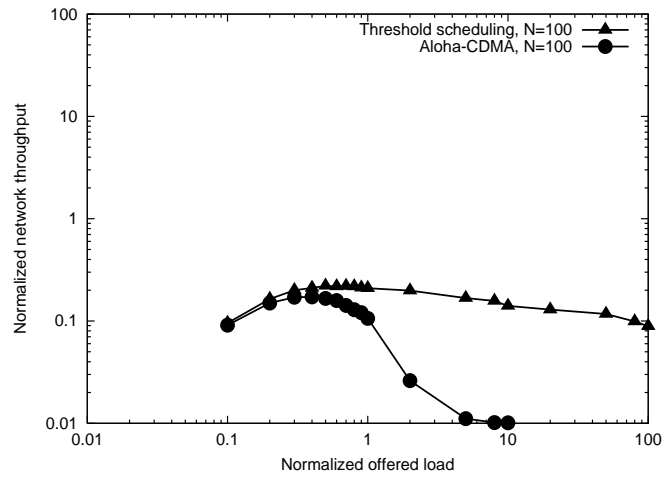


Figure 6.30: Normalized network throughput vs. normalized offered load for the threshold scheduling transmission scheduling algorithms and continuous state estimation with real traffic. The codeset length is 100. The results are based on simulation. All other parameters are specified in Table 6.2

collapses depends on the codeset length. As described in the previous sections, the effect of collisions reduces as the codeset length increases. The graphs in the figures indicate that for codeset lengths of around 100, collapse can be prevented upto loads of around 100.

Finally, Figure 6.30 shows the performance of Interference Avoidance under heavy loads with realistic traffic. It shows the normalized throughput vs. normalized offered load curve for threshold scheduling with continuous state estimation. The average distance between the nodes and the coupler is 1000m and the codeset length is 100. All other parameters are specified in Table 6.2. The graph indicates that Interference Avoidance, with suitable codeset parameters, can be used in a real network, even at very high offered load.

## 6.6 Conclusions and future work

This study has indicated that the performance of state estimation depends on several parameters. In particular, as the distance of the nodes from the coupler increases, the throughput falls due to collisions and inaccurate state estimation. By increasing the codeset length and decreasing the codeset weight, the performance of realistic state estimation tends to perfect state estimation.

An open area of research is the tuning of the state estimation algorithms to specific network traffic distributions by using more complex estimators. The objective is to design estimators that approach the bound of perfect state estimation for low values of the codeset length.

## **Chapter 7**

### **Implementation**

This chapter describes an experimental evaluation of Interference Avoidance and how it may be implemented. Section 7.1 describes a laboratory testbed for optical CDMA experimentation. The testbed was used to measure the performance of transmission scheduling. The testbed was set up at the USC Optical Communication Laboratory by members of the Optical Communications Laboratory. The transmission scheduling experiments were carried out by the author. Results from the measurements show that in the presence of physical layer noise, transmission scheduling can provide up to 2 orders of magnitude improvement in Bit Error Rate (BER). Section 7.2 describes the design and operation of a Network Interface Card (NIC) that implements Interference Avoidance over optical CDMA.

#### **7.1 Experimental demonstration and measurement of transmission scheduling**

The optical CDMA testbed is a network with 4 nodes (also called *users*). A simplified diagram of the network is shown in Figure 7.1. The testbed has a transmitter component and a

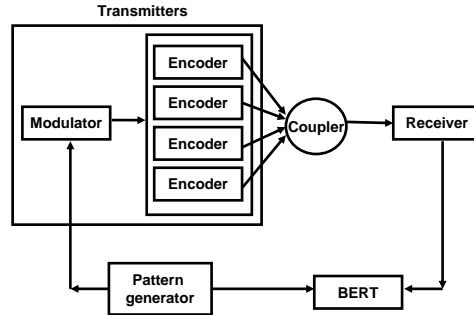


Figure 7.1: A block diagram of the optical CDMA experimental testbed constructed at the USC Electrical Engineering department's Optical Communications Laboratory.

receiver component. The transmitter has one modulator and 4 encoders, one for each user. The 4 encoders transmit 4 different codewords ( $C1 - C4$ ). The testbed has only one receiver. The receiver receives the codeword  $C4$ , *i.e.*, the codeword for user 4. The transmitter and the receiver components are discussed in detail in the next two sections.

The optical CDMA codeset is a  $(11, 6, 6, 1)$  codeset, *i.e.*, the codeword length  $N$  is 11 chips, the number of wavelengths  $\Lambda$  is 6, the weight  $w$  is 6 and maximum cross-correlation parameter  $\kappa$  is 1. The chipping rate used is 11Gc/s. The data rate is 1Gb/s. The codewords are shown in Table 7.1.

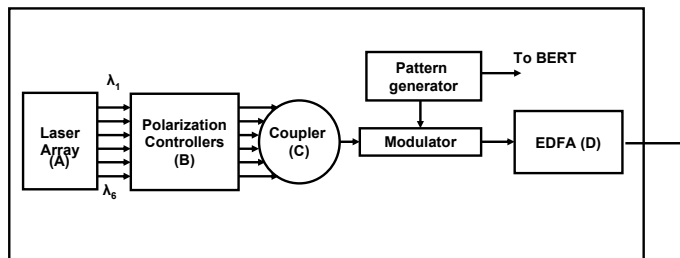


Figure 7.2: A block diagram of the modulator part of the Optical CDMA testbed.

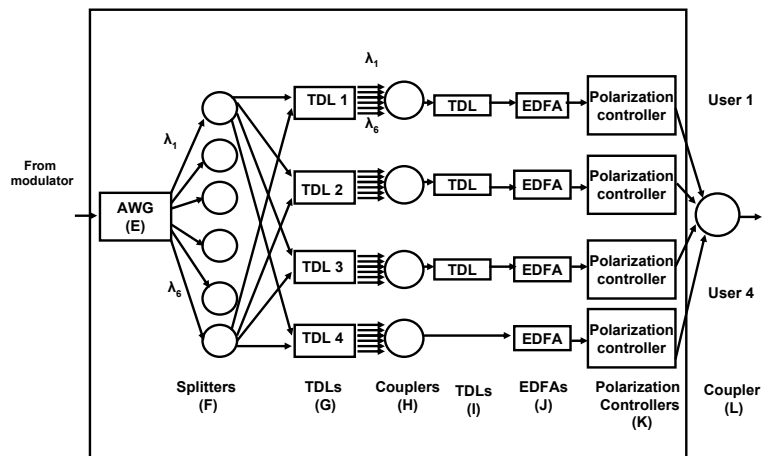


Figure 7.3: A block diagram of the four encoders of the Optical CDMA testbed.

User 1	$\lambda_1$	1 0 0 0 0 0 0 0 0 0 0 0
	$\lambda_2$	0 0 0 0 0 1 0 0 0 0 0 0
	$\lambda_3$	0 0 0 0 0 0 0 0 0 0 0 1
	$\lambda_4$	0 0 0 0 0 1 0 0 0 0 0 0
	$\lambda_5$	0 0 1 0 0 0 0 0 0 0 0 0
	$\lambda_6$	0 0 0 0 0 0 0 1 0 0 0 0
User 2	$\lambda_1$	0 0 0 0 0 1 0 0 0 0 0 0
	$\lambda_2$	0 0 0 0 0 0 0 0 0 1 0 0
	$\lambda_3$	0 0 0 0 0 0 0 0 0 0 0 1
	$\lambda_4$	0 0 1 0 0 0 0 0 0 0 0 0
	$\lambda_5$	0 0 0 0 0 1 0 0 0 0 0 0
	$\lambda_6$	0 0 0 1 0 0 0 0 0 0 0 0
User 3	$\lambda_1$	0 0 0 0 0 0 1 0 0 0 0 0
	$\lambda_2$	0 0 0 0 1 0 0 0 0 0 0 0
	$\lambda_3$	0 0 0 0 0 0 0 0 0 0 0 1
	$\lambda_4$	0 0 0 0 0 0 0 0 0 1 0 0
	$\lambda_5$	0 0 0 0 0 0 0 0 0 0 1 0
	$\lambda_6$	1 0 0 0 0 0 0 0 0 0 0 0
User 4	$\lambda_1$	0 0 0 0 0 0 0 1 0 0 0 0
	$\lambda_2$	1 0 0 0 0 0 0 0 0 0 0 0
	$\lambda_3$	0 0 0 0 0 0 0 0 0 0 0 1
	$\lambda_4$	0 0 0 1 0 0 0 0 0 0 0 0
	$\lambda_5$	0 1 0 0 0 0 0 0 0 0 0 0
	$\lambda_6$	0 0 0 0 0 0 0 0 0 1 0 0

Table 7.1: Codewords for the 4 nodes in the experimental testbed. The codewords are from a 2 D codeset (more than one wavelengths is used) with parameters (11, 6, 6, 1).

### 7.1.1 Optical CDMA transmitter component

The transmitter component consists of one modulator and 4 encoders. A diagram of the modulator is shown in Figure 7.2. The function of the modulator is to generate the six wavelengths and modulate them with the data. Ideally, there should be one modulator for each user (encoder). However, four modulators could not be built for the four users because the

necessary equipment was not available. Therefore, a single modulator was used. The modulated data from the modulator is sent to the 4 encoders. The encoders encode the modulated wavelengths with the optical CDMA codewords. A diagram of the encoders is shown in Figure 7.3. The limitation of using one modulator for 4 encoders is that data generated for the four users is identical and bit synchronized. To eliminate the bit synchronization, fibers of different lengths and tunable delay lines give each user a random delay.

#### **7.1.1.1 Modulator**

The modulator consists of a laser array, polarization controllers, a coupler, the data modulator, and an Erbium Doped Fiber Amplifier (EDFA). Figure 7.2 shows a block diagram of the modulator.

- Laser array (A): The array of lasers generates the wavelengths for transmission. The six wavelengths are denoted by  $(\lambda_1 \dots \lambda_6)$ .
- Polarization controllers (B): The polarization controllers are used to adjust the polarizations of the wavelengths for maximum output power. The data modulator (described below) modulates with the maximum power gain when the wavelengths are at a particular polarization. By tuning the polarization controllers, the modulated signal can be adjusted to obtain maximum output power.
- Coupler (C): The coupler couples the wavelengths on to a single fiber. The coupler is used so that all the wavelengths can be modulated together.

- **Data modulator:** The data modulator modulates the incoming wavelengths. It is driven by a pattern generator. The pattern generator generates a sequence of bits. It operates at 11 Gb/s (the chipping rate). The pattern generator is configured to send a pattern of 0000000001 to represent a 1 bit. It sends a pattern of 0000000000 to represent a 0 bit. These two patterns are 11 chips long. Note that the pattern for a 1 bit has a single ‘1 chip’ and the pattern for a 0 bit has no ‘1 chips’. These patterns are used to generate the codeword. The reason for this pattern is explained in the next section.
- **Erbium Doped Fiber Amplifier EDFA (D):** The EDFA is used to amplify the signal before the encoders.

#### **7.1.1.2 Encoders**

The encoders encode the modulated data. The encoder consists of an Arrayed Waveguide (AWG), a splitter, tunable delay lines, couplers, tunable delay lines, EDFAs and polarization controllers. Figure 7.3 shows a block diagram of the encoders.

- **Arrayed Waveguide (AWG) (E):** The AWG splits the incoming signal into its composite wavelengths.
- **Splitters (F):** The six splitters split each wavelength into 4 signals. A group of six wavelengths is used to encode each user’s codeword.
- **Tunable delay lines (G):** The delay lines are used to encode the data. Each user has its own set of tunable delay lines. Figure 7.4 shows how it works. Assume the data modulator has sent a 1 bit. Then on each wavelength, the pattern 0000000001 enters

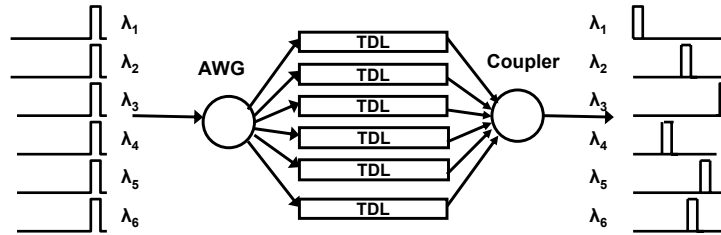


Figure 7.4: The optical CDMA encoders. The tunable delay lines (TDLs) of the encoder encode an optical CDMA codeword.

the delay lines. Each delay line delays the pulse (the '1 chip' in the pattern) by a different amount of time. The codeword for user 1 is shown in Table 7.1. For codeword 1 the delays for the wavelengths 1 to 6 are respectively 1, 7, 11, 6, 3 and 8. By carefully tuning the delay lines, the total delay of a path from the splitter F to the coupler H can be controlled to generate the appropriate delays. In the testbed, some delay lines are free space, some are fiber based and some are tunable. Note that the output of a single modulator enters all four encoders. This creates a problem: the data of all four users are identical and bit synchronized. Random delays are introduced after the encoders by using optical fibers of different lengths to eliminate the synchronization.

- Couplers (H): Each encoder has a coupler that couples the six wavelengths. The output of the coupler is the encoded data.

- Tunable delay lines (I): Tunable delay lines are used after the coupler to give random delays to each of the users. The tunable delay lines along with the different fiber lengths between the couplers (H) and the coupler L eliminate the bit synchronization of the codewords of the users.
- EDFA (J): The EDFAs are used to amplify the signals. The wavelengths undergo power attenuation due to loss at the free space delay lines. The EDFAs compensate for this.
- Polarization controllers (K): The polarization controllers are used to protect against beat noise [71]. ‘1 chip’ overlaps at the same time and wavelength causes beat noise. The beat noise can be avoided if the users’ wavelengths are on different polarizations. This eliminates the beat noise but does not affect the interference.
- Coupler (L): The coupler couples the encoded data from each user on to a single fiber. The signal at the output of the coupler is a multilevel optical signal equal to the sum of the multiplexed codewords.

### **7.1.2 Optical CDMA receiver**

The optical CDMA receiver is a correlation receiver. Note that it is not a hard-limiting receiver. The receiver decodes the codeword  $C_4$ . It consists of an arrayed waveguide, tunable delay lines, a coupler, an optical attenuator, a photodetector, an amplifier, an attenuator, and a threshold detector.

- Arrayed waveguide (N): The AWG splits the signal into its six wavelengths.
- Tunable delay lines (O): The tunable delay lines are tuned to the delays corresponding

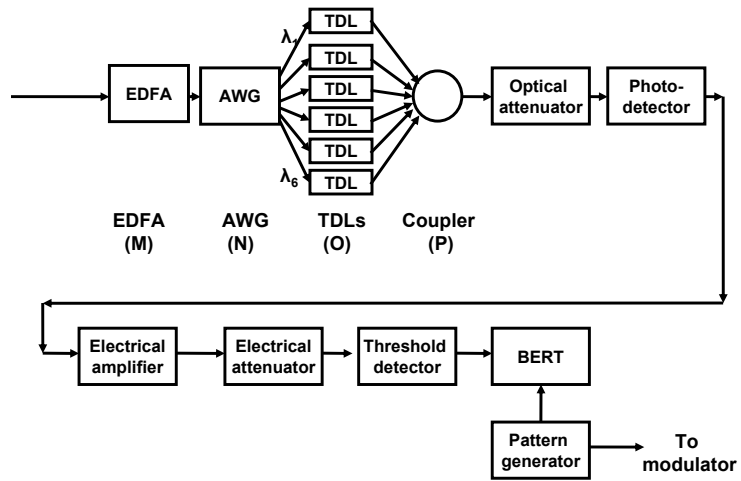


Figure 7.5: The receiver of the optical CDMA testbed.

