Algorithms for Interference Sensing in Optical CDMA Networks

Purushotham Kamath, Joseph D. Touch and Joseph A. Bannister

{pkamath, touch, joseph}@isi.edu

Information Sciences Institute, University of Southern California

Abstract

Optical CDMA Local Area Networks allow shared access to a broadcast medium. Every node is assigned an Optical Orthogonal Codeword (OOC) to transmit or receive on. Optical CDMA systems designed without any media access control have low throughput under moderate to heavy offered load due to interference between codewords. *Interference Sensing* is a media access architecture where nodes on the network sense the amount of interference on the line before transmission. Nodes defer transmissions if there is interference on the line, thereby improving network throughput. We discuss and analyze three algorithms for interference sensing. Through simulation we show that these algorithms reduce or eliminate throughput degradation at high loads. We conclude that algorithms that attempt to eliminate interference completely by determining the exact codewords on the line can be inefficient, given the nature of packet arrivals and lengths. Simpler algorithms such as selfish and threshold based algorithms can result in negligible (less than 5%) degradation from the maximum theoretical throughput. We validate our analysis through simulation with realistic network traffic traces.

KEYWORDS: System design

[1]This material is based upon work supported by the Defense Advanced Research Projects Agency under contract no. N66001-02-1-8939 issued by the Space and Naval Warfare Systems Center (SPAWAR). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, SPAWAR, or the U.S. Government.

I. INTRODUCTION

Code Division Multiple Access (CDMA) has been widely used as a multichannel access technology in wireless networks such as the cellular phone system for several years because of its resilience to multiuser interference and graceful degradation under heavy load. Its use on an optical link has been studied extensively [1], [2], [3]. However, several concerns have been expressed about the use of spread spectrum on an optical link due to low network throughput [4].

The primary difference between wireless and optical CDMA is that optical fiber is an intensity medium. Hence, binary data is sent using pulses of light. A pulse of light is used to signal a 1 and the absence of a pulse signals a 0.

An Optical Orthogonal Code (OOC) set is a set of (0,1) sequences of length N that satisfies certain autocorrelation and cross-correlation constraints. The term *codeset* is used to refer to the set of sequences, while the term *codeword* is used for a member of the set. Each 0 or 1 of a sequence is called a *chip*, while the sequence represents a data *bit*. The number w of 1 chips of a codeword of the codeset is called its Hamming weight. This paper considers *constant weight* codesets, i.e. codesets with all codewords having the same weight. Most OCDMA networks are ON-OFF keyed optical CDMA networks i.e. the presence of a codeword signifies an 1 and absence signifies a 0.

Most codeset designs limit the autocorrelation λ_a and cross-correlation λ_c of the codeset to a fixed value, the *Maximum Collision Parameter* κ or λ. Most optical CDMA networks are designed to use $\kappa = 1$ or 2, to ensure that interference between codewords is low. A particular codeset is specified by the parameters (N, w, κ). The implication of designing under the correlation constraints is that only a fraction of the available bandwidth is utilized. If we attempt to increase the bandwidth utilization by increasing the number of codewords on the line, the interference between codewords increases, packets are lost and the network throughput falls.

To summarize, optical CDMA allows nodes to transmit asynchronously without any media access delay. However, it has two disadvantages: low spectral efficiency and low throughput under heavy loads. At high

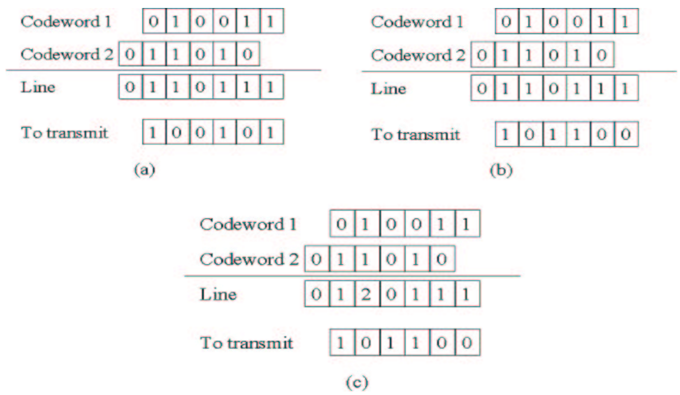

Fig. 1. Examples of interference between codewords

offered loads, the cause for low throughput is the interference between codewords.

