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 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. This paper discusses and analyzes three algorithms for interference sensing. Through simulation it is shown that these algorithms reduce or eliminate throughput degradation at high loads. The study shows that simple algorithms such as selfish or threshold based algorithms are sufficient to eliminate throughput degradation.**

KEYWORDS: System design

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. A pulse of light is used to transmit a signal.

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 codeset designs limit the autocorrelation and cross-correlation of the codeset to a fixed value called the *Maximum Collision Parameter* κ.

Most optical CDMA networks are ON-OFF keyed optical CDMA networks i.e. the presence of a codeword signifies the transmission of a 1 data bit and absence signifies a 0. The network is a single wavelength broadcast network. Multiple nodes can transmit simultaneously. The optical medium is additive i.e. because signals are sent as power, signals sent simultaneously are added. The Optical CDMA receivers are correlation based receivers.

Optical CDMA allows nodes to transmit asynchronously without any media access delay. One of it's major disadvantages is its low throughput under heavy loads. At high offered loads, the cause for low throughput is multiuser interference i.e. the interference between codewords.

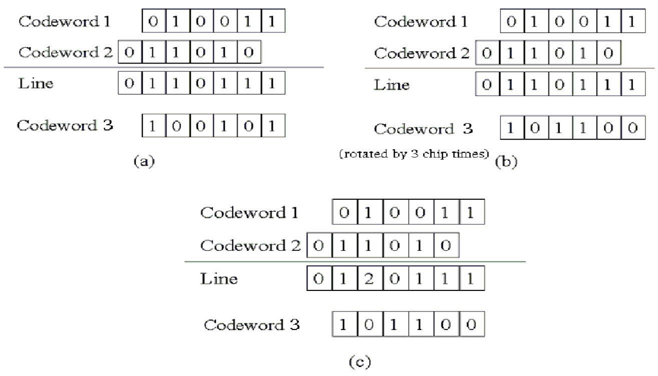

Fig. 1. Examples of interference between codewords

II. MOTIVATION

The interference between codewords on an Optical CDMA network 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). Each node on the network is transmitting a 1 data bit. 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. 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 rotated by 3 chip times -it shows the end of one data bit and the start of the next).

The throughput of optical CDMA under heavy loads can be improved by media access mechanisms that prevent inter-

[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 authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, SPAWAR, or the U.S. Government.

fering codewords from being sent simultaneously by deferring transmissions upon detecting interference. This mechanism is called *Interference Sensing / Interference Detection (IS/ID)* [5]. Without such a mechanism throughput degrades rapidly at high loads, dropping close to zero under 100% offered load [5]. This paper describes and analyzes three different interference sensing based media access algorithms. The three algorithms are compared using a simulation based approach.

III. System Architecture

The network is a single wavelength broadcast star coupler based system. Every node on the network 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, the study assumes a tunable transmitter-fixed receiver system, where nodes choose which codeword to receive on when they start up. A mechanism exists to map the node address to the codeword it will receive on. This mechanism may be as simple as a hash function.

Because the network is a broadcast network, every node sees every transmission. Because a star coupler is used, every node sees the same additive data on the line. Every node has a ranging mechanism that allows it to determine its round trip time to the coupler to the accuracy of a chip time.

In this analysis losses due to nodes being unable to synchronize to the preamble are neglected. Losses due to receiver contention (two packets simultaneously sent to the same receiver on the same codeword) are neglected. For the purposes of simulation, chip synchrony is assumed. A more detailed description of the system architecture may be found in [5].

IV. The IS/ID Media Access Protocol

There are several parameters and mechanisms that impact the performance of the IS/ID media access protocol. This section discusses the following parameters and algorithms in detail:

- The codeset parameters
- The Interference Sensing algorithm
- The Interference Detection algorithm
- The sensing mode
- The defer mode

A. Codeset parameters

The effectiveness of the interference sensing algorithm depends on the codeset parameters N, w and κ. This paper does not discuss these parameters. The effect of these parameters on the aggregate network throughput is discussed in [5].