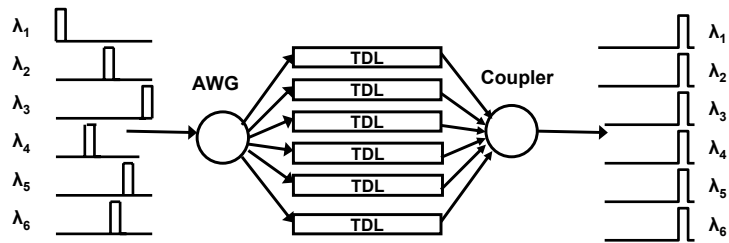


Figure 7.6: The decoder of the optical CDMA receiver. The figure shows how the Tunable Delay Lines (TDLs) of the receiver decode the optical CDMA codeword.

to a receiver for user 4. Figure 7.6 shows how the decoder works. The decoder reverses the operation of the encoder. For codeword *C4* the delays are 4, 11, 1, 8, 10, 3 chip times. The codeword enters the delay lines and at the output, the pulses are stacked up in time.

- Coupler (P): The signal from each of the wavelengths is coupled. At the output of the coupler, the '1 chips' of the codeword should be aligned in time.
- Optical attenuator: The optical attenuator attenuates the optical signal to a value within the photodetector's range.
- Photo detector: The photodetector converts the incident light energy to electrical power. The photodetector is a PIN diode.
- Attenuator: An attenuator is used to bring the electrical power down to within the threshold detector's range.
- Threshold detector: The threshold detector generates a pulse if the input is higher than the threshold.

### **7.1.3 Measurement instruments**

The measurement instruments are the pattern generator and the Bit Error Rate Tester (BERT).

- Pattern generator: The pattern generator generates a sequence of bits. As described earlier, to send a 1 data bit, the pattern 0000000001 is sent. To send a 0 data bit, the pattern 0000000000 is sent. The sequence is 1024 bits long.

- BERT: The Bit Error rate Tester (BERT) compares the input from the pattern generator to the detected signal and reports the bit error rate.

#### **7.1.4 Limitations of the testbed**

The experimental testbed has several limitations that makes it difficult to compare results to the theoretical work described in Chapters 5 and 6. Firstly, the limited number of components in the testbed meant that only a four user system could be tested. For the codeset (11, 6, 6, 1), a four user system is a very lightly loaded system. The normalized offered load is 0.06 (= (Number of codewords multiplexed on the line) / ( $\Lambda * N$ ) =  $4/(6*11)$ ). According to the theory of the previous chapters, transmission scheduling improves throughput only at high offered load.

Secondly, the system was noisy. Noise was introduced through several sources: the free space delay lines, the amplifiers, mode hopping of the lasers. Due to the noise, the receiver threshold for the receiving user had to be set to half the weight of the code. This is the optimal setting for detecting a codeword when only one user is transmitting under noisy conditions.

Thirdly, the specification of the power input to the receiver was low and therefore the power input had to be kept below a limit. This was done by a target amplifier that limited the input power to the receiver. Therefore, the input power to the receiver was the same whether there were 1 , 2, 3 or 4 users on the system. Therefore, when four users are transmitting, the power of each user's individual codeword is reduced. To compensate for this, the threshold had to be lowered each time a new user was added. The consequence of this is that the noise had a greater impact when the number of users was increased.

Lastly, the receiver was not a hard-limiting receiver. Therefore, noise and interference effects were greater than with a hard-limiting receiver.

The experiment shows that the last three limitations introduce enough noise that transmission scheduling resulted in a performance benefit (a reduction in BER) even at the low loads used in the experiment.

### **7.1.5 Objective of the experiment**

The objective of the experiment was to determine whether transmission scheduling provides any benefit under realistic conditions.

To do this the bit error rate was measured as the number of simultaneous users was increased. The transmission delays of the users were varied and the bit error rates were noted. The number of chip overlaps in each case was determined from knowledge of the codesets. The chip overlaps were also viewed on an oscilloscope connected to the output of coupler L. To evaluate the performance of transmission scheduling, the performance of a threshold scheduling algorithm with a low threshold of 1 overlap (threshold parameter = 0.015) was used. This threshold allows codewords to be transmitted only if there are no overlaps. The readings from the experiment and the number of chip overlaps for each case were used to determine the BER of threshold scheduling. Any reading with zero overlaps was considered to be a case of threshold scheduling with threshold of 0.015. This was compared against the worst case bit error rate and the average case bit error rate. The average case bit error rate can be considered to be the bit error rate of Aloha-CDMA.

## **7.1.6 Experimental procedure**

This section describes the experimental procedure. First, preliminary checks are performed to ensure that the system is functioning normally. These checks are made before taking any reading and during the experiment.

### **7.1.6.1 Preliminary checks**

First, the encoders are checked to determine whether the data is being encoded properly. Only one of the EDFAs (J) in Figure 7.3 is switched on. An oscilloscope and a spectrum analyzer are connected to the output of coupler L. The output is visually checked to determine whether all six wavelengths are present and that their powers are equal. The transmission delay of the codeword is checked to see if they are correct. These checks are performed for all 4 codewords.

Next, the threshold detector clock (trigger) is synchronized to the output of the photodetector. The EDFA (J) corresponding to user 4 is switched on. All other EDFAs are switched off. An oscilloscope is connected to the output of the photodetector, the threshold detector, and the clock input. The positive edge of the clock input to the threshold detector is aligned to the output of the photodetector.

### 7.1.6.2 Measurements

Three experiments were conducted. In each experiment, the receiver was tuned to the codeword of user 4. The codewords for users 1, 2, 3 were the interfering codewords. The experiments measured the bit error rate (BER) for user 4 in the presence of one, two, and three interfering users.

The steps were as follows:

- User 4 (the receiving user) is switched on using EDFA J.
- The receiver threshold is adjusted for the best possible BER.
- One user is added to the system by switching on its EDFA (J).
- The receiver threshold is adjusted for the best possible BER.
- The transmission delay of the user is adjusted using the tunable line delay (I) and different fiber lengths between the TDL (I) and the EDFA (J) until the best BER is obtained.
- The transmission delay of the user is varied and BER readings are noted. The oscilloscope is used to check the transmission delay is used.
- Each measurement is repeated 5 times and the mean of the readings is noted.
- The experiment is repeated by adding more interfering users.

Codeword delay of User 4	Codeword delay of User 1	Number of overlaps	Bit error rate
0	0	1	2E-7
0	1	1	4E-7
0	2	0	1E-9
0	3	0	3E-9
0	4	0	1E-9
0	5	0	7E-9
0	6	1	2E-7
0	7	1	6E-7
0	8	0	1E-9
0	9	1	3E-7
0	10	1	4E-7

Table 7.2: Testbed bit error rate in the presence of 1 interfering user.

### 7.1.7 Results

The results for 1, 2 and 3 interfering users are shown in Tables 7.2, 7.3, and 7.4. The results show the bit error rates for particular transmission delay of the users. The table also shows the number of chip overlaps. Only a few results are presented here to illustrate the effect of transmission scheduling on BER. Several more readings were taken.

The limitation of the setup (four users and a (11, 6, 6, 1) codeset) means that the system is operating under very low load. Under low loads, according to the analysis in the previous chapters, transmission scheduling should not provide any benefit. In fact, Aloha-CDMA should perform just as well as any transmission scheduling. However, there are three reasons why transmission scheduling improves the performance even under low loads.

Firstly, under normal operating conditions the bit error rate is very high. This is because of the noise introduced by other sources, *e.g.*, thermal noise, amplifiers, inaccurate delay

Codeword delay of User 4	Codeword delay of User 1	Codeword delay of User 2	Number of overlaps	Bit error rate
0	10	0	2	3E-5
0	10	1	2	1E-5
0	10	2	2	4E-5
0	10	3	2	3E-5
0	10	4	1	4E-7
0	10	5	2	2E-5
0	10	6	1	3E-7
0	10	7	1	3E-7
0	10	8	1	2E-7
0	10	9	1	1E-7
0	10	10	1	3E-7
0	8	0	1	4E-7
0	8	1	1	3E-7
0	8	2	1	5E-7
0	8	3	1	4E-7
0	8	4	0	9E-9
0	8	5	1	2E-7
0	8	6	0	3E-9
0	8	7	1	3E-7
0	8	8	0	2E-8
0	8	9	0	1E-8
0	8	10	0	4E-9

Table 7.3: Testbed bit error rate in the presence of 2 interfering users.

lines, free space delay lines, mode hopping lasers, and environmental conditions. The major reason for these sources of noise is that the testbed is built from discrete components. Secondly, the power input to the receiver AWG must be kept below a limit. This is done by using an EDFA (M). This means that as the number of users increases, the power per chip decreases. To compensate for this the receiver threshold must be lowered when a new user is added on. Therefore, as the number of users increases, the effect of noise increases. Thirdly,

Codeword delay of User 4	Codeword delay of User 1	Codeword delay of User 2	Codeword delay of User 3	Number of overlaps	Bit error rate
0	10	3	0	3	Non recoverable
0	10	3	1	3	Non recoverable
0	10	3	2	2	4E-5
0	10	3	3	2	7E-5
0	10	3	4	2	5E-5
0	10	3	5	2	2E-5
0	10	3	6	2	2E-5
0	10	3	7	3	3E-4
0	10	3	8	2	8E-5
0	10	3	9	2	2E-5
0	10	3	10	2	3E-5
0	10	8	0	2	2E-5
0	10	8	1	2	8E-5
0	10	8	2	1	6E-7
0	10	8	3	1	9E-7
0	10	8	4	1	7E-7
0	10	8	5	1	1E-7
0	10	8	6	2	3E-5
0	10	8	7	2	2E-5
0	10	8	8	2	7E-5
0	10	8	9	1	4E-7
0	10	8	10	1	2E-7
0	8	4	0	1	2E-7
0	8	4	1	1	7E-7
0	8	4	2	0	3E-8
0	8	4	3	1	8E-7
0	8	4	4	0	3E-8
0	8	4	5	0	6E-8
0	8	4	6	1	8E-7
0	8	4	7	1	4E-7
0	8	4	8	1	9E-7
0	8	4	9	0	3E-8
0	8	4	10	0	2E-8

Table 7.4: Testbed bit error rate in the presence of 3 interfering users.

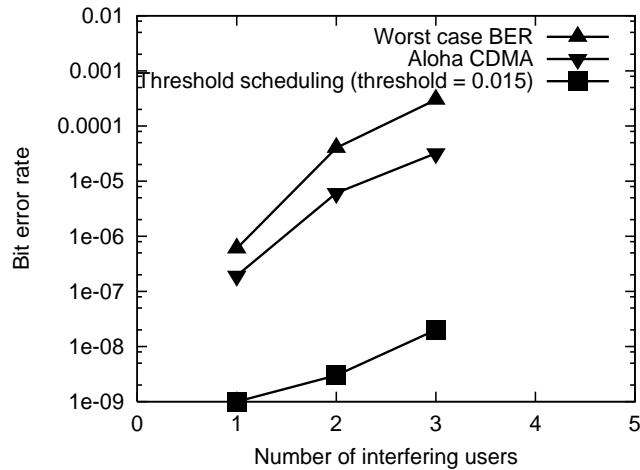


Figure 7.7: Experimental results: The transmission scheduling algorithm results in the best case BER. It occurred under cases when there were zero overlaps. The worst case BER is the worst possible bit error rate that was measured. It occurred in cases when the number of overlaps was the maximum possible. Aloha-CDMA does not control the number of overlaps. Its BER is the average over all the measured cases.

the receiver does not do hard-limiting which increases the interference errors [57]. Finally, beat noise also was a source of noise.

The multi-user interference increases the bit error rate further. Therefore, in this system, even under low loads, transmission scheduling helps improve the throughput. This may be seen from the difference in bit error rates when the number of overlaps increases. As few as one chip overlap can cause the bit error rate to increase by 2 orders of magnitude from  $10^{-9}$  to  $10^{-7}$

Table 7.2 shows the bit error rates for different transmission delay when there is one interfering user on the line. The number of chip overlaps for each case is also shown. Tables 7.3 and 7.4 show the same results for 2 interfering users and 3 interfering users. Note that only a few transmission delays are shown because the total number of combinations of transmission

delays is too large to present. Entries marked ‘non-recoverable’ indicate that the bit error rate was above  $1E-4$ . The BERT did not provide any BER readings above  $1E-4$ . Figure 7.7 shows the bit error rate for threshold scheduling, Aloha-CDMA and the worst case BER. Note that the measured cases are a small subset of the total number of possible combinations. The average case is closer to the worst case than to the best case. This is because the worst case BER is 2 orders of magnitude worse than the best case and the worst case condition happens fairly frequently as may be seen from the Tables 7.2, 7.3 and 7.4.

Under high load conditions, more and more overlaps are likely to occur and the worst case conditions are seen more often for an Aloha-CDMA protocol. A transmission scheduling algorithm however allows only a limited number of codewords on the line and would schedule the transmissions such that the system would have the best case BER.

The two conclusions of this study are:

*In the presence of significant physical layer noise, transmission scheduling can improve performance even under low offered load.*

## **7.2 Implementation of a network interface card**

This section describes the high level hardware design of a Network Interface Card (NIC) for the Interference Avoidance MAC protocol over optical CDMA. The design goals of this implementation are:

- To design a NIC card that functions correctly.
- To minimize the latency of the state estimation and transmission scheduling.

As the latency of state estimation and transmission scheduling increases, the state estimate becomes less accurate and therefore the throughput of Interference Avoidance falls. For a network of maximum diameter 2000m, the delay between the estimation and the merging of packets is approximately  $5\mu\text{s}$ . The throughput of Interference Avoidance does not degrade significantly at this latency. Therefore, as long as the latency of the hardware is a small fraction of  $5\mu\text{s}$ , the throughput does not fall significantly. The primary design goal is to ensure that the latency of the hardware is a small fraction of  $5\mu\text{s}$ . The secondary design goals are to reduce the required logic and the memory.

The NIC card is designed for a network of maximum diameter 2000m, and uses 1 wavelength and a chipping rate  $B$  of 10Gc/s. It supports codeset lengths of up to  $N = 100$  and a weight of  $w = 3$  (can be extended). The card design is for a Tunable Transmitter-Fixed Receiver (TT-FR) architecture (the transmitter may be tuned to any codeword while the receiver codeword is hardcoded into the NIC). The state estimation algorithm is continuous state estimation using the sample means algorithm (Chapter 6). The transmission scheduling algorithm is the threshold scheduling algorithm (Chapter 5).