II. MOTIVATION

The throughput of an optical CDMA network at any instant of time depends on the codewords that are on the line at that instant. The interference depends on the exact codewords on the line and their phase shifts with respect to each other. E.g. consider the 6 chip length codewords shown in Figure 1(a). The figure is a snapshot of a single data bit from three packets i.e 2 nodes are currently transmitting on codewords 1 and 2 and a third node is going to transmit (on codeword 3). Codeword 1 has been sent 1 chip later than codeword 2. The figure represents an instant in time when all three packets had a 1 data bit on the line. A single data bit (i.e. one codeword) is shown in the figure - the packets could contain several 0 and 1 data bits, extending in both directions. The packets sent on codewords 1 and 2 can be transmitted without any problem under the phase shifts shown. However if a packet with codeword 3 were to be transmitted with the phase shift shown, it would not be received properly. Codewords 1 and 2 would interfere with the reception of codeword 3. Consider a case where the third node sends a 0 bit on codeword 3 and codewords 1 and 2 are transmitting 1s. The detector for codeword 3 would detect it as a 1 bit. The checksum on the packet would fail and it would be discarded. On the other hand, if the packet on codeword 3 was sent three chip times later, the three packets could be transmitted without interfering with each other as in Figure 1(b) (codeword 3 is shown phase shifted by 3 chip times).

Previous optical CDMA network designs used Aloha-CDMA i.e. there was no form of media access control. Under such conditions throughput degrades rapidly at high loads, dropping close to zero under 100% offered load [7]. [7] also illustrated that the throughput of optical CDMA under heavy loads can be improved by simple media access mechanisms that prevent interfering codewords from being sent simultaneously, a mechanism called *Interference Sensing / Interference Detection (Is/Id)* An optical CDMA network was modeled and a probabilistic analysis showed that interference sensing would prevent throughput degradation at moderate to high load at the cost of low delays. The contribution of this paper is to describe and analyze three different interference sensing based media access algorithms. In this paper we take a simulation based approach to analyze the algorithms.

III. System architecture

The following sections describe the network architecture, the addressing, code allocation mechanism and the design of a receiver. A much more elaborate description of the system architecture may be found in [7].

The network we describe is a broadcast star coupler based system. Every node is equipped with at least one transmitter and one receiver. The transmitter and receiver may be tuned to any codeword. Every node has a unique *node address* which is distinct from the codewords in the codeset. The packet header has a preamble to allow nodes to detect the start of a packet and an error detection mechanism such as a checksum to detect corrupted packets. For simplicity, we assume a tunable transmitter-fixed receiver system, where nodes choose which codeword to receive on when they start up. The codeword chosen is a hash of the node address. The hash function is known to all nodes, so a node that wishes to transmit to another node can determine the codeword on which to transmit using the node address of the receiver and the hash function.

Our design means that nodes will share codewords, i.e several nodes may receive on the same codeword at a time. Any codeword or any of its cyclic shifts may be on the line at any time. Nodes accept or discard packets they receive based on the node address in the packet header. Each receiver is tuned to a particular optical CDMA codeword. It continuously listens for that codeword and as soon as it successfully detects the packet preamble, it continues to listen for a packet and performs a checksum operation on the packet once it has been completely received. If two nodes transmit to a single node at around the same time, the receiver receives the first packet and synchronizes to it and the second packet is dropped. Alternatively if a node has multiple receivers, this event can be avoided. Our analysis assumes that such packet drops are relatively infrequent events. We assume that the receivers do not do any form of power limiting.

Since the network is a broadcast network, every node sees every transmission. We assume that every node sees the same superimposed data on the line, as a result of using a star coupler. That data stream is delivered to nodes at different times, due to variations in propagation delays from the coupler to the nodes.