B. The Interference Sensing algorithm

Interference Sensing is a media access control mechanism by which nodes sense the state of the line before transmission. If the state of the line interferes then transmission is deferred.

The *state of the line* at any time is the sum of codewords of the packets being transmitted on the network at that time. It is the sum of several codewords, each codeword possibly shifted by different amounts. The state of the line depends on the exact instant of time (i.e. the chip time) when it is measured. Noe that if it is measured one *data bit* time slot later the state will be the same unless new codewords arrive or old codewords leave.

The Interference Sensing algorithm has two purposes:

- To estimate the state of the line
- To determine if it can transmit and if so, when to transmit

The *line state estimation algorithm* that was used in this study is as follows: A node senses the line for a duration of time equal to a single data bit i.e. N chips. This represents a single state reading. Several such state readings are taken for a window of time (say W data bits) called the *sensing window*. Assume that each transmission consists of an equal number of 0 and 1 data bits. Then the state of the line is twice the sum of the states averaged over the window.

Once the node has estimated the state of the line, it must determine if it can transmit its codeword and the exact instant it may transmit. The time measured from the start of a state bit to the time at which transmission is possible (in chip times) is called a *departure instant*. For a given line state and codeword to be transmitted, there may exist up to N potential departure instants. This paper discusses three algorithms which may be used to determine departure instants: a selfish algorithm, a threshold based algorithm, and codeword estimation. The algorithms are discussed in Section V.

C. The Interference Detection algorithm

Due to the finite propagation delay of the medium, interference sensing may not be able to accurately estimate the state of the line. E.g. Two nodes may begin sensing at the same time. They may both sense a non interfering line state and transmit, but their transmissions may interfere with each other. Interference Detection is a mechanism by which nodes continue to sense the line during transmission. If interference is detected the transmission is aborted and deferred. A dedicated receiver is used for interference detection. The receiver is tuned to the codeword of the packet being sent and verifies that the codeword being transmitted is received correctly. Note that the detection algorithm is a selfish algorithm because the node aborts transmission only if its own packet is received incorrectly.

D. Sensing mode

The Interference Sensing mode describes how a node performs its sensing operation. There are two modes:

- *On-demand sensing*: Sensing is started when a packet arrives for transmission. The node senses for a window of time. After the window expires, the state of the line is estimated and the interference sensing algorithm decides whether to transmit or not.
- *Continuous sensing*: Sensing is performed continuously irrespective of whether the node has a packet to transmit.

After a window of time expires or on packet arrival, the state of the line is estimated. The estimated state of the line is used by the interference sensing algorithm to decide whether to transmit or not.

E. Defer mode

If the sensing algorithm is unable to determine a departure instant or if the detection operation detects interference, the node defers its transmission. The way the node defers is referred to as the *defer mode*. The purpose of the defer mechanism is to reduce the probability of multiple nodes which interfere with each other from accessing the medium at the same time. Conventional CSMA/CD protocols use several schemes to decide how to defer. These schemes can be used as defer modes in the interference sensing protocol. There are three defer modes which are explained in detail in [6]:

- *Non-persistent mode*
- p*-persistent mode*
- 1*-persistent mode*

In this study two modes were examined: non-persistent mode with a fixed *back off interval* i.e. if interference was sensed, transmission was deferred for a fixed interval of time after which sensing was retried and 1-persistent. Both schemes used a fixed *retry limit* i.e. a limit on the number of retransmission attempts.

V. Algorithms for Interference Sensing

This section describes the three algorithms for Interference Sensing: *Selfish sensing*, *Threshold based sensing* and *Codeword estimation based sensing*

A. Selfish sensing

In selfish sensing, a node transmits if the state of the line will not cause interference with the codeword that it is transmitting. Otherwise 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.