Figure 7.8 shows a block diagram of the hardware of the NIC. The NIC hardware consists of the *Controller module*, *State observation collection module*, *State estimation module*, *Transmission scheduling module*, *Ranging module*, *Synchronization module*, *optical CDMA transmitter* and *optical CDMA receiver*<sup>1</sup>. When a packet arrives on the receive fiber, it is decoded by the optical CDMA receiver and written to the receive buffer. When a packet arrives

---

<sup>1</sup>The block diagram does not show any amplifiers that may be needed to amplify signals.

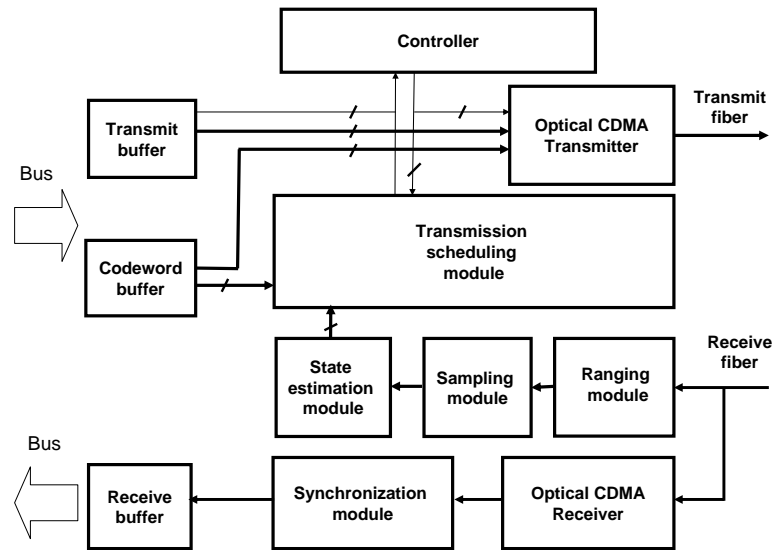


Figure 7.8: A block diagram of an Interference Avoidance Network Interface Card.

on the bus, it is written to the transmit buffer. The transmission scheduling algorithm calculates the appropriate transmission delay tunes the tunable delay lines (TDLs) and signals the transmitter. The optical CDMA transmitter transmits the packet on the transmit fiber. The state observation collection and state estimation modules run continuously and write the state estimate to the transmission scheduling module.

The following sections describe the operation of the individual components of the NIC card.

### 7.2.1 Bus interface

The function of the bus interface is to transfer data between the node processor and the network interface card. When a packet is available for transmission at the node processor, the

packet to be transmitted and the codeword to be used for optical CDMA encoding are sent by the node processor to the NIC across the data lines of bus<sup>1</sup>. The packet and the codeword are stored in buffers (called the transmit buffer and the codeword buffer respectively). The transmit buffer should be large enough to store at least 1 packet, while the codeword buffer must be large enough to store at least 1 codeword. If packets are arriving faster than the transmission scheduling module can process them (100 ns) then additional buffering is needed. The bus could be a commercially available or future version of a bus such as PCI or USB. The bus could be replaced by a network interface for another LAN such as Ethernet.

## 7.2.2 State observation collection module

The function of the state observation collection module is to convert a multivalued optical input signal to an electronic digital signal. The state observation collection module has both electronic and optical components.

Figure 7.9 shows a diagram of the state observation collection module. The input to the state observation collection module is the incoming signal on the receive fiber<sup>2</sup>. The module consists of a splitter, a photodetector, threshold detectors, and collection module. The diagram shows an implementation of the state observation collection module with four threshold detectors. In any given chip time, the signal is a multivalued input because it is the sum of a particular number of '1 chips'. The photodetector converts the multilevel input signal to an electrical signal. The threshold detectors and the logic convert the signal to a digital value.

---

<sup>1</sup>As described in Chapter 4, the codeword used for encoding is determined by a higher layer address to codeword mapping. This mapping is out of the scope of this work

<sup>2</sup>For this discussion the ranging module is ignored.

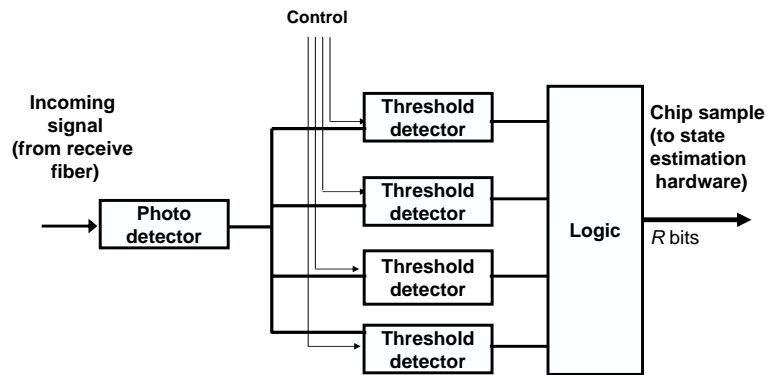


Figure 7.9: State observation collection module. The input to the state observation collection module is a multivalued optical signal that arrives at the chipping rate. The output is an electronic digital value, a chip observation, which changes at the chipping rate. Each chip observation represents the magnitude of the multivalued signal in a chip time. The figure shows 4 threshold detectors, enough to detect up to 4 different chip magnitudes. The number of detectors can be modified based on the accuracy desired. Section 7.2.6 discusses why a minimum of 3 is needed.

The thresholds of the threshold detectors are set to values that correspond to increasing values of the multilevel input signal. For example in Figure 7.9, they are set to correspond to 0.5, 1.5, 2.5 and 3.5 chips that are the values needed to differentiate between chip magnitudes 0, 1, 2, 3, and greater than 3. The number of threshold detectors that are triggered depends on the value of the signal. The logic is an encoder that converts the output of the threshold detectors to bits. The output of the state observation collection module is a digital signal that corresponds to the multivalued input. This output is called the *chip observation*. The flash ADC is a good candidate for this function and flash ADC with resolutions of 3 bits have been

demonstrated at speeds of up to 10G samples/s [21]<sup>1</sup>. The output of the state observation collection module is connected to the state estimation hardware.

The state observation collection module operates at the chipping rate (10 Gc/s). It writes new chip observations to the state estimation module at 10GHz. It can be implemented as a custom ASIC which interfaces with the photodetector.

The number of bits in the chip observation is called the *resolution*  $R$  of the state observation collection module. The larger the resolution of the module, the more accurate the state estimate. The resolution should be  $\log N$  for accurate representation of the state (at 100% normalized offered load,  $N$  codewords are on the line, and therefore the maximum size of a chip observation is  $N$ ). However, the larger the resolution, the greater the number of components in the hardware and therefore the higher the cost of implementation. Section 7.2.6 discusses how to reduce the required resolution without substantially affecting the accuracy of state estimation.

### 7.2.3 State estimation module

The function of the state estimation module is to calculate the estimated state vector from the chip observations provided as input from the state observation collection hardware. This implementation is based on the continuous state estimation algorithm as specified in Chapter 6. It works by collecting a number of state observations ( $n_e$ ), adding them, multiplying the result by 2 and dividing the result by the number of state observations (Refer Chapter 6).

Hardware can use parallelism to operate concurrently on every chip observation of the state.

---

<sup>1</sup>An alternative is to perform optical analog to digital conversion. Optical analog to digital conversion has been demonstrated at resolutions of 5 bits at optical transmission rates (10 Gb/s) [7].

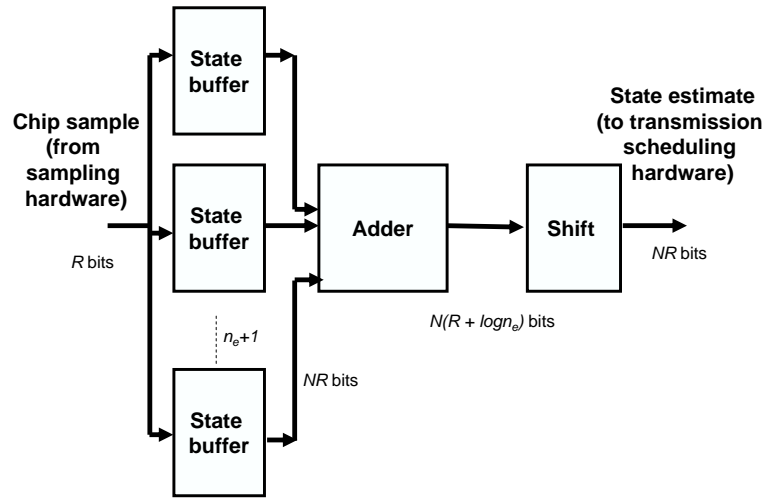


Figure 7.10: State estimation module. The input is a stream of chip observations arriving at the chipping rate. The output is a state estimate that is output at the clock rate of the hardware. The figure also shows the width of the data lines in the hardware.

Figure 7.10 shows a diagram of the state estimation hardware. The module consists of latches, an adder and a shifter. The latches are arranged into  $(n_e + 1)$  groups called state buffers. Each state buffer stores one state observation. Each state buffer consists of  $N$  chip buffers. Each chip buffer stores one chip observation. The  $(n_e + 1)$  state buffers are designed as a circular buffer, *i.e.*, new values of the chip observations overwrite the older values. The state observation collection module writes chip observations to the input of the state estimation module at the chipping rate (10Gc/s). Therefore, the latches must operate at clock speed of 10GHz. Every clock cycle, a chip observation is written to a different chip buffer.  $N$  clock cycles after startup, one state buffer has a state observation.  $n_e N$  clock cycles after startup, all state buffers are filled. Subsequent chip observations overwrite the older ones. The adder

adds  $n_e$  state observations (the state buffer that is currently being written not used as input to the adder). The output of the adder is multiplied by 2 and divided by  $n_e$ . In this implementation  $n_e$  is set to 16<sup>1</sup>. Therefore, the result can be obtained by dividing by  $16/2 = 8$ , *i.e.*, shifting it left by 3 chips. The division can be done by a simple reordering of the output of the adder. A 1 must be added to the output of the adder before the division to ensure that the division rounds up. The output of the shifter is the state estimate. The adder is run only when a new state observation is available, *i.e.*, after  $N$  chip observations are collected. Note that the state buffer is written to at the chipping rate, *i.e.*, 10 GHz. However, the adder and the shifter need only run at a speed of  $10\text{GHz}/N = 100 \text{ MHz}$ .

FPGA hardware available currently along with a custom ASIC may be used for implementation. The first stage of the hardware (the state buffers) may need to be a custom ASIC because of the high rate at which chip observations are written to the latches (10GHz). Subsequent stages operate at a lower clock speed (100MHz) and can be implemented in an FPGA. If the hardware clock is lower than the data rate, the design may need to be modified by adding sufficient buffering. The clock speed of current FPGA hardware is 400MHz, well above the requirements of the hardware.

The latency of this implementation is well below the design goal of  $5\mu\text{s}$ . If the adder completes its operation in 1 clock cycle, *i.e.*,  $1/100 \text{ MHz} = 10 \text{ ns}$ , then for  $n_e = 16$ ,  $N = 100$ ,  $B = 10\text{Gc/s}$ , it takes 160ns for the first state estimate and 10ns for the subsequent state estimates.

---

<sup>1</sup>Chapter 6 showed that any value of  $n_e$  greater than 10 is sufficient for state estimation in a codeset of length  $N = 100$ .

The processing requirements of the state estimation module are within the capability of current FPGA hardware. Each operation on a component of the vector needs a separate hardware unit. The adder must perform  $N$  parallel additions, each of which has inputs of  $n_e$  chip observations of  $R$  bits each. Therefore, it needs  $Nn_e$  hardware units. For  $N = 100$ ,  $n = 16$  this is 1600 hardware units. This is well within the capabilities of current FPGA chips which can have around 100000 hardware units [73]. Note that the design allows the latency of the module to be traded against the complexity of implementation by serializing some operations. A deeper study of this trade-off is an area for future work.

The buffer requirements for the hardware algorithm are very low. The state buffers contain  $NR(n_e + 1)$  bits. For  $N = 100$ ,  $R = 2$ ,  $n_e = 16$  this is 3400 bits or approximately 0.5kB. This is a negligible compared to current ASIC chips or even FPGA chips, which have up to 100000 latches and typical on-chip memories of 8MB [73].

#### **7.2.4 Controller module**

The controller module provides top level control to the NIC card. On arrival of a packet for transmission, it signals the transmission scheduling module. On completion of transmission scheduling, it reads the results and signals the transmitter to begin transmission at the appropriate time.

#### **7.2.5 Transmission scheduling module**

The function of the transmission scheduling module is to take as input the state estimate and the codeword to be transmitted and calculate the delay at which to transmit the codeword.

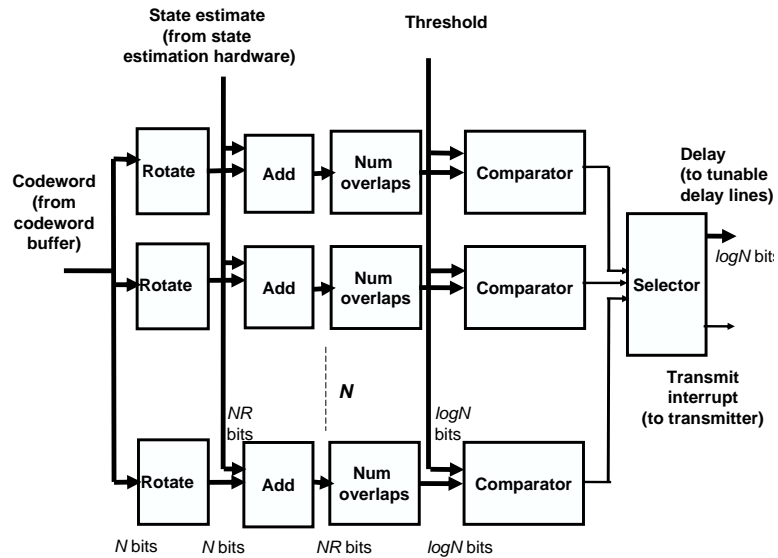


Figure 7.11: Transmission scheduling hardware. There are two inputs: the codeword that arrives on packet arrival and the state estimate, which changes every clock cycle. The output is the delay in terms of number of chips from the start of the hardware clock cycle. This delay is used to delay the transmission. The width of the data lines is also shown in the figure.

The hardware implements the threshold scheduling algorithm. It checks every possible delay of the codeword to see if it satisfies two criteria: The number of overlaps in the resulting state should be below a threshold and the codeword should not be lost due to an error. It chooses a transmission delay that satisfies these criteria. This algorithm can be implemented using parallelism to operate concurrently on different rotations of the codeword vector and to operate concurrently on each component of the vector.