IV. Interference sensing

Interference sensing is a media access mechanism where nodes sense the interference on the line and determine whether to transmit or defer. The term refers both to the action of sensing information from the line and to the algorithm used to decide whether to transmit or defer.

A. Sensing algorithms

The sensing algorithm should allow a node to determine as much information about the conditions of the line in as short a duration as possible. It tries to ensure that the interference between codes on the line is minimal and that media access delays are kept within bounds.

At any instant, the combination of codes being transmitted by all nodes on the network is referred to as the *state of the line*. As indicated in Fig 1, the state of the line is a superposition of all other codes currently being transmitted. The state is measured over an interval equal to the length of the code N. E.g. In Fig 1(a) the code length is 6 and the state of the line measured by the node about to transmit is 110111. Note that in the next *chip* time slot, the state will change, because the codes have been shifted by 1. Thus the state of the line is dependent on the exact instant of time when it is measured. However one *data bit* time slot later, the state will be the same unless next codewords arrive or old codewords leave. State can be measured at any time i.e. the system is asynchronous. Thus, the state of the line is the superposition of several codewords, each codeword possibly shifted by different amounts.

When a node transmits a codeword, one of the following events will occur:

1) This node's transmission will be received without error and it will not interfere with other codewords on the line.
2) This node's transmission will be received without error, but this node's transmission (possibly in combination with some other node's transmissions) will cause some other node's transmission to be incorrectly received due to interference.
3) This node's transmission will fail due to interference from other codewords. However its own transmission does not interfere with other codewords on the line.
4) This node's transmission will fail due to interference from other codewords and its own transmission interferes with other codewords on the line.

For the purpose of defining our algorithms we merge events (3) and (4) into a single event (3a) i.e. this node's transmission will fail and it may/may not cause interference with other node's transmissions. Ideally we would like to achieve the first result.

Based on these conditions we define three sensing algorithms:

- Selfish: A nodes transmits if the state of the line will not cause interference with its codeword.
- Selfish with thresholds: A node transmits if the amount of interference between its codeword and the state of the line is below a threshold.
- Code estimation: A node transmits only if its transmission will not impact any other codeword on the line.

B. Transmission instant

Once a node has sensed the state of the line, it must determine the exact instant to transmit its data. The time (in chip times measured from the start of a state bit) at which transmission is identified to be possible is called the *departure instant*. Here, the term instant refers to a particular chip time at which transmission may take place. If a node detects interference, it defers transmission of data. The duration for which the node defers depends on the state of the line. E.g. In Figure 1(a, b) at t=0, node 3 senses interference, however 3 chip times later there is no interference. Hence the departure instant is sensed to be t=3. The packet is transmitted exactly 1 data bit + 3 chip times later. A single sense operation may indicate N potential departure instants for the packet, one of which may be chosen either randomly or some other decision algorithm.

The node may transmit either at the next data bit time or an integral number of data bits later. The state of the line is likely to be the same an integral number of data bits later, unless there are packet arrivals or departures. Therefore the interference sensing algorithm is likely but not guaranteed to prevent most of the potential interference. In our analysis we assume that the data will consist of as many 0s as 1s. 4B/5B encoding can be used to prevent long sequences of 0s or 1s. Therefore sensing over a window is likely to obtain an accurate estimate of the state of the line.

C. Effectiveness of Interference Sensing

The effectiveness of the interference sensing algorithm depends on several factors:

- Code set: The codeset parameters
  - The code length determines the transmission time of a packet and the code length and weight determine how many simultaneous codewords can be on the line without interference
  - The cross correlation parameter κ dictates the number of codes and therefore the accuracy of code estimation
- Sensing algorithm: Sensing can be done in two ways - either *on demand* i.e. when a packet arrives for transmission, sensing is started. Alternatively a node may sense the medium in *continuous* mode, building an estimate of the state of the line that is used on packet arrivals.