The state estimation algorithm is described in Section IV. The time required for estimating the state of the line depends on the sensing window. The selfish sensing algorithm is described above. The running time of this algorithm is $O(N)$.

```
C_t ← Codeword to be transmitted
Retry count rc ← 0
Sensing algorithm:
  Sense the line according to the sensing mode
  Estimate the state of the line
  Departure instant t_d ← 0
  While (t_d <= N)
    If (state & C_t ≠ C_t) then break
    Rotate C_t by one chip
    t_d ++
  If t_d <= N
    Defer till departure instant t_d
    Transmit data
  If t_d > N
    rc ++
    If (rc < retrylimit)
      Defer according to the defer mode
      Repeat sensing algorithm
    Else send packet back to higher layer
```

The selfish interference sensing algorithm

B. Threshold based sensing

Threshold based sensing limits the amount of interference caused to other users.

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 in a single chip time)

Based on these limits, two thresholds may be defined: 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.

```
C_t ← Codeword to be transmitted
Retry count rc ← 0
Sensing algorithm:
  Sense the line according to the sensing mode
  Estimate the state of the line
  Departure instant t_d ← 0
  While (t_d <= N)
    If ((state & C_t ≠ C_t) AND
      (weight(state|C_t) ≤ thresh_c) AND
      (maximum magnitude of all overlaps≤ thresh_m)) then break
    Rotate C_t by one chip
    t_d ++
  If t_d <= N
    Defer till departure instant t_d
    Transmit data
  If t_d > N
    rc ++
    If (rc < retrylimit)
      Defer according to the defer mode
      Repeat sensing algorithm
    Else send packet back to higher layer
```

The threshold based interference sensing algorithm

A node first performs the selfish test to determine whether the line state allows it to transmit without interfering with its own codeword. It then checks the overlap count and overlap magnitudes between its codeword and the state of the line. If the counts are below the thresholds, the node transmits its codeword. If not the node defers transmission. Decreasing $thresh_c$ and $thresh_m$ reduces the number of codewords on the line and thereby reduces interference.

The state estimation algorithm is discussed in Section IV. The time required for estimating the state of the line depends on the sensing window. The threshold based sensing algorithm is described above. The running time of this algorithm is $O(N)$.

C. Codeword estimation

Codeword estimation attempts to determine which codewords are on the line and what their relative phase shifts are. Given this information, a node can determine a departure instant that does not cause interference either with its code-

```
Ct ← Codeword to be transmitted
Retry count rc ← 0
Sensing algorithm:
  Sense the line according to the sensing mode
  Estimate most likely codewords according to estimation algorithm
  Departure instant td ← 0
  While (td <= N)
    For each most likely codeword C
        state ← All other most likely codewords + Ct
        If (state & C == C)
            Rotate Ct by one chip
            td + +
            Continue the while loop
    Break from the while loop
  If td <= N
    Defer till departure instant td
    Transmit data
  If td > N
    rc + +
    If (rc < retrylimit)
      Defer according to the defer mode
      Repeat sensing algorithm
    Else send packet back to higher layer
```

The codeword estimation interference sensing algorithm

word or with any other node's codeword. The efficiency of this sensing algorithm depends on how accurate the codeword estimate is.

This study used a *window based estimation* technique: A counter was maintained for every codeword in the codeset. The state of the line was sensed and the counter was incremented for every codeword that could be a part of the state. At the end of the window the counters were divided by the duration of the window. This gave an estimate of the probability of that codeword having been present. The codewords which had probabilities above a threshold were chosen as the *most likely codewords*. As sensing progresses the estimated codewords tended towards the actual codewords. Other estimation algorithms based on vector decomposition or maximum likelihood are also possible.

The window based codeword estimation algorithm runs in $O(NP)$ where P is the number of codewords in the codeset. (From the Johnson bound $P = O(N^{\kappa})$.) The total time required for estimating the state of the line also depends on the sensing window. The sensing algorithm to determine the departure instant is described above. The running time of the algorithm is $O(NP)$.