Figure 7.11 shows the transmission scheduling hardware. The module consists of rotators, adders, combinational logic, comparators, and an encoder. The input to each of these is either one or two vectors. The input to the module is the codeword for encoding and the state estimate. The rotate logic rewires each codeword such that a different rotation enters the next

stage. Each rotation corresponds to transmission of the codeword at a different delay. There are  $N$  possible rotations, so there are  $N$  parallel paths. The next stage (the adder) adds the rotated codeword to the state estimate. The two stages (the number of overlaps combinational logic, and the comparator) determine if the rotation satisfies the two criteria. The output of the comparator sets its output bit to 1 if the criteria are satisfied. The next stage (the selector) selects one of the enabled inputs at random. The output of the selector is used to tune the tunable delay lines. The delay lines can be tuned in steps of the chip time  $t_c$ , from 0 to  $Nt_c$ . The transmitter can transmit after the transmission scheduling and tuning of the delay lines.

It is assumed that each stage takes one clock cycle to complete. Therefore, after 4 cycles the result is available. For a clock speed of 100 MHz, this is 40ns. The tunable delay lines can be tuned at speeds of up to 1 GHz, *i.e.*, latencies of 1ns. Therefore, the latency of the module is a very small fraction of  $5\mu s$ .

The processing requirements of the hardware for transmission scheduling are within the capability of current FPGA hardware. Each operation on a component of the vector needs a separate hardware unit. There are 3 stages (adder, combinational logic and comparator) which operate on the vectors. There are a total of  $3 * N * N = 3N^2$  units. For  $N = 100$ , this corresponds to 30,000 hardware units. This is within the capabilities of current FPGAs, which have up to 100,000 units. The tunable delay lines can be implemented using photonic VLSI tunable delay lines [40], [20]. For a codeset with  $N = 100$ , at a chipping rate of 10 Gc/s, the tunable delay lines must be tunable from 0 to 10ns in steps of 0.1ns. This corresponds to delay lines of length 2m that can be tuned in steps of 2cm. These figures are well within the range of photonic VLSI tunable delay lines.

The memory requirements for this algorithm are negligible. It has to store the result, which is a delay in the range 0 to  $N$ . This is  $\log N$  bits of memory which is 8 for  $N = 100$ . The hardware may also need latches to store intermediate results or to store inputs. The latches are not indicated in the diagram. The hardware may be integrated with the state estimation hardware on a single FPGA.

### 7.2.6 Optimizing the state observation collection and state estimation module

This section discusses how the limitations of the state observation collection module may be overcome. The required sampling rate of the ADC is equal to the chipping rate (10Gc/s). In general, for an ADC, the maximum possible resolution reduces as the sampling rate increases. High resolution, high sampling rate ADCs cost significantly more, occupy large die space, and dissipate more power. Therefore, a low resolution has several advantages. The number of components of the other hardware units (state estimation and transmission scheduling) and their latency are also reduced.

With a resolution of 2 bits, the hardware can differentiate between 0, 1 and an overlap (a minimum of 3 threshold detectors are needed for this). The output of the state observation collection module is a 2 bit value (00 for a chip magnitude of 0, 01 for a chip magnitude of 1 and 10 for an overlap (a chip magnitude greater than 1)).

The consequence of this reduction of resolution is that the state observations and the estimated state have no information about the magnitude of an overlap. However, the threshold scheduling algorithm can operate with such a reduced state estimate <sup>1</sup>. The algorithm needs

---

<sup>1</sup>Other transmission scheduling algorithms such as codeword estimation may need a complete state estimate.

to know the number and the location of overlaps in the estimated state. It does not need the magnitude of each overlap. Therefore, the threshold scheduling algorithm can use this reduced estimated state as input. The state estimation and transmission scheduling algorithms and hardware remain unchanged.

This results in savings in the state observation collection hardware. While this does not result in significant cost reduction in the state estimation and transmission scheduling hardware, the size and latency of the circuitry reduces. Determining the extent of these gains is an area for future research.

### **7.2.7 Ranging and synchronization modules**

The function of the ranging module is to compensate for the difference in the bit clock at the transmission and estimation points. It does this by adding a delay to the signal before it enters the state observation collection module.

The state is estimated by collecting observations at the estimation point. This state is used for transmission scheduling at the transmission point. Chapter 6 explained how the state at the state estimation point can be used for scheduling at the transmission point only if they are separated by an integral number of state spans. However, the separation may not be an integral number of state spans because the fiber length may not necessarily be an integral number of state spans. The ranging module compensates for this by adding a delay to the signal before it enters the state observation collection module. It consists of a comparator and tunable delay lines. When the node is booted up, the ranging module modulates the optical CDMA transmitter to continuously transmit a ranging sequence. The sequence passes

through the coupler and returns on the receive fiber and passes through the delay lines. The ranging module then compares a stored value of the transmitted signal with the signal at the output of the delay lines. It tunes its delay lines until the sequences are identical, *i.e.*, there is no phase difference between the two. The signal from the receive fiber passes through the delay lines before entering the state estimator. Care must be taken during layout of the chips to ensure that the delay between ranging hardware output and the state estimator is insignificant compared to a chip time or that it is compensated for. The ranging operation is done at startup and periodically thereafter at intervals to be determined empirically. It may be performed on a different wavelength from the data.

The function of the synchronization module is to detect the start of a frame destined for the node. Several synchronization algorithms have been proposed for optical CDMA [29, 30]. These algorithms may be implemented in hardware as a separate block on the interface card. Synchronization hardware may be constructed of Phase Locked Loops (PLLs), Voltage Controlled Oscillators (VCOs) and other electronic components.

Note that the performance (latency) of ranging and synchronization may be affected by multi-user interference. The interaction between the ranging/synchronization algorithms and Interference Avoidance is an area for future work.

### **7.2.8 Optical CDMA transmitter**

The function of the transmitter is to encode the packet to be transmitted using the codeword of the receiver. The transmitter consists of tuning hardware, an encoder, and a modulator.

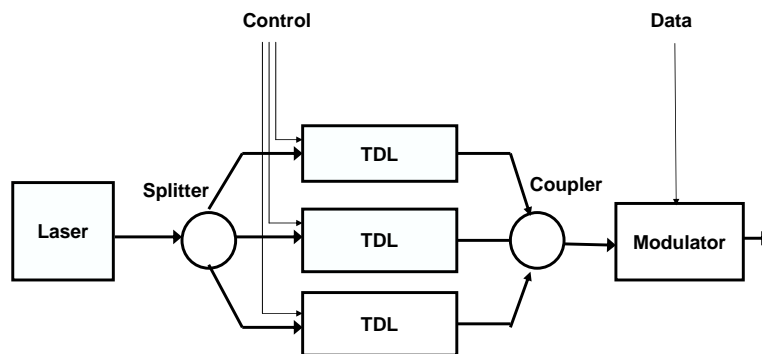


Figure 7.12: Optical CDMA transmitter hardware implementation. The input is a stream of bits. It modulates a stream of codewords being transmitted by the laser and encoder.

Figure 7.12 shows a typical transmitter. The encoder consists of a laser followed by a splitter and  $w$  (in the figure  $w$  is 3) tunable delay lines. The output power of the laser is around 10dBm. Every  $t_b$  (10ns) the laser produces a pulse of width equal to the chip time  $t_c$  (0.1ns). The splitter splits the pulse and three pulses enter the tunable delay lines. Each tunable delay line can be tuned from 0 to  $Nt_c$  in steps of  $t_c$ . The coupler couples the signals from the delay lines and the codeword emerges from the coupler. For a codeset with  $N = 100$  and  $w = 3$  at a chipping rate of 10 Gc/s, this means 3 arrays of tunable delay lines each of which can be tuned from 0 to 10ns in steps of 0.1ns. This corresponds to a length of 2m, which can be tuned in steps of 2cm. The delay lines are tuned by switching the required delay lines in or out of the path. Photonic VLSI tunable delay lines can be used for this [40], [20].

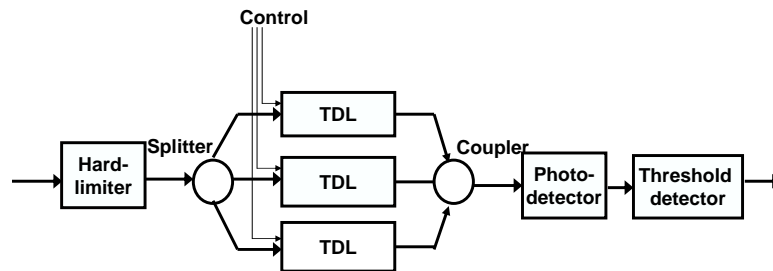


Figure 7.13: Optical CDMA receiver hardware. The input is the signal on the optical fiber at the chipping rate and the output is the received data at the bit rate.

The encoder produces a stream that consists of the repeated codeword. This stream is externally modulated by the data. The modulator is based on a laser modulator. It is modulated at the data rate (100 Mb/s) which is well within the capabilities of current modulators.

The encoder hardware polls the codeword buffer. It reads the codeword from the codeword buffer and tunes the delay lines of the encoder. The tunable delay lines can be tuned at speeds of up to 1 GHz. Therefore, the latency of the tuning is around 1ns.

### 7.2.9 Optical CDMA receiver

The function of the receiver is to decode the received signal and receive packets. The receiver is a hard-limiting correlation receiver. Chapter 2 discussed the design of such a receiver.

Figure 7.13 shows a diagram of the receiver. It consists of a hard-limiter, photodetector, delay lines, and a threshold detector.

The optical hard-limiter [57] limits the power in each chip time to the power of a single chip. The splitter splits the optical signal. The delay lines are designed such that the specific chips corresponding to a particular codeword align in time at the coupler. The delay lines are the same type as those described in the transmitter design. The coupler couples the signals. The photodetector is a PIN diode. It converts incident light power to electric current. The receiver sensitivity of the photodetector is around -20dBm. A threshold detector [57] determines whether the signal exceeded the threshold which is set to  $w$  (in the figure, 3) times the value of a single chips power. If the signal exceeds the threshold, a 1 is detected, else a 0 is detected. The data is written to the receive buffer.

The network architecture is a Tunable Transmitter Fixed Receiver architecture. The receiver codeword may be hardwired into the NIC. However, to allow NICs to be used across nodes, the receiver codeword is made tunable by providing tunable delay lines as shown. This feature is not used to tune the receiver during network operation. Hardware to perform link layer functions such as link layer address checking and error checking is placed after the receive buffer.

### **7.2.10 Timing**

A timing diagram based on the timing constraints discussed in the previous sections is shown in Figure 7.14. The diagram shows a single cycle from arrival of the packet to transmission scheduling to eventual transmission and merging. The first line shows the state estimation

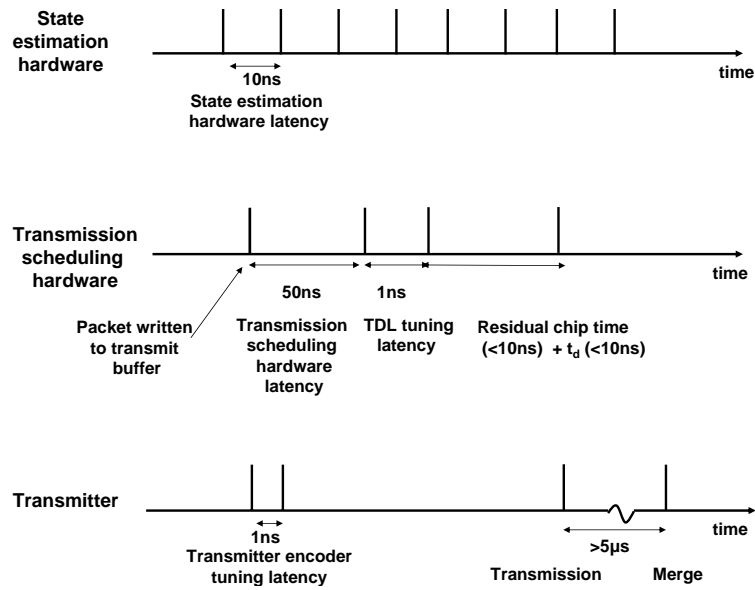


Figure 7.14: Timing diagram (not to scale). The diagram shows the timing of events on a NIC for a 10Gc/s network using a codeset of length  $N = 100$ , weight  $w = 3$ . The hardware clock speed is 100MHz. The hardware design is as described in this section. A new state estimate is generated every 10ns. On packet arrival, the transmission scheduling hardware operates on its input and produces its result in 50ns. After tuning delays and the calculated delay ( $t_d$ ), the packet is transmitted. The residual chip time (the time between the delay line tuning and the start of the next hardware clock cycle) is also shown.

hardware. Every 10ns a new state estimate is written to the output. The second line shows the operation of the transmission scheduling algorithm. When a packet arrives, the codeword and state estimate are read and the algorithm takes 50ns to execute. After 50ns the tunable delay lines are tuned (1ns) and the transmitter is interrupted. The third line shows the transmitter operation. The transmitter is tuned on packet arrival (1ns) and the transmission is started when the transmission scheduling hardware raises an interrupt.

### **7.3 Conclusions**

This chapter studied the performance of transmission scheduling under real world conditions. The testbed operated under conditions of severe physical layer noise and had a limitation on the offered load. The main sources of noise were the discrete components. The non-hard-limiting receiver and the reduction of the receiver threshold further increased the impact of noise. Under these conditions, threshold transmission scheduling with a very low threshold improved the BER by two orders of magnitude. The laboratory testbed provided insight into how a NIC for Interference Avoidance can be designed. This work indicates that a NIC based on present day hardware and optical devices can implement Interference Avoidance.

## **Chapter 8**

### **Related work**

Prior work related to Interference Avoidance may be classified based on the type of physical layer:

- Wired networks with a single channel.
- Wired networks with multiple orthogonal channels.
- Wired networks with multiple pseudo-orthogonal channels (Optical CDMA networks).
- Wireless networks.

Chapter 3 discussed the differences in state and its properties in optical CDMA and single channel, shared medium, wired, and wireless networks. This chapter extends that discussion with descriptions of the media access control protocols and features that distinguish them from Interference Avoidance.

## 8.1 Wired networks with a single channel

A single channel network is a network where the transmission medium has only a single channel, *e.g.*, a frequency that is shared by all nodes on the network for transmission and reception. Media access control for shared medium single channel networks has been classified [5] into the following types

- Contention, *e.g.*, Aloha [5], CSMA/CD [68], 802.11 [55]
- Reservation, *e.g.*, Token ring, token bus, DQDB [5], IEEE 802.14 [19], FDDI [5]

### 8.1.1 Contention protocols

In contention protocols (also called random access protocols) nodes contend for access to the channel. Chapter 3 discussed the differences between optical CDMA and single channel shared medium networks in the areas of state, interference, and correlation of state. To summarize that chapter, state is a scalar value (Idle, Busy, or Collision). Collisions are mutually destructive and the network exhibits state correlation at low speeds and low diameter.