  If the sensing window length is too short the estimate will be inaccurate, if it is too long packet arrivals or departures may change the state of the line. The sensing window is important for threshold based sensing and the code estimation algorithms since they try to estimate the state of the line.

- Defer algorithm: After detecting interference a node may continue to sense the medium (1-persistence) or back off (non-persistence or p-persistence). Conventional CSMA/CD [8] protocols use persistence mechanisms (non-persistent, 1-persistent or p-persistent) to handle the case when a node senses the medium as busy. The purpose of the persistence mechanisms is to reduce to probability of multiple nodes accessing the medium at the same time. With CDMA, packet lengths are longer due to the spreading of the data bits. However multiple packets will be on the line simultaneously. Therefore the probability of sensing interference after a back off may be sufficiently different from the probability of sensing a carrier after back off in a single channel CSMA/CD type protocol. If the sensing is not 1-persistent, a node may choose to resume sensing after a *back off interval*. It can attempt to do this until a *retry limit* is exceeded. This issue is discussed again with respect to interference detection in Section VI.

It is interesting to note that the media access delay does not necessarily mean that packets suffer queuing delays. Packets can be transmitted out of order. E.g. a node may have to transmit two packets on two different codewords. Interference on the line may prohibit the sending of the first packet, but may allow the second packet to be transmitted.

V. Algorithms for Interference Sensing

We describe three algorithms for Interference Sensing based media access: *Selfish media access*, *Code estimation based media access* and *Selfish with threshold based media access*.

A. Selfish media access

In selfish media access, a node transmits if the state of the line will not cause interference with its codeword. If it does, the node defers until the state of the line allows transmission. The node does not check whether the transmission of its codeword would interfere with other codewords on the line.

Note that there may still be interference if another node begins its transmission during the sensing window and it is not sensed due to finite propagation delay. This issue will be discussed further in Section VI. Sensing of the line may be done over a window of several bits. The effect of the duration of the sensing window is discussed in Section VII-D.

A node senses the line for a duration of time equal to a single data bit i.e. N chips, where N is the length of the codewords. This represents a single state reading. After each sensing, the node determines the *cumulative state of the line* i.e. the sum of the states sensed so far ($state = state|linereading$). Since there may be packet arrivals and departures and data consists of zeros and ones, the cumulative state ensures that interference over a window is taken into consideration. The selfish media access algorithm is below:

```
Sense the line and determine the cumulative state of the line.
Set departure instant t_d = 0
While (t_d < N)
    Check whether the state of the line would destroy this codeword
        i.e. if (state & code ≠ code) then break
    Rotate the line state
    Increment departure instant
If t_d < N, then defer till departure instant and then transmit data
If t_d > N, then continue sensing/back off according to the defer algorithm
```

B. Selfish with threshold based media access

In contrast to the selfish media access, a threshold based mechanism attempts to limit the amount of interference caused to other users, by reducing the probability with which a node's transmission will interfere with other transmissions.

Interference may be limited in two ways:

- By limiting the number of overlapping chips
- By limiting the magnitude of each overlap (i.e how many chips overlap at a single slot)

To limit interference we define two thresholds: the *overlap count limit*, $thresh_c$ i.e. the maximum number of the overlaps between the codeword and the state of the line. and *overlap magnitude limit*, $thresh_m$ i.e.

the maximum magnitude of a single overlap between the codeword and the state of the line.

A node first performs the selfish test to determine whether it can transmit without damage to its own codeword. It then checks the overlap count and overlap magnitudes between its codeword and the state of the line. If the resulting state is below the thresholds, the node transmits its codeword. If not the node defers transmission until it is.

Decreasing $thresh_c$ and $thresh_m$ will reduce the number of codewords on the line and thereby reduce interference. The overlap count threshold may be considered analogous to the cross correlationκ. Cross correlation limits the interference between a pair of codewords, while the overlap threshold limits the interference between codewords on the line and the codeword being sent.

Detecting the magnitude of an overlap needs threshold receivers that can sense power magnitudes of overlaps. In addition decision circuits will be needed to determine the exact magnitude. Given power attenuation of 0.2dB/km, for a LAN environment, an approximate determination of the magnitude should be feasible.