VI. Simulation results

This section compares the three sensing algorithms and the effect of the algorithmic parameters on throughput and delay characteristics.

A. Simulation parameters

A simulator was designed to simulate an IS/ID based optical CDMA network. The chipping rate of the network was $100Mb/s$. The codeset used was (N, w, κ) = (10,3,2) giving a data rate of $10Mb/s$. This allowed for interference sensing, given a network of maximum diameter 1000m (propagation delay 5μs). Though the simulation used a chipping rate of $100Mb/s$, the concept of interference sensing can be scaled to networks of higher speeds as long as the codelength is increased proportionately to allow interference sensing to be feasible [5]. The traffic used to drive the simulation had an offered load that varied from $10Mb/s$ to $100Mb/s$. 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 from 10 to 100% of the chipping rate. The codeword used to transmit a specific packet was chosen randomly from the codeset. On-demand sensing was used for the selfish and the threshold based estimation and continuous sensing was used for the codeword estimation algorithm. Both 1-persistent and p-persistent defer algorithms were used. The retry limit was set to 10. The sensing window, unless specified otherwise, was set to 10 data bits.

B. Throughput and delay

Figure 2 shows the effect of the three sensing algorithms on the throughput of the network. The throughput for an optical CDMA system which does not use any form of media access (called Aloha-CDMA) is also shown.

The selfish sensing algorithm maintains throughput close to the offered load up to a normalized load of around 20%. Thereafter the throughput is maintained constant. The reason why the throughput is below the maximum possible throughput is that the selfish algorithm is greedy - a node checks interference with respect to its own codeword and not with respect to other nodes' codewords. The selfish algorithm gives its maximum throughput when used in the on-demand sensing mode.

The threshold based sensing algorithm controls of the amount of interference on the line providing better throughput. The throughput increases by approximately 10%. Threshold based sensing gave best results using a on-demand sensing mode.

The codeword estimation technique gives a throughput that is worse than both selfish and threshold based sensing. This is because the state of the line when decomposed gives a large number of potential codewords, in turn giving a large number of false positives. The number of false positives can be reduced by increasing the sensing window. However packet arrivals and departures nullify these gains. Efficiency will reduce further as N increases. Codeword estimation needed a large window and hence continuous sensing was used.

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.

C. Effect of sensing window

Figure 4 shows the effect of the sensing window on throughput. A window of around 10 bits is required for accurate sensing using selfish and threshold based sensing. The optimal window size for the codeword estimation method is larger (around 100 bits) to eliminate the false positives that the algorithm identifies.

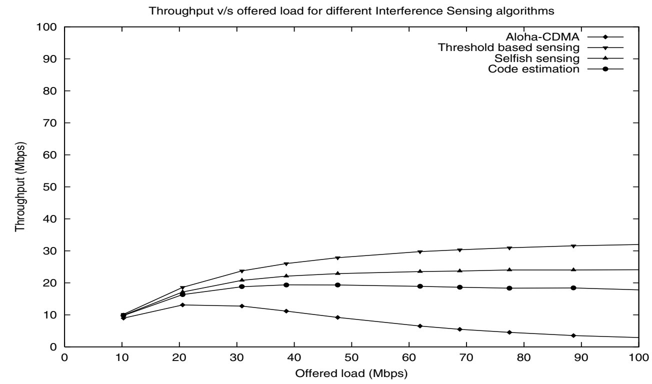

Fig. 2. Throughput vs. offered load for different sensing algorithms for a (10,3,2) code on a $100Mb/s$ network. A continuous sensing algorithm was used with a sensing window of 10 bits for the selfish and threshold based algorithms and 100 bits for codeword estimation. A 1-persistent defer algorithm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