Several media access control techniques have been devised to reduce the probability of collisions. Among the well studied random access protocols are the Aloha mechanisms and the Carrier Sense Multiple Access (CSMA) mechanisms. With the Aloha [5] protocol, the maximum throughput is limited to around 18% because of collisions. Time slotted Aloha reduces collisions and increases the throughput to 36%. Carrier sensing [68] is a mechanism by which nodes first sense the medium to ensure that the medium is free before transmitting. Collision detection [68] is a mechanism by which nodes continue to monitor the medium

during transmission and abort if they detect a collision. Both mechanisms further reduce collisions, raising the maximum throughput to around 80% [34]. Carrier Sensing and collision detection are effective for networks where the normalized propagation delay is low. As the diameter of the network increases, or the transmission speed increases or packet size decreases, CSMA/CD becomes ineffective. To compensate for this effect Gigabit Ethernet [18] has defined two mechanisms: Carrier Extension and Packet Bursting. In Carrier Extension, the packet transmission is followed by transmission of a carrier signal that prevents other nodes from transmission. In Packet Bursting, packets are buffered until a larger size packet may be created. The larger size of the packet means that the normalized propagation delay is reduced. Both mechanisms improve the throughput of Gigabit Ethernet. Interference Avoidance may be regarded as another method of extending Carrier Sensing. It improves performance by increasing the normalized propagation delay by a factor equal to the codeset length  $N$  because of encoding (Appendix F).

In the case of a collision in a single channel network, several conflict resolution algorithms [44] and splitting algorithms [5] have been proposed to resolve collisions. Each mechanism results in some improvement in throughput and reduction of delay. Mechanisms to reduce collisions by controlling when a node transmits (1-persistent, non-persistent,  $p$ -persistent) [68] and algorithms to defer retransmissions (Binary Exponential Backoff (BEB), Binary Logarithmic Arbitration Method (BLAM)) have also been proposed. The performance analysis of Interference Avoidance in this work assumes 1 persistent sensing and that

packets are dropped if they cannot be transmitted. The throughput, delay and stability performance of Interference Avoidance with the other defer mechanisms is an area for further research.

### **8.1.2 Reservation protocols**

In reservation protocols, nodes are selected to transmit by a controller, either centralized or distributed. The controller may be a central server, a broadcast channel, or a token. Examples of this type of access mechanism are Token ring/bus and polling protocols such as DOCSIS/IEEE 802.14. Reservation MAC protocols [5] do not have a limitation on diameter or speed of the network. However, reservation MAC protocols need a centralized scheduler and a channel for reservation. The chief drawbacks of these networks are the management of the scheduling mechanism and the access delay. Reservation protocols represent the other end of the spectrum of media access control protocols where collisions are eliminated by centralized control.

The DOCSIS 2.0 [19] specifications define a standard called Synchronous-CDMA for the physical layer of a cable TV based network. It is based on a combination of TDM and synchronous CDMA. Time is divided into minislots. The MAC protocol is a reservation type protocol. Nodes reserve codewords to use in a time slot. In any given minislot, several nodes on the network may transmit simultaneously using different CDMA codes. The codeset specifies a set of 128 orthogonal codes. The specification limits the number of simultaneous codewords. The disadvantages of this mechanism include the necessity for slot synchronization, the use of fixed length codesets and a media access control based on limiting the number

of codewords. This mechanism is similar to the Interference Avoidance techniques in that it performs admission control and restricts the number of simultaneous users on the network in a wired CDMA network. However, it is centralized and admission control is static (by codeword assignment) and does not depend on the current state of the line. The application of reservation protocol principles to optical CDMA is an area of open research.

## **8.2 Wired networks with multiple orthogonal channels**

The bandwidth of a large bandwidth (high speed) physical medium may be divided into slower channels. One way to do this is through Frequency Division Multiplexing (on high speed optical LANs wavelengths can be used as channels). The data rate on each channel is lower than the data rate of the single high speed channel. If the frequencies are sufficiently far apart, the channels are orthogonal, *i.e.*, they do not interfere with each other. Carrier sensing techniques can be used on the slower channels. [41] studied the use of Carrier Sensing in multi-channel networks (MC-CSMA). State in a multichannel network is a vector. Each component of the vector represents the state of a particular channel and can take on a value of Idle, Busy, or Collision. Collisions on a channel are mutually destructive and the network exhibits correlation of state if the channel has low normalized propagation delay. MC-CSMA also extends the carrier sensing principle to high speed networks. The difference is that Interference Avoidance operates on pseudo-orthogonal channels. In networks where the cost of creating orthogonal channels is high (tunable lasers for WDM may be expensive), Interference Avoidance may be preferred.

### **8.3 Wired networks with pseudo-orthogonal channels**

Optical CDMA is a physical layer with pseudo-orthogonal channels. Transmissions on different channels (codewords) interfere with each other. Interference Avoidance is a contention based media access control for pseudo-orthogonal channel. It is carrier sensing applied to a network with multiple interfering channels. Both mechanisms attempt to determine the state of the medium and schedule packet transmissions to improve throughput. As discussed, Interference Avoidance techniques can be reduced to the simpler Carrier Sensing and Aloha protocols (Appendix F).

Other work in the area of MAC protocols for optical CDMA networks has focused on codeword choice [63] and eliminating receiver collisions [74]. Shalaby [63] studied a random access MAC protocol for optical CDMA. The protocol is based on pre-transmission coordination between the sender and receiver before transmission. Codewords are selected from a codeword pool with or without replacement. Codewords are randomly shifted before use. A control packet is sent on a common control channel before the transmission. These protocols control the use of codewords of a codeset, in contrast to the Interference Avoidance techniques studied in this work that focus on throughput maximization by scheduling transmissions.

### **8.4 Wireless networks**

Chapter 3 discussed the differences between optical CDMA and single channel shared medium wireless networks in the areas of state, interference, and correlation of state. To summarize

that chapter, state is a scalar value (depends on the channel conditions), collisions are non-mutually destructive (transmissions may be captured) and the network exhibits very low correlation. Depending on the receiver design, modulation and channel condition, the receiver can *capture* [72] transmissions. Capture is the ability of a receiver to receive a packet even if it overlaps with other packets. In a network that exhibits capture, a node can preserve itself by raising its transmission power, while possibly destroying other transmissions. While capture is similar to the non-mutually destructive optical CDMA interference, the difference is that it is controlled by power rather than delay. The subset of packets being destroyed cannot be controlled. As mentioned in Chapter 3, a multi-packet reception model [69] has been proposed for modeling the reception of a subset of packets.

This section looks at other techniques that exploit the non-destructive property or capture through means such as centralized admission control.

#### **8.4.1 Channel load sensing**

Channel load sensing [1, 24, 70] is a packet radio system where nodes sense the channel load and refrain from transmitting if the load exceeds a threshold. The load is measured by estimating the number of simultaneous transmissions. One method of estimating the load is to measure the received pseudo noise power. Abdelmonem et al. [1] analyzed this type of protocol and developed throughput-delay curves and showed the effects of threshold on channel backlog. Judge et al. [24] considered a similar scheme where the multiple access interference threshold (number of simultaneous users) is tuned dynamically. They have analyzed such a scheme and found it can increase throughput. Toshimitsu et al. [70] examined a

channel load sensing protocol based on a slotted system where slot size was less than packet size, thereby enabling the system to handle access timing delays (in the absence of such a scheme, access timing delays normally cause a degradation of throughput). Note that in all these schemes the media access is through admission control, *i.e.*, the number of simultaneous users is controlled. Interference Avoidance differs from these by performing scheduling rather than admission control. It may be noted that scheduling can be reduced to admission control, *e.g.*, by using a threshold scheduling algorithm with a low value of threshold.

#### **8.4.2 Multi-user interference estimation**

Multi-user interference estimation is used to estimate interference levels in wireless networks. [75] uses a multi-timescale interference predictor to predict the occurrence of interference. Depending on the level of interference, rate and admission control are used to control transmissions on the network. The time scales at which these algorithms work are usually on the order of seconds (1 s). The algorithms depend on the self similar nature of traffic. In contrast, state estimation makes no assumption on traffic characteristics and operates on smaller time scales.

#### **8.4.3 MIMO and CSIT**

Multiple Input-Multiple Output (MIMO) is a technique in wireless communication. It exploits spatial diversity by using multiple transmit and multiple receive antennas. The receiver receives signals on multiple channels. The signals can be processed to obtain the transmitted data. If channel information is known (Channel State Information at the Transmitter (CSIT))

the diversity obtained by using multiple antennas can be exploited to improve performance. State is conveyed to the sender by the receiver.

MIMO/CSIT attempts to replicate signal transmissions based on channel state information to improve detection probability. This technique is similar in principle to state estimation and transmission scheduling mechanisms described in this work. The difference is that Interference Avoidance attempts to schedule packet transmissions based on line state estimates to minimize interference with other packets.

Adireddy [56] proposed a decentralized access protocol called *transmission control* that uses channel state information to schedule packet transmissions. The channel state calculated by a node in a slot interval is used to calculate the probability with which a node should transmit a packet in the next interval. Results for this mechanism show that it stabilizes throughput when compared to simple Aloha mechanisms. The differences between this mechanism and Interference Avoidance are the state description. In transmission control, state is a single valued variable which is used to decide a probability of transmission in a slotted system. In Interference Avoidance, state is a vector used to calculate the transmission times.

#### **8.4.4 Other related work in transmission scheduling**

Shepard [64] proposed scheduling algorithms for wireless CDMA networks. A node decides its transmission schedule and broadcasts it to surrounding nodes. Surrounding nodes choose their schedules based on schedules they receive. Nodes synchronize their clock and choose schedules to minimize interference. It was shown that this mechanism can reduce loss of packet errors when compared to random access. Similar to Interference Avoidance,

this mechanism also schedules based on state of the medium. However, the difference is that Interference Avoidance calculates state in a distributed manner whereas this mechanism uses broadcasts of future schedules to propagate state information. Similar to this mechanism, Chockler [8] considered adaptive algorithms to eliminate collisions in a wireless network. The algorithms allowed nodes on the network to reach consensus about transmission schedules.

## **8.5 Other relevant work in optical CDMA networks**

Work in the area of optical CDMA can be classified into the following categories: Bit error rate analysis, codeset design, effect of FEC and media access control. The work described below is relevant to this work because it describes the behavior of optical CDMA under different conditions or parameters. However, none of these focuses on the behavior of the system in the high load regime.

### **8.5.1 Codeset design**

The objective of most of the work in codeset design is to generate codesets with as large a number of codewords as possible. The work has focused on construction of codesets whose size approaches the Johnson bound. Chung et al. [10] described several algorithms to construct OOCs. These constructions are for codes with maximum cross-correlation parameter  $\kappa = 1$ . Chung and Kumar [9] described a method for construction of codes with  $\kappa = 2$ . Several construction methods for OOCs are described in [45] and [11] among others. The construction methods focus on codes with low cross-correlation parameter. This work has

shown that irrespective of the codeset parameters, at high offered load, throughput will collapse without media access control (or other form of error handling).

### **8.5.2 Bit error rate performance analysis**

In the area of optical CDMA network design, Salehi [57, 58] analyzed an optical CDMA based network and developed expressions for the bit error rate of a network that uses codesets with  $\kappa = 1$ . Azizoglu et al. [3] determined the error rate for codesets with  $\kappa = 2$ . They showed that the bit error rate does not degrade significantly when  $\kappa$  is increased from 1 to 2. Shalaby [61, 62] examined the effect of two different cross-correlation parameters on the throughput of an OOK-CDMA network. The result of the analysis was that under low load conditions, throughput could be increased by increasing the  $\kappa$  from 1 to 2. All of the above work considers the performance under low load conditions unlike Interference Avoidance, which focuses on the high load regime.

### **8.5.3 Forward error correction**

Hsu et al. [16, 17] analyzed the performance of slotted and unslotted optical CDMA packet networks. They developed expressions for the throughput of the network and showed performance can be improved using Forward Error Correction (FEC) codes and hard limiters. Lee et al. [39] analyzed the performance of OOCs by assuming cross-correlation distributions to be Gaussian. They used their analysis to compare the performance of different code constructions. They showed that under low load conditions the performance of optical CDMA networks depends on the mean and variance of cross-correlation values. Muckenheim et

al. [46, 47] studied the effect of bit error probability on the packet error probability and suggested the use of block codes to reduce packet errors. They also described a random delay protocol to reduce the errors incurred during periods of high activity. Forward Error Correction may be added to Interference Avoidance to further reduce interference losses. The impact of FEC on Interference Avoidance is an area for future work.

## **Chapter 9**

### **Conclusions and future directions**

This chapter discusses the contributions and conclusions of this work and future directions that the research opens up.

#### **9.1 Conclusions**

Optical CDMA has been studied for years as an experimental technology for optical LANs. Several researchers have proposed its use for different types of networks. The current effort to deploy Fiber to the Home networks has provided motivation for studying those properties of optical CDMA that may prove useful in an all optical access network.

However, optical CDMA has not yet reached the stage where it can be deployed at a viable, high performance, scalable system. Several concerns have been expressed about the use of spread spectrum on an optical link. The main concern is that the use of optical CDMA results in low network throughput [38].

The contribution of this dissertation is the concept of Interference Avoidance. Interference Avoidance mitigates the problem of throughput collapse through media access control.

Prior to this work, little work had been done in the area of media access control for optical CDMA. This work identified two properties exhibited by optical CDMA LANs: non-mutually destructive interference and state correlation. These properties are exploited by Interference Avoidance, which is a generalization of Carrier Sensing (Appendix F). This led to the proposal of two mechanisms: State estimation and Transmission scheduling. This study demonstrates that when all the configurable parameters (codeset, transmission scheduling, and state estimation) are chosen correctly, throughput collapse under high offered load can be prevented. Pure selfish, overlap section and threshold scheduling algorithms can effectively prevent throughput degradation at high loads when used with low codeset weights. The algorithms performed within 15% of an optimal codeword scheduling algorithm. Optical CDMA networks have a wide range of parameters at the physical and media access control layers. The techniques presented in this work are effective tools that may be used to determine the optimal network configuration for a given condition. Based on the performance analysis, the ideal parameter settings are determined to be the threshold scheduling algorithm with a low threshold ( $< 0.3$ ), continuous state estimation with a low number of observations ( $< 10$ ) and a long codeset length ( $> 100$ ) and low weight ( $< 5$ ).

While this work solves the throughput collapse problem, it represents a small piece in a larger, complex, optical CDMA network architecture. In the wireless domain, CDMA networks have been built by integrating several complex pieces which address many different problems such as code acquisition, the near far problem, equalization etc. Similarly, an optical CDMA network will need several other pieces before it can be considered a viable technology. Firstly, the physical layer and link layer need extensive research. There are three

main areas where significant analytical and experimental work needs to be done. They are: power control, ranging algorithms and synchronization algorithms. These three functions represent the bare minimum needed, along with Interference Avoidance, to provide low latency, low packet error rate access to a shared medium. Secondly, once these areas are well understood, the interactions between the mechanisms need to be studied. For example, the Interference Avoidance mechanism may improve the serial search algorithm, leading to shorter synchronization times. Cross layer or cross mechanism design can be used to optimize the performance of the four mechanisms. Such a study represents a challenge in understanding how the independently designed pieces will interact to impact overall throughput, latency and other quality of service metrics. Finally, an integrated systems implementation of all four mechanisms will be needed to study real world performance.