The threshold based media access algorithm is below:

```
Sense the line. Determine the cumulative line state with power levels.
Set departure instant t_d = 0
While (t_d < N)
    Check whether the state of the line would cause this codeword
        to be destroyed or the thresholds to be exceeded
        i.e. if the weight of (state|codeword) is less than thresh_c
        AND if the power level of any chip is less than thresh_m, then break
    Rotate the line state
    Increment t_d
If t_d < N, then defer till departure instant and then transmit data
If t_d > N, then continue sensing/back off according to the defer algorithm
```

C. Code estimation

The threshold mechanism seeks to limit, but not entirely eliminate interference. An alternative mechanism, is to determine the exact codewords on the line and try to avoid causing interference altogether.

Code estimation attempts to determine complete information about what codewords are on the line and what the relative phase shifts are. Given complete information, a node can determine a departure instant that does not cause interference either with its codeword or with any other node's codeword.

Using a sensing window mechanism a node can attempt to determine the probability with which codes are currently on the line. The effectiveness of this mechanism depends on the number of codes in the code set, the pattern of data bits on the line and the length of packets. The state of the line is a superposition of several codewords, possibly phase shifted. Decomposition of the state into possible codewords results in a large number of codewords. The codewords that have been seen the most and their phase shifts may be determined by a counting procedure i.e. a count is maintained of all possible codes and there phase shifts and a running estimate of the probability of each code is determined. As sensing progresses, some codewords will be seen more frequently and a more accurate estimate may be obtained.

In practice, this sensing algorithm is not very effective. Firstly, even for codesets with a small codelengths such as 10 chips, it gives a large number of false positives. In addition a large sensing window is needed to determine codes with some degree of accuracy. The large sensing window means that small packets may not be detected. These issues are further discussing in Section VII.

The code estimation algorithm is

Sense the line and determine the cumulative state of the line

  Estimate the possible codes using an estimation algorithm

  Increment the count for the estimated codewords

At the completion of a window determine the most likely codewords

Set departure instant $t_d = 0$

While ($t_d < N$)

    For each potential codeword, check for interference

      i.e. if (all other codewords & codeword $\neq$ codeword), then break

  Increment t_d

If $t_d < N$, then defer till departure instant and then transmit data

If $t_d > N$, then continue sensing or back off according to the defer algorithm

VI. Interference detection

Due to the finite propagation delay of the medium, interference sensing may not accurately estimate the interference on the line. E.g. A node may begin its transmission when another node is sensing. Due to the finite propagation delay, that codeword will not be sensed.

In order to avoid throughput degradation, a separate receiver is used to perform *Interference Detection* [7], a technique analogous to collision detection in CSMA/CD networks [8]. Upon detection of interference, the transmission is stopped and the packet is deferred.

As mentioned previously, the impact of defer algorithms for Interference Detection can be significantly different than that of a single channel CSMA/CD type protocol. In particular, because of the nature of interference (a single difference in chip time shift may result in no interference) the *detection algorithm* can operate at a different time granularity. We assume that in case of interference, the node continues to sense the line (in a 1-persistent fashion) and transmits on the next departure instant.

VII. Simulation results

In this section we define the metrics we use to evaluate the network performance. The metrics of interest include throughput and delay. To evaluate the system we use simulation and obtain throughput results, determine the effect of the sensing window and the effect of the threshold parameters.

A. Definitions

- Offered load: The offered load λ (in bits per second) is the rate at which data bits are offered to the network. This definition is independent of the size of the codeset used or number of codewords simultaneously on the line. The offered load may be increased by reducing the inter arrival time of packets.
- Throughput: The throughput, B (in bits per second) is the rate at which data bits are transmitted without error. Packets may be delayed because of interference. Packets can be lost due to interference on the line, or because the limit on number of transmission attempts is exceeded These factors are taken into account when calculating the throughput.
- Channel access delay: The delay is the time between the arrival of the packet at the network interface card and its transmission on the line. Packets which are lost due to interference are not included in the delay calculations.
- Maximum theoretical throughput: The maximum theoretical throughput of a codeset is the maximum throughput that it is capable of providing. It depends on the maximum number of simultaneous codewords that can be on the line at any given time which in turn depends on the N, w and κ.