Once these three steps are complete, these four mechanisms will provide physical, link and media access control layers that are scalable and fault tolerant. Network designers will then be able to integrate optical CDMA easily with higher layer protocols such as TCP/IP. This design of optical CDMA does not constrain a designer to any particular technology, *e.g.*, a higher layer could be circuit or packet switched. However, the network protocol most likely to be used with this system is IP, with the any other needed functionality being provided by higher layer transport and application layers. The resulting architecture will allow the creation of low latency, low packet error rate access networks capable of providing variable quality of service at a low cost.

## 9.2 Future directions

Three areas of future work have been identified: enhancements to Interference Avoidance and application of Interference Avoidance mechanisms to two other domains: power control and switch design.

### 9.2.1 Analysis of Interference Avoidance

This section discusses further improvements that may be made to Interference Avoidance.

#### 9.2.1.1 Optimal transmission scheduling algorithms

An open area of research is whether new transmission scheduling algorithms may be defined which close the gap between the proposed algorithms and the upper bounds shown. An *optimal transmission scheduling algorithm* is an algorithm that maximizes the network throughput. A transmission scheduling algorithm based on a cooperative strategy can improve the throughput above that of pseudo-cooperative algorithms. One way of implementing a cooperative strategy is to use a *decomposition algorithm* that decomposes the state vector into its constituent codewords. It may be possible to construct codesets that guarantee decomposition of the state vector into unique codewords. To do this the codeset must consist of codewords that are linearly independent. The design of such codesets is an open area of research. To be useful, the algorithm must be able to perform decomposition in polynomial time (in terms of codeset length).

### **9.2.1.2 Delay**

Another area of research is study of delay and the effect of defer algorithms. When a node cannot transmit a packet, it needs to defer transmission. Algorithms to defer transmissions in single channel, shared medium networks have been studied extensively. Several backoff algorithms [5] such as binary exponential backoff and binary logarithm arbitration method have been proposed. Algorithms to control when a packet is transmitted have been classified into 1, non- or  $p$ -persistent algorithms [68]. The effect of these algorithms on the access latency of optical CDMA is an area for further study.

### **9.2.1.3 Fairness and quality of service**

An issue not addressed in this work is fairness. As the distance from the coupler increases the probability of errors in state estimation increases. Interference Avoidance could be unfair to nodes that are at greater distances from the coupler. A quantitative study of fairness and the tuning of parameters to ensure fairness are needed. Another issue that is not addressed in this work is quality of service. Optical CDMA with variable codeset parameters is ideally suited for providing differentiated service. State in such a system must be redefined to take into account variable codeset parameters. A quantitative study of variable codeset parameters and Interference Avoidance algorithms is an open area.

### **9.2.1.4 Error control**

Some improvements that may be added to Interference Avoidance include Interference Detection [27]. In Interference Detection, nodes have a receiver to receive packets that they have

transmitted. If an error is detected during transmission due to interference, the transmission is aborted. This is similar in principle to collision detection and could provide some performance improvement. Another area is the design of Forward Error Correction optimized for Interference Avoidance.

## **9.2.2 Architectural issues**

The architecture of the network will need to address issues such as automated network configuration, network management, and security. These issues have been studied in the context of existing networks. However, optical CDMA may require that these issues be re-evaluated. For example, this work assumed pre-existing configuration. Codewords were allocated to nodes and the assigned codewords were static. In reality, a automated bootstrapping mechanism for optical CDMA which handles codeword allocation may be needed. Such a mechanism may need a central server or a broadcast mechanism or both. The architecture will need to address such problems.

### **9.2.2.1 Other areas**

The analysis discussed in this work made several assumptions for ease of analysis. One assumption made was that synchronization phase losses were zero. An open area of research is to understand the impact of Interference Avoidance on synchronization loss. Does the line state created by this protocol improve or degrade the performance of synchronization algorithms? Another area of research is a quantification of the maximum tolerable error

in scheduling. Another interesting area is the behavior of a heterogeneous network, *i.e.*, a network where nodes use a mixture of transmission scheduling algorithms or thresholds.

### **9.2.3 Power control in optical CDMA LANs**

Power control is the mechanism used by nodes to tune the output power of their transmitters. This work assumed that the power transmitted by each node on the network is the same. Also, the power at the output of the coupler is the same for all users. In reality, this may not be true. Transmitter power constantly fluctuates, the fiber attenuates the signal at 0.2dB/km and there is loss in the coupler. Therefore, an optical CDMA network needs a power control mechanism. State estimation could be used as part of a distributed power control estimation algorithm where the nodes on the network use the state estimate to determine how many nodes are transmitting and whether the transmitter power needs to be adjusted before transmission. Further experiments and analysis are needed before it can be determined whether this is feasible.

### **9.2.4 Optical CDMA switch design**

Optical CDMA has been proposed as a technology for switch design [15], [25]. The primary use is to avoid output port contention in an all optical switch. Note that although optical CDMA multiplexing eliminates contention, it reduces the data rate. Though it eliminates output port contention, interference still causes packet errors at the output port. Transmission scheduling may be used at the output port to schedule packet transmission at the output port, thus reducing error losses. Delay lines are needed to buffer packets that cannot be scheduled.

Interesting areas to study include the comparison of the performance of this type of switch to a unbuffered switch, a study of the trade-offs between the number and length of the delay lines and the trade-off between output port contention and data rate.

## Reference List

- [1] A. H. Abdelmonem and T. N. Saadawi. Performance analysis of spread spectrum packet radio network with channel load sensing. *IEEE Journal on Special Areas in Communications*, 7(1):161–166, January 1989.
- [2] A. Allen. *Probability, statistics and queueing theory with computer science applications*. Academic Press, 1990.
- [3] M. Azizoglu, J. Salehi, and Y. Li. Optical CDMA via temporal codes. *IEEE Transactions on Communications*, 40(7):1162–1170, August 1992.
- [4] B. S. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan. Improving performance of TCP over wireless networks. In *International Conference on Distributed Computing Systems (IDCS)*, pages 365–373, May 1997.
- [5] D. Bertsekas and R. Gallager. *Data networks*. Prentice Hall, 1991.
- [6] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LAN's. In *SIGCOMM Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, 1994.
- [7] L. Brzozowski and E. Sargent. All-optical analog-to-digital converters, hard limiters, and logic gates. *IEEE /OSA Journal of Lightwave Technology*, 19(1):114–119, 2001.
- [8] G. Chockler, M. Demirbas, S. Gilbert, C. Newport, and T. Nolte. Consensus and collision detectors in wireless ad hoc networks. In *ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, July 2005.
- [9] H. Chung and P. V. Kumar. Optical orthogonal codes - new bounds and an optimal construction. *IEEE Transactions on Information Theory*, 36(4):866–873, July 1990.
- [10] H. Chung, J. Salehi, and V. K. Wei. Optical orthogonal codes: Design, analysis, and applications. *IEEE Transactions on Information theory*, 35(3):595–605, May 1989.
- [11] R. Fuji-Hara and Y. Miao. Optical orthogonal codes: Their bounds and new optimal constructions. *IEEE Transactions on Information Theory*, 46(7):2396–2406, November 2000.

- [12] S. Galli, R. Menendez, P. Toliver, T. Banwell, J. Jackel, J. Young, and S. Etemad. Novel results on the coexistence of spectrally phase encoded OCDMA and DWDM. In *IEEE International Conference on Communications (ICC)*, pages 1608–1612, May 2005.
- [13] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, 1979.
- [14] R. Gholizadeh and P. Saghari. Balanced detector for optical CDMA. In *Private communication*.
- [15] D. Gurkan, S. Kumar, A. Sahin, A. Willner, K. Parameswaran, M. Fejer, D. Starodubov, J. Bannister, P. Kamath, and J. Touch. All-Optical Wavelength and Time 2-D Code Converter for Dynamically-Reconfigurable OCDMA Networks Using a PPLN Waveguide. In *Optical Fiber Communication Conference (OFC)*, pages 9–11, Mar 2003.
- [16] C. S. Hsu and V. O. K. Li. Performance analysis of slotted fiber-optic code division multiple access (CDMA) packet networks. *IEEE Transactions on Communications*, 45(7):819 – 828, July 1997.
- [17] C. S. Hsu and V. O. K. Li. Performance analysis of unslotted fiber-optic code division multiple access (CDMA) packet networks. *IEEE Transactions on Communications*, 45(8):978 – 987, August 1997.
- [18] IEEE. Media access control (MAC) parameters, physical layer, repeater and management parameters for 1000Mb/s operation. *IEEE Std P802.3z/D5.0*, 1998.
- [19] Cable Television Labs Inc. Data Over Cable Service Interface Specifications (DOCSIS) 2.0, Radio Frequency Interface Specification. (SP-RFIV2.0), April 2004.
- [20] Little Optics Inc. 8-bit long path length programmable delay lines. *Press release*, January 2004.
- [21] J. Lee and P. Roux and U. Koc and T. Link and Y. Baeyens and Y. Chen. A 5-b 10-GSample/s A/D converter for 10-Gb/s optical receivers. *Journal of Solid-State Circuits*, 39(10):1671 – 1679, Oct 2004.
- [22] Kyle Jamieson, Bret Hull, Allen K. Miu, and Hari Balakrishnan. Understanding the Real-World Performance of Carrier Sense. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, Philadelphia, PA, August 2005.
- [23] S. Johnson. A new upper bound for error correcting codes. *IRE Transactions on Information Theory*, pages 203–207, April 1962.
- [24] G. Judge and F. Takawira. Spread Spectrum CDMA Packet Radio MAC Protocol using Channel Overload Detection and Blocking. *Wireless Networks*, 6:467–479, December 2000.

- [25] K. Kamakura, O. Kabranov, D. Makrakis, and I. Sasase. OBS networks using optical code division multiple access techniques. In *IEEE International Conference on Communications*, volume 3, pages 1725–1729, June 2004.
- [26] P. Kamath, J. D. Touch, and J. A. Bannister. Algorithms for interference sensing in optical CDMA networks. In *IEEE International Conference on Communications*, volume 3, pages 1720–1724, June 2004.
- [27] P. Kamath, J. D. Touch, and J. A. Bannister. The need for media access control in optical CDMA networks. In *IEEE Infocom*, volume 4, pages 2208–2219, March 2004.
- [28] L. Kazovsky, S. Benedetto, and A. Willner. *Optical fiber communication systems*. Artech House, 1996.
- [29] A. Keshavarzian and J. A. Salehi. Optical orthogonal code acquisition in fiber optic CDMA systems via simple search method. *IEEE Transactions on Communications*, 50(3):473–483, March 2002.
- [30] A. Keshavarzian and J. A. Salehi. Multiple-shift code acquisition of optical orthogonal codes in optical CDMA systems. *IEEE Transactions on Communications*, 53(4):687–697, April 2005.
- [31] Kyeong Soo Kim. On the evolution of PON-based FTTH solutions. *Information Sciences, Informatics and Computer Science: An International Journal*, 149(1-3):21–30, 2003.
- [32] L. Kleinrock. Packet switching in radio channels: Part I- Carrier Sense Multiple Access Modes and their Throughput-Delay characteristics. *IEEE Transactions on communications*, 23(12):1400–1414, Dec 1975.
- [33] L. Kleinrock. *Queueing systems Volume I: Theory*. John Wiley, 1975.
- [34] L. Kleinrock. *Queueing systems Volume II: Computer Applications*. John Wiley, 1976.
- [35] P. V. Kumar, R. Omrani, J. Touch, A. Willner, and P. Saghari. Three-fold increase in the number of active users in an O-CDMA system using novel code cycle modulation. *Under submission*, 2004.
- [36] W. Kwong, P. Prucnal, and P. Perrier. Synchronous versus asynchronous cdma for fiber-optic lans using optical signal processing. In *Global Telecommunications Conference (GLOBECOM)*, volume 2, pages 1012–1016, November 1989.
- [37] Bell labs. Bell labs scientists set world record, transmitting 160 billion bits of data over 300 kilometers on a single color of light. *Press release*, <http://www.lucent.com/press/1199/>, Nov 1999.

- [38] C. F. Lam. To spread or not to spread: The myths of optical CDMA. In *IEEE Lasers and Electro-Optics Society Annual Meeting*, volume 2, pages 810–811, 2000.
- [39] S. Lee and D. Green. Performance analysis of optical orthogonal codes in CDMA LANs. In *IEE Proceedings*, 1998.
- [40] B. Little and S. Chu. Towards very large scale integrated photonics. *Optics and photonics news*, pages 24–29, November 2000.
- [41] M. A. Marsan and D. Roffinella. Multichannel local area network protocols. *IEEE Journal on Selected Areas in Communications*, 1(5):885–897, Nov 1983.
- [42] A. Mendez, R. Gagliardi, H. Feng, J. Heritage, and J. Morookian. Strategies for realizing optical CDMA for dense, high-speed, long span, optical network applications. *Journal of Lightwave Technology*, 18:1685 – 1696, Dec 2000.
- [43] P. Mitra and J. Stark. Nonlinear limits to the information capacity of optical fibre communications. *Nature*, (411):1027 – 1030, 2001.
- [44] M. Molle. A new binary logarithmic arbitration method for Ethernet. *Technical report CSRI-298, Computer Systems Research Institute, Toronto, Canada*, 1994.
- [45] O. Moreno, Z. Zhang, P.V. Kumar, and V. Zinoviev. New constructions of optimal cyclically permutable constant weight codes. *IEEE Transactions on Information Theory*, 41(2), March 1995.
- [46] J. Muckenheimer and K. Iversen. A new design method for optical CDMA computer networks. In *IEEE ISCC*, 1997.
- [47] J. Muckenheimer, K. Iversen, and D. Hampicke. Construction of high-efficient optical CDMA computer networks: Statistical design. In *IEEE International Conference on Communications*, volume 3, pages 1289–1293, June 1998.
- [48] B. Mukherjee. WDM based local lightwave networks. Part I: Single hop systems. *IEEE Network Magazine*, 6(3):12–27, May 1992.
- [49] L. B. Nelson and H. Vincent Poor. Performance of multiuser detection for optical CDMA. Part 1: Error probabilities. *IEEE Transactions on Communications*, 43(11), Nov 1995.
- [50] T. Nguyen and G. Armitage. Experimentally derived interactions between TCP traffic and service quality over DOCSIS cable links. In *IEEE GLOBECOM Global Telecommunications Conference*, pages 1314–1318, Nov 2004.
- [51] NLANR. Traffic traces from an OC48 link. [www.nlanr.net](http://www.nlanr.net), March 2005.