B. Simulation parameters

A simulator was designed to simulate an optical CDMA network of chipping rate 100 Mbps. The codeset used was (N, w, κ) = (10,3,2) giving a data rate of 10Mbps. This allowed for interference sensing, given a network of maximum diameter 1000m (propagation delay 5μs). Though the simulation used a network of 100Mbps, the concept of interference sensing can be scaled to networks of higher speeds as long as the codelength is increased proportionately to allow sensing.

The traffic used to drive the simulation had an offered load varying from 10Mbps to 100Mbps. The packet lengths were chosen uniform randomly from between 40 bytes and 1500 bytes. The inter arrival times of packets were chosen uniform randomly to obtain the required offered load.

The codeword used to transmit a specific packet was chosen randomly from the codeset. The normal mode of operation unless otherwise specified was that the node operated in a on demand sensing mode for the selfish and the threshold based estimation and a *continuous sensing* mode was used for the code estimation algorithm. A 1-*persistent defer* algorithm was used - i.e. in case no departure instant could be determined, the transmission was retried immediately with probability 1. The sensing window, unless specified otherwise, was set to 10 data bits. The retry limit on the number of transmission attempts was set to 10. Interference detection was not implemented in these simulations and therefore no *detection* algorithm was employed.

C. Throughput and delay

Figure 2 shows the effect of the three sensing algorithms on the throughput of the network. As expected the throughput of Aloha-CDMA degrades rapidly as the offered load increases.

The selfish sensing algorithm maintains throughput close to the offered load up to around a normalized load of around 0.2. Thereafter the throughput is maintained constant at that load. The reason why the throughput is below the maximum theoretical throughput, is that the selfish algorithm is greedy - a node limits interference with respect to its own codeword and not with respect to other nodes' codewords. We found that the selfish algorithm gives optimal throughput under an on demand sensing mode.

The threshold based sensing algorithm provides a better control of the amount of interference on the line, providing better throughput increasing it by approximately 10%. However as will be discussed, this is highly sensitive to the choice of $thresh_o$ and $thresh_c$. The threshold based sensign gave best results using a on demand sensing mode.

The code estimation technique gives a throughput that is worse than both selfish and threshold based sensing. The reason for this is that the state of the line may be decomposed to a large number of potential codewords, thus giving a large number of false positives. The number of false positives reduce as the sensing window is increased. However packet arrivals and departures nullify these gains. In practice this method is not efficient and the efficiency will reduce further as the length of the code N increases. Code estimation needed a larger window and hence continuous sensing was the optimal method of sensing.

Other factors that impact the throughout for all algorithms include the limit on the number of retransmission attempts and the defer algorithm (1-persistent, p-persistent, non-persistent). Another factor that affects the throughput is the packet size. It was noted that the percentage of packets being successfully transmitted was far greater that the percentage of bytes successfully transmitted. The reason for this is that larger packets are more likely to suffer from interference than shorter packets that are on the line for a shorter duration. Thus larger packets are less likely to get through (fate sharing), but each lost packet costs the same as a small packet to the packet total, but it costs more than a small packet to the byte total. E.g. Assume that out of a total of 20000 packets transmitted, half are large (1000 bytes) and half are small (100 bytes). Assume that 90% of large packets are lost and 10% of small packets are lost. Packet throughput is around 50% while byte throughput is around 20%.

The study of how these affect throughput is an area currently under research.

Figure 3 shows the channel access delay for the sensing algorithms. It is measured only for successful transmissions. The delay for all algorithms is kept within reasonable bounds i.e. in the order of μs.