- [52] R. Omrani, P. Elia, and P. V. Kumar. New constructions and bounds for 2-D optical orthogonal codes. *SETA*, pages 389–390, 2004.
- [53] Vern Paxson and Sally Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.
- [54] P. Prucnal, M. Santoro, and T. Fan. Asynchronous multiplexing for an optical fiber local area network. *IEEE Journal of Lightwave Technology*, 4:547, 1986.
- [55] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.
- [56] S. Adi Reddy and L. Tong. Exploiting decentralized channel state information for random access. *IEEE Transactions on Information Theory*, 51(2):537–561, February 2005.
- [57] J. Salehi. Code division multiple-access techniques in optical fiber networks - Part 1: Fundamental principles. *IEEE Transactions on Communications*, 37(8):824–833, August 1989.
- [58] J. Salehi and C. Brackett. Code division multiple-access techniques in optical fiber networks - Part 2: Systems performance analysis. *IEEE Transactions on Communications*, 37(8):834–842, August 1989.
- [59] J. Salehi, A. Weiner, and J. Heritage. Coherent ultrashort light pulse code division multiple access communication systems. *IEEE Journal of Lightwave Technology*, 8(3):478–491, March 1990.
- [60] J. Shah. Optical CDMA. *Optics and photonics news*, pages 42–47, April 2003.
- [61] H. M. H. Shalaby. The capacities of optical CDMA communication channels with different code-correlation constraints. *IEEE International Symposium on Spread Spectrum Techniques and Applications*, 1:179–183, Sep 2000.
- [62] H. M. H. Shalaby. Complexities, error probabilities and capacities of optical OOK-CDMA communication systems. *IEEE Transactions on Communications*, 50(12), Dec 2002.
- [63] H. M. H. Shalaby. Optical CDMA random access protocols with and without pretransmission coordination. *IEEE Journal of Lightwave Technology*, 21(11), Nov 2003.
- [64] T. Shepard. Decentralized channel management in scalable multihop spread-spectrum packet radio networks. *PhD. Thesis*, (MIT/LCS/TR-670), 1995.
- [65] D. Son, B. Krishnamachari, and J. Heidemann. Experimental analysis of concurrent packet transmissions in low-power wireless networks. *Information Sciences Institute Technical Report (ISI-TR-609)*, November 2005.

- [66] Sprint ATL Labs. Packet size distributions from an internet backbone. *www.sprintlabs.com*, March 2005.
- [67] L. Tancevski and I. Andonovic. Hybrid wavelength hopping/time spreading schemes for use in massive optical networks with increased security. *IEEE Journal of Lightwave Technology*, 14(12):2636–2647, December 1996.
- [68] F. A. Tobagi and V. B. Hunt. Performance analysis of carrier sense multiple access with collision detection. *Computer Networks*, 4(5):245–259, October 1980.
- [69] L. Tong, Q. Zhao, and G. Mergen. Multipacket reception in random access wireless networks: From signal processing to optimal medium access control. *IEEE Communications Magazine*, Nov 2001.
- [70] K. Toshimitsu, T. Yamazoto, M. Katayama, and A. Ogawa. A novel spread slotted Aloha system with channel load sensing protocol. *IEEE Journal on Special Areas in Communications*, 12(4):665–672, May 1994.
- [71] X. Wang and K. Kitayama. Analysis of beat noise in coherent and incoherent time spreading OCDMA. *Journal of Lightwave Technology*, 22(10):2226–2235, October 2004.
- [72] A. Woo, K. Whitehouse, X. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect for collision detection and recovery. In *IEEE Workshop on Embedded Networked Sensors (EmNetS)*, May 2005.
- [73] Xilinx Inc. Virtex II Pro / Virtex II Pro X Platform FPGAs: Data sheet. *http://www.xilinx.com*, October 2005.
- [74] F. Xue, Z. Ding, and B. Yoo. Performance analysis for optical CDMA networks with random access schemes. In *Global Telecommunications Conference (GLOBECOM)*, November 2004.
- [75] J. Zhang, M. Hu, and N. B. Shroff. Bursty traffic over CDMA: Predictive MAI temporal structure, rate control and admission control. *Computer Networks Journal*, 42(6):779–796, Aug 2003.

## Appendix A

### State representation

This appendix defines a representation for the state of the line. The representation allows analysis of the transmission scheduling algorithms.

Consider a shared medium optical CDMA network where multiple nodes are transmitting packets simultaneously. Figure A.1 shows a snapshot of the codewords on the line for a network using a codeset  $(N, w, \kappa) = (8, 3, 3)$ . The state of the line is shown below the codewords. Below the state, the chips of the state have been rearranged for convenience. Chip positions with overlapping ‘1 chips’ have been moved to the right, chip positions with ‘0 chips’ have been moved to the left. The state of the line can be divided into three sections: the *overlap section*, *i.e.*, the chip positions with overlapping chips, the *ones section*, *i.e.*, the section with ‘1 chips’ and the *zeroes section*, *i.e.*, the section with ‘0 chips’.

The state of the line can be represented by a pair  $(n_0, n_1)$  where  $n_0 < N$  and  $n_1 < N$ .  $n_0$  is the length of the zeroes section and  $n_1$  is the length of the ones section. The size of the overlap section is  $n_{ov} = N - (n_1 + n_0)$ . The term *state* refers to this reduced representation of the state of the line.

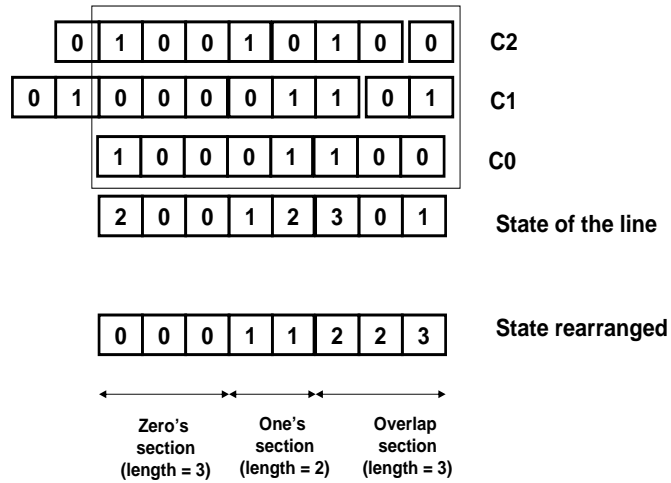


Figure A.1: State can be represented mathematically by a pair  $(n_0, n_1)$  where  $n_0$  is the size of the zeros section and  $n_1$  is the size of the ones section.

The state of the line could be any value  $(n_0, n_1)$  where  $0 \leq n_0 \leq N$  and  $0 \leq n_1 \leq N$ . A *valid state* is defined as a state where  $n_0 + n_1 \leq N$ . All other states are *invalid*. The *initial state* is defined as the state of the line when no codewords are on the line, *i.e.*,  $(N, 0)$ . A *reachable state* is defined as a state that can be reached from the initial state by a series of state transitions due to packet arrivals. The initial state  $(N, 0)$  is, by definition, a reachable state. The set of reachable states depends on the transmission scheduling algorithm.

When there are no codewords on the line, *i.e.*, no node is transmitting packets, the state at a point on the line is the initial state,  $(N, 0)$ , *i.e.*,  $N$  zeroes and 0 ones and overlaps. The

arrival of a single codeword adds  $w$  ‘1 chips’ to the state of the line and the state changes. This is called a *state transition*.

Let the start state of a state transition be  $(from_{n0}, from_{n1})$  and the destination state be  $(to_{n0}, to_{n1})$ . A state transition may be caused only by a packet arrival or departure<sup>1</sup>. When a codeword is added to the line,  $w$  ‘1 chips’ are added. A ‘1 chip’ could go to the zeroes section, ones section or the overlap section. Let the number of ‘1 chips’ going to the zeroes, ones and overlap sections be  $c_0$ ,  $c_1$  and  $c_{overlap}$  respectively.

Then:

$$c_0 = from_{n0} - to_{n0}$$

$$c_1 = to_{noverlap} - from_{noverlap}$$

$$c_{overlap} = w - (c_0 + c_1)$$

A *valid transition* is defined as a transition where the start state and the destination state are valid, reachable states and:

$$c_0 \geq 0$$

$$c_1 \geq 0$$

$$c_{overlap} \geq 0$$

An *admissible transition* is defined as a valid transition which is permitted by the transmission scheduling algorithm. A *same state transition* is defined as a transition from a state to itself.

---

<sup>1</sup>ON-OFF keyed modulation does not affect the state. The state is defined as the sum of codewords when all codewords are transmitting 1 data bits.

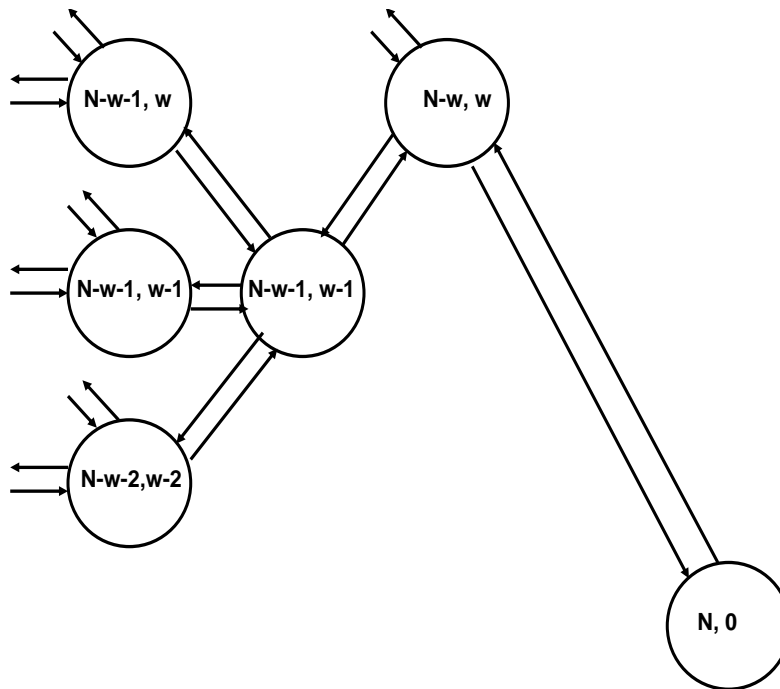


Figure A.2: The initial few states of a state transition diagram.

For example, Aloha-CDMA allows a transition from any state to any state, with the constraint that  $w$  chips must be added to the line during an arrival. A selfish algorithm allows only those transitions where at least one ‘1 chip’ goes to the zeroes section, *i.e.*,  $c_1 > 0$ .

A *state transition diagram* can be drawn based on the admissible transitions. Figure A.2 shows a portion of a state transition diagram. Because of the large size of the diagram, only a few states are shown. Invalid states are not indicated in the diagram. The diagram shows the initial state  $(N, 0)$  and state transitions to a few states.

## Appendix B

### Packet error rate and number of codewords multiplexed on the line

This appendix explains how to calculate the probability of error and the number of codewords on the line when the system is in state  $(n_0, n_1)$ .

#### B.1 Packet error rate

The probability of error depends on the state. For a state  $(n_0, n_1)$ , the probability of error is the probability that for any codeword, all its '1 chips' overlap with other '1 chips' or overlaps. To calculate the probability of error, the locations of the '1 chips' in the codeword are assumed to be chosen uniform randomly, *i.e.*, the codeset construction uses  $\kappa = w$ . For any selfish algorithm, in state  $(n_0, n_1)$ , if  $N_{online}(n_0, n_1)$  codewords are multiplexed at a point on the line, then at most  $n_1$  codewords are transmitted without error. Therefore, the probability of packet error can be approximated by the expression below:

For valid states:

$$P_e(n_0, n_1) = 0 \quad \text{when } N_{online}(n_0, n_1) \leq n_1$$

$$= \frac{(N_{online}(n_0, n_1) - n_1)}{N_{online}(n_0, n_1)} \quad \text{otherwise}$$

For invalid states:

$$P_e(n_0, n_1) = 0$$

Note that a packet is considered lost if there are other codewords on the line whose chips overlap with all  $w$  chips of the packet's codeword. The other codewords on the line are assumed to be ON, *i.e.*, transmitting 1 bits although that may not necessarily be true. Therefore the calculated packet error rate is the worst case packet error rate.

## B.2 Number of codewords multiplexed on the line

The state transition diagram is used to calculate the number of codewords multiplexed at a point on the line.

Consider a graph where the nodes of the graph are the states and each state transition due to an arrival forms a directed edge (neglect same state transitions). Appendix D shows that for any selfish transmission scheduling algorithm, this graph is a directed acyclic graph. In the graph, each edge represents the arrival of exactly one packet. Therefore the number of codewords on the line for a state  $(n_0, n_1)$  depends on the number of edges from the initial state  $(N, 0)$  to  $(n_0, n_1)$ . Note that there may be several paths (possibly of different lengths)

between the initial state and the desired state. Because the graph is a DAG, the length of the longest path from the initial state to that state is an upper bound on the number of codewords on the line when the line is in that state. The length of the shortest path from the initial state  $(N, 0)$  to that state is a lower bound on that number. The length of the longest path is an upper bound on that number.<sup>1</sup> Because the graph is a DAG, both the shortest and the longest path may be calculated in polynomial time [13].

Therefore, for valid states:

$$N_{online}(n_0, n_1) \geq \text{ShortestPath}((N, 0), (n_0, n_1))$$

$$N_{online}(n_0, n_1) \leq \text{LongestPath}((N, 0), (n_0, n_1))$$

For invalid states:

$$N_{online}(n_0, n_1) = 0$$

---

<sup>1</sup>In a real system the actual number of codewords on the line is between these two values and its distribution depends on the transmission scheduling algorithm. This work assumes the number of codewords multiplexed on the line is equal to the lower bound. Therefore, it is a worst case assumption and the calculated throughput is a lower bound on the achievable throughput.

## Appendix C

### Equilibrium state probabilities

The performance of the transmission scheduling algorithms depends on the equilibrium state probabilities. This section calculates the equilibrium state probabilities.

#### C.1 Assumptions

As mentioned earlier it is assumed that perfect state estimation is possible by all nodes and the only reason for a state transition is an arrival or a departure of a packet.

Packet arrivals are assumed to be Poisson arrivals and packet lengths are exponentially distributed. Assume that codeword selection is random, *i.e.*, the particular codeword that arrives is uniformly chosen from the codeset. Under this assumption, the probability of transitioning to a particular state on an arrival is dependent only on the current state and not on the path taken to reach that state. However, the probability of transitioning to a particular state on a departure does not have this property. On a departure the system returns to a state that depends on the remaining packets multiplexed on the line. However, if it is assumed that the probability of departure to a state is proportional to the rate of arrival from that particular

state then the next state is dependent only on the current state and not on the path taken to reach that state. If the probability of departure to a state is proportional to the probability of arrival then on an average the trajectory followed by a departure follows that of an arrival. Under these circumstances, the state diagram for arrivals and departures is a Markov chain. Equilibrium probabilities may be found by solving the balance equations for the system.

The following are the steps needed to determine the rates of transition between two states ( $from_{n0}, from_{n1}$ ) and ( $to_{n0}, to_{n1}$ ): First, it must first be determined whether the transition is admissible or not. Then the probability of an arrival or departure transition between the two state must be calculated. This probability can be used to determine the transition rates. The next few sections describe how to determine whether a transition is admissible and if it is, how to calculate the transition probabilities and write the balance equations.