D. Effect of sensing window

If the codewords remain static on the line during a sensing window, the probability of any node sending at least 1 data bit within 5 data bit times is $(1 - 0.5)^5 = 97\%$. In practice, as can be seen from Figure 4 a slightly larger window of around 10 bits is required because of packet arrivals and departures. The optimal window size for the code estimation method is lager than that of the other two in order to eliminate the false positives that the algorithm identifies.

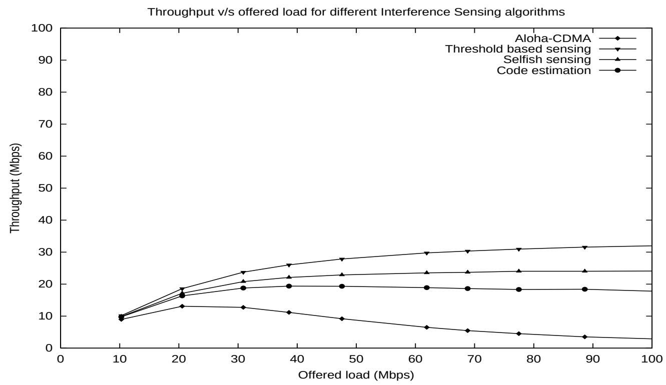

Fig. 2. Throughput v/s offered load for different sensing algorithms for a (10,3,2) code on a 100 Mbps network. A continuous sensing algorithm was used with sensing window = 10 bits for the selfish and threshold based algorithms and 100 bits for code estimation. A 1-persistent defer algorithm was used with a retry limit of 10.

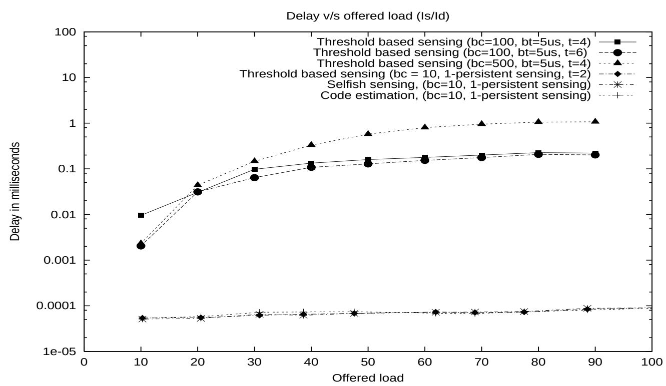

Fig. 3. Channel access delay v/s offered load for different sensing algorithms for a (10,3,2) code on a 100 Mbps network. The results are shown for different values of the threshold settings (overlap count threshold (t)), different retry limits (bc). Two types of defer algorithms were used: a 1-persistent defer algorithm and a backoff based algorithm with backoff time (bt) = 5μs. Continuous sensing was used with a sensing window of 10 bits for the selfish and threshold based algorithms and 100 bits for code estimation.

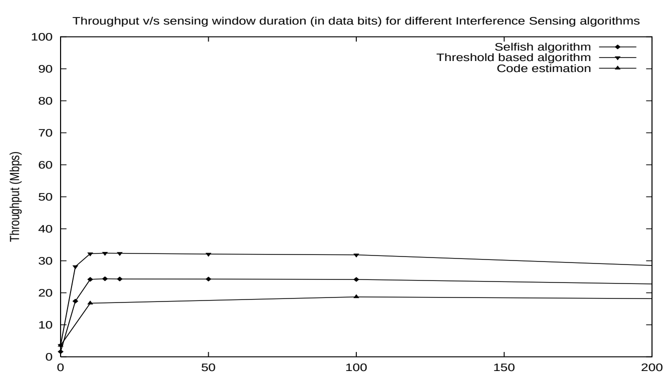

Fig. 4. Throughput v/s sensing window for a (10,3,2) code on a 100 Mbps network at 100% offered load. A on demand sensing mode was used for selfish and threshold sensing and continuous sensing was used for code estimation to provide maximum throughput. A 1-persistent defer algorithm was used with a limit of 10 retries.