## **C.2 Admissible transmissions**

This section describes how to identify admissible state transitions given a codeset and a transmission scheduling algorithm.<sup>1</sup>

### **C.2.1 Aloha-CDMA**

For Aloha-CDMA, all transitions are admissible.

---

<sup>1</sup> It is assumed that if a packet cannot be transmitted upon arrival, the system undergoes a same state transition and the packet is dropped. Note that during a same state transition the number of packets on the line does not change. Only packet arrivals which arrive at times instants that allow admissible transitions are allowed on the line.

### C.2.2 Pure selfish scheduling

A pure selfish algorithm schedules transmission such that at least 1 '1' chip goes to the zeroes section.

Therefore a transmission is admissible if,

$$c_0 \geq 1$$

### C.2.3 Threshold scheduling

The threshold scheduling algorithm schedules transmission such that at least 1 '1' chip goes to the zeroes section and the number of overlaps (the size of the overlap section) is below a threshold. The threshold can be expressed as a fraction of the length of the codeword  $\alpha$ .

Therefore, a transmission is admissible if:

$$c_0 \geq 1$$

$$t_{O_{overlap}} \leq \alpha N$$

### C.2.4 Overlap section scheduling

The overlap section scheduling algorithm schedules transmission such that at least 1 '1' chip goes to the zeroes section and the number of overlaps (the size of the overlap section) is below the number of ones (the size of the ones section).

Therefore, a transmission is admissible if:

$$c_0 \geq 1$$

$$t_{\text{nooverlap}} \leq t_{\text{n1}}$$

### C.3 Arrival state transition probabilities

The arrival and departure probabilities for state transitions can be calculated, based on whether a transition is admissible or not. Two matrices are generated: an arrival probability matrix and a departure transition matrix whose elements are the state transition probabilities,  $P_{\text{arrival}}(n_0, n_1, s_0, s_1)$  and  $P_{\text{departure}}(n_0, n_1, s_0, s_1)$ .

The arrival rates from a state ( $from_{n0}, from_{n1}$ ) to a state ( $to_{n0}, to_{n1}$ ) can be calculated as below.

For an admissible transition:

$$P_{\text{arrival}}(from_{n0}, from_{n1}, to_{n0}, to_{n1}) = N_{\text{admitted}}/N_{\text{total}}$$

where:

$$N_{\text{admitted}} = \begin{pmatrix} from_{n0} \\ c_0 \end{pmatrix} \begin{pmatrix} from_{n1} \\ c_1 \end{pmatrix} \begin{pmatrix} from_{\text{nooverlap}} \\ c_{\text{overlap}} \end{pmatrix}$$

$$N_{\text{total}} = \begin{pmatrix} N \\ w \end{pmatrix}$$

For a non-admissible transition:

$$P_{arrival}(from_{n_0}, from_{n_1}, to_{n_0}, to_{n_1}) = 0$$

For a same state non-admissible transition:

$$P_{arrival}(s_0, s_1, s_0, s_1) = 1 - \sum_{n_0=0}^N \sum_{n_1=0}^N P_{arrival}(s_0, s_1, n_0, n_1) \\ ((n_0, n_1) \neq (s_0, s_1))$$

## C.4 Departure state transition probabilities

As described in the assumptions, the departure probabilities from one state to another state are proportional to the arrival rates into the state. Therefore:

$$P_{departure}(s_0, s_1, d_0, d_1) = \frac{P_{arrival}(d_0, d_1, s_0, s_1)}{\sum_{n_0=0}^N \sum_{n_1=0}^N P_{arrival}(n_0, n_1, s_0, s_1)}$$

## C.5 Balance equations

When the system is in equilibrium, the flow into any state must equal the flow out of the state.

Therefore, for a valid, reachable state  $(s_0, s_1)$ :

$$(\lambda + N_{online}(s_0, s_1)\mu) P_{state}(s_0, s_1) = \sum_{n_0=0}^N \sum_{n_1=0}^N P_{state}(n_0, n_1) \lambda P_{arrival}(n_0, n_1, s_0, s_1) \\ + \sum_{n_0=0}^N \sum_{n_1=0}^N P_{state}(n_0, n_1) N_{online}(n_0, n_1) \mu P_{departure}(n_0, n_1, s_0, s_1)$$

Also,

$$\sum_{n_0=0}^N \sum_{n_1=0}^N P_{state}(n_0, n_1) = 1;$$

These equations can be solved for the equilibrium state probabilities  $P_{state}(s_0, s_1)$ . The variables  $n_0$  and  $n_1$  can take on values in the range  $[0, N]$ , subject to  $n_0 + n_1 \leq N$ . Therefore, there are  $N(N + 1)/2$  states. The analysis above provides a system of  $N(N + 1)/2$  linear equations. The system of linear equations is sparse because the number of states between which admissible transitions occur is much less than the total number of pairs of states. The equations can be solved rapidly using mathematical tool sets.

## Appendix D

### Arrival state transition diagram

*Lemma: For any selfish transmission scheduling algorithm, the directed graph constructed from the arrival state transition diagram is acyclic.*

Consider a directed graph where the vertices of the graph are the states and each state transition due to an arrival forms a directed edge.<sup>1</sup> Assume there exists a cycle in the graph.

Let the sequence of states in the cycle be

$$(N_0^0, N_1^0), (N_0^1, N_1^1), \dots, (N_0^i, N_1^i), (N_0^0, N_1^0)$$

By the definition of a selfish scheduling algorithm, for any admissible transition:

$$c_0 \geq 1$$

Consider any edge of the graph. The edge represents an admissible transition. Let the vertices be the states  $(N_0^i, N_1^i)$  and  $(N_0^{i+1}, N_1^{i+1})$ .

---

<sup>1</sup>Neglect same state transitions (for selfish scheduling, same state transitions do not represent arrivals to the line)

Therefore,

$$N_0^i - N_0^{i+1} = c_0$$

Therefore,

$$N_0^i > N_0^{i+1}$$

Therefore,  $N_0^0 > N_0^1 > \dots > N_0^n > N_0^0$  which is a contradiction. Therefore the graph cannot contain a cycle. Therefore it is a directed acyclic graph.

## Appendix E

### Throughput of Centralized Perfect Scheduling

This section determines the throughput of Centralized Perfect Scheduling (CPS). It is shown that its throughput is an upper bound to the performance of any selfish transmission scheduling algorithm. The proof is organized into three parts. First, *Centralized Perfect Scheduling* is defined. Next, it is shown that the throughput of this algorithm is greater than or equal to that of any selfish transmission scheduling algorithm. Finally, it is shown that the state model of this algorithm corresponds to a birth death process easily modeled by an  $M/M/1/K$  queuing system where  $K = N - (w - 1)$ . An expression for the average throughput of the centralized perfect scheduling is derived.

Consider a scheduling algorithm that operates as follows. Nodes can do perfect state estimation and perfect line state decomposition, *i.e.*, the line state can be decomposed uniquely into codewords on the line. Transmitters and receivers can both be tuned instantaneously to any codeword. When a packet is to be transmitted, there is a centralized mechanism (such as a centralized scheduler or control channel) which coordinates the allocation of a codeword

and a transmission delay to be used for transmission. The allocation is such that the codeword and delay ensure that

- The codeword does not destroy itself or any other codeword on the line, *i.e.*, the event is always a *Preserve self/Preserve others* event.
- It maximizes the number of codewords that can be added to the line after this transmission.

On arrival of the first codeword, any codeword may be allocated. The state of the line is now  $(N - w, w)$ . When the second packet arrives, it is allocated a codeword such that exactly one '1 chip' goes to the zeroes section and the remaining  $w - 1$  chips overlap with the other chips on the line. Thereafter, any codeword that is added has exactly one chip that goes to the zeroes section and the remaining  $w - 1$  go to the overlap section. This continues until no more codewords can be added. When this happens there are  $N - (w - 1)$  codewords on the line and the state is  $(0, N - (w - 1))$ . No further codewords are allowed on the line until a codeword departs. The next packet that arrives is allocated the codeword that departed and so on.

Consider a time instant case when the system (all nodes in the network) has  $N_{offered}$  packets to transmit. Under these condition, the transmission scheduling algorithm allows  $N_{online}$  packets on the line and the system enters a particular state  $(N_0, N_1)$ . The throughput of the system depends on the number of codewords on the line  $N_{online}$  and the probability of error  $P_e$  in that state. Two cases are considered: when  $N_{offered} > N - (w - 1)$  and when  $N_{offered} \leq N - (w - 1)$ .

$$\mathbf{E.1} \quad N_{offered} \geq N - (w - 1)$$

In this case for CPS:

$$N_{online} = N - (w - 1)$$

$$P_e = 0$$

Therefore:

$$Th = (N - (w - 1))/N \tag{E.1}$$

For any selfish transmission scheduling algorithm, when a codeword is added to the line the size of the zeroes section is reduced by at least one. The first codeword reduces it by  $w$ .

Therefore:

$$N_{online} \leq N - N_0 - (w - 1)$$

$$P_e \geq 0$$

$$Th \leq ((N - N_0 - (w - 1))/N)(1 - P_e) \tag{E.2}$$

From inspection of results E.1 and E.2, it can be seen that the throughput of the Centralized Perfect Scheduling is greater than or equal to the throughput of any other selfish transmission scheduling algorithm. The maximum throughput occurs in state  $(0, N - (w - 1))$ . The value of the maximum throughput is  $(N - (w - 1))/N$ .

**E.2**  $N_{offered} < N - (w - 1)$

In this case for CPS:

$$N_{online} = N_{offered}$$

$$P_e = 0$$

Therefore:

$$Th = (N_{offered})/N \tag{E.3}$$

For any other selfish transmission scheduling algorithm, the number of codewords on the line can be at most  $N_{offered}$  and the probability of error is greater than or equal to zero.

$$N_{online} \leq N_{offered}$$

$$P_e \geq 0$$

Therefore:

$$Th \leq ((N_{offered})/N)(1 - P_e) \quad (E.4)$$

From inspection of results E.3 and E.4, it can be seen that the throughput of the Centralized Perfect Scheduling is greater than or equal to the throughput of any other selfish transmission scheduling algorithms.

For any number of packets  $N_{offered}$  that the system has to transmit, the Centralized Perfect Scheduling results in the transmission of the maximum number of error free packets, *i.e.*, its throughput is greater than any other selfish algorithm.

The centralized perfect scheduling goes through a sequence of states  $(N, 0)$ ,  $(N - w, w)$ ,  $(N - (w - 1), 2)$ ,  $(N - (w - 2), 3)$ ..... $(0, N - (w - 1))$ . The state transition diagram reduces to a simple birth death Markov chain. It forms an  $M/M/1/K$  queuing system where  $K = N - (w - 1)$ . The state probabilities may be found easily using well known results [33]. The throughput is given by:

$$Th = \sum_{n=0}^{n=K} p(n) * n/N$$

where  $p(n)$  is the probability that  $n$  packets are on the line,

$$p(n) = \frac{(1 - (\lambda/\mu))(\lambda/\mu)^n}{1 - (\lambda/\mu)^{(K+1)}}$$

$$\begin{aligned}
Th &= \sum_{n=0}^{n=K} \frac{(1 - (\lambda/\mu))(\lambda/\mu)^n}{1 - (\lambda/\mu)^{(K+1)}} n/N \\
&= \frac{K(\lambda/\mu)^{K+2} - (K+1)(\lambda/\mu)^{K+1} + (\lambda/\mu)}{N(1 - (\lambda/\mu))(1 - (\lambda/\mu)^{K+1})}
\end{aligned}$$

## Appendix F

### Interference Avoidance is a generalization of Carrier Sensing and CDMA

Interference Avoidance is a generalization of Carrier Sensing. Carrier Sensing has been studied in single channel shared medium wired networks and wireless networks. A node senses the state of the network (idle, busy or collision) and schedules its transmission if the state is idle. Carrier Sensing is effective if the normalized propagation delay is low. The normalized propagation delay of a network  $a$  is the ratio of the propagation delay of the network to the packet transmission time. Carrier sensing can significantly reduce collision losses, for

Protocol	Length of codeset $N$	Weight of codeset $w$	State estimation algorithm	Transmission scheduling algorithm
Aloha-CDMA	Any	Any	None	Pure threshold scheduling $th = \infty$
Aloha	1	1	None	Pure threshold scheduling $th = \infty$
CSMA	1	1	On demand, $n_e = 1$	Pure selfish scheduling

Table F.1: Reduction of Interference Avoidance to the Aloha, Aloha-CDMA, and CSMA protocols.

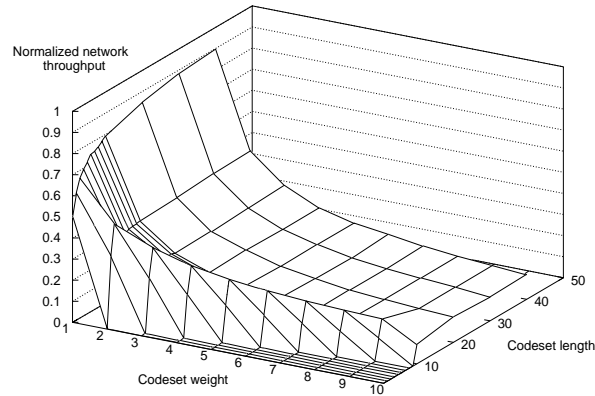


Figure F.1: The performance of Interference Avoidance vs. codeset length and codeset weight. The results are based on simulation. The graph shows the normalized network throughput at a normalized offered load of 1 and normalized propagation delay of  $a = 0$ . The traffic model is Poisson arrivals, exponentially distributed packet sizes.

certain values of  $a$  ( $0.01 < a < 0.2$ ) [68]. For a network of speed  $100Mb/s$ , packet length 1000 bytes, diameter 4000m  $a = 0.1$ . As normalized propagation delay increases (the speed of the network or the diameter of the network increases, or packet size decreases), the correlation of state reduces and the efficiency of carrier sensing decreases. For an optical CDMA network the normalized propagation delay is reduced by a factor of  $N$ . For a network of diameter 4000m, chipping rate 10 Gc/s, packet length 1000 bytes, codeset length  $N = 100$ ,  $a = 0.25$ . This is the reason why Interference Avoidance works. The codeset length increases the normalized propagation delay of a packet. This in turn increases the correlation of state.

By changing the value of four parameters: the codeset length  $N$ , the codeset weight  $w$ , the state estimation algorithm and the transmission scheduling algorithm, Interference Avoidance can be reduced to Aloha, Aloha-CDMA or to CSMA. As the codeset length  $N$  is reduced,

Interference Avoidance reduces to Carrier Sensing. Table F.1 shows the values of  $N$  and  $w$  for different media access control algorithms.

Figure F.1 shows the throughput of Interference Avoidance with selfish scheduling for different values of codeset length  $N$  and codeset weight  $w$  and a normalized propagation delay of  $a = 0$ . As  $N$  and  $w$  tend to 1, Interference Avoidance reduces to Carrier Sensing. The throughput at  $N = 1, w = 1$  is equal to 0.5. This is the value predicted for Carrier Sensing with non persistent sensing [68].