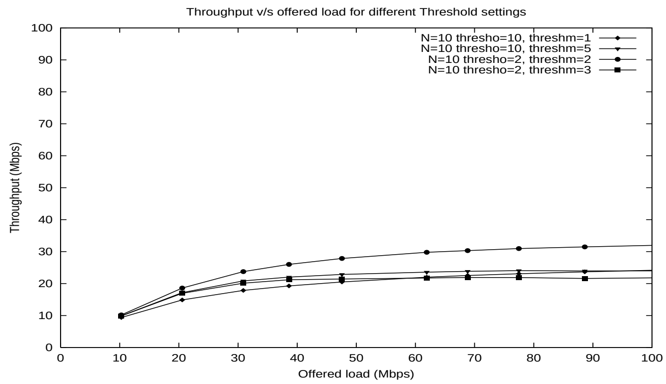

Fig. 5. Throughput v/s offered load for different values of threshold sensing parameters of (10,3,2) code on a 100Mbps network at 100% offered load. On demand sensing and a 1-persistent defer mode was used.

E. Effect of threshold settings

Earlier work [7] describes a probabilistic analysis of the threshold based sensing algorithm method for a setting of $thresh_o$ from 1 to N and $thresh_m$= 2. Our simulation studies indicate that the choice of threshold parameters must be made carefully. When $thresh_m$ is at its minimum value i.e. 1, there are no overlaps and the choice of $thresh_o$ does not affect the throughput. At this setting, the number of simultaneous codewords on the line is approximately N/w. To increase the number of codewords words on the line, $thresh_m$ must be increased. However to ensure that the interference is limited, $thresh_o$ must be reduced. For the codeset used in the simulations, we found that a setting of $thresh_o = 2$, $thresh_m = 2$, gave maximum throughput, around 5% more than the other threshold values and 10% more than selfish sensing. An incorrect selection of the threshold values renders the scheme no more effective than selfish sensing.

Figure 5 shows the throughput values for different values of threshold settings.

VIII. Conclusion

Optical CDMA networks have been studied for several years. However concerns about their throughput have led to skepticism about their utility.

Interference sensing is a method of improving throughput under heavy load. We analyzed three algorithms for Interference Sensing and have shown it is possible to operate an optical CDMA LAN at close to its maximum possible throughput at high loads. A judicious choice of the interference sensing parameters can ensure that the delay is kept within reasonable bounds. In particular, simple algorithm such as the greedy selfish algorithm and threshold based estimation can reduce interference levels significantly, increasing throughput by up to 10% over more complex code estimation algorithms.

REFERENCES

[1] P.R. Prucnal, M.A. Santoro, and T.R. Fan, "Asynchronous multiplexing for an optical fiber local area network," *IEEE Journal of Lightwave Technology*, vol. 4, pp. 547, 1986.

[2] J. Salehi, "Code division multiple-access techniques in optical fiber networks - Part 1: Fundamental principles," *IEEE Transactions on Communications*, vol. 37, no. 8, pp. 824–833, Aug. 1989.

[3] J. Salehi and C. Brakett, "Code division multiple-access techniques in optical fiber networks - Part 2: Systems performance analysis," *IEEE Transactions on Communications*, vol. 37, no. 8, pp. 834–842, Aug. 1989.

[4] C.F. Lam, "To spread or not to spread: The myths of optical CDMA," in *IEEE Lasers and Electro-Optics Society Annual Meeting*, 2000, vol. 2, pp. 810–811.

[5] H. Chung and P.V. Kumar, "Optical orthogonal codes - new bounds and an optimal construction," *IEEE Transactions on Information theory*, vol. 36, no. 4, pp. 866–873, July 1990.

[6] S. Johnson, "A new upper bound for error correcting codes," *IRE Transactions on Information Theory*, pp. 203–207, April 1962.

[7] P. Kamath, J. Touch, and J. Bannister, "The need for media access control in Optical CDMA networks," *ISI Technical Report ISI-TR-2003-575 (also under submission)*, 2003.

[8] F.A. Tobagi and V.B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," *Computer Networks*, vol. 4, no. 5, pp. 245–259, Oct 1980.