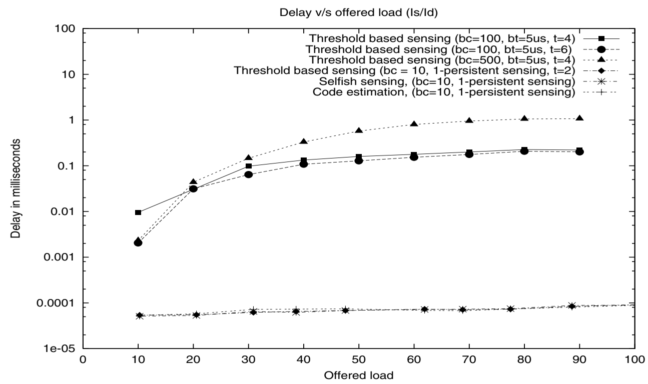

Fig. 3. Channel access delay vs. offered load for different sensing algorithms for a (10,3,2) code on a $100Mb/s$ network. The results are shown for different values of the threshold settings (overlap count threshold (t)), different retry limits (bc). Two types of defer algorithms were used: a 1-persistent 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 codeword estimation.

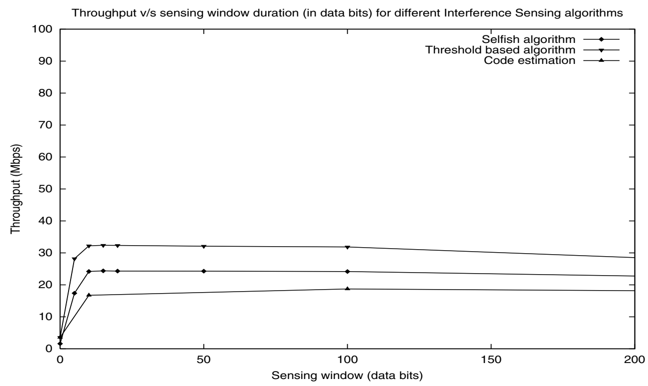

Fig. 4. Throughput vs. sensing window for a (10,3,2) code on a $100Mb/s$ network at 100% offered load. A on-demand sensing mode was used for selfish and threshold sensing and continuous sensing was used for codeword estimation to provide maximum throughput. A 1-persistent defer algorithm was used with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

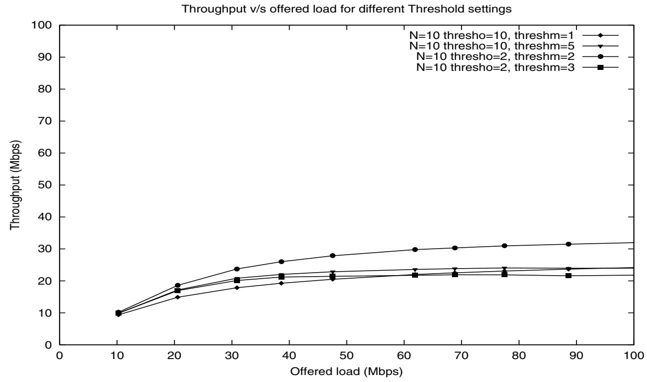

Fig. 5. Throughput vs. offered load for different values of threshold sensing parameters of (10,3,2) code on a $100Mb/s$ network. On demand sensing and a 1-persistent defer mode were used.

D. *Effect of threshold settings*

Figure 5 shows the throughput values for different values of threshold settings. When $thresh_m$ is at its minimum value i.e. 1, no overlaps are allowed and the choice of $thresh_c$ does not affect the throughput and may be set to its maximum value of 10. To increase the number of codewords on the line, $thresh_m$ must be increased. However to ensure that the interference is limited, $thresh_c$ must be reduced. For the codeset used in the simulations, 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.

VII. Conclusion

Interference sensing is a media access mechanism that can improve the throughput of an Optical CDMA network under heavy load. Three algorithms for Interference Sensing were analyzed and it is shown that 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 algorithms such as the selfish and threshold based algorithms can reduce interference levels significantly, increasing throughput by up to 10% over more complex codeword estimation algorithms.

References

[1] P. Prucnal, M. Santoro, and T. Fan, "Asynchronous multiplexing for an optical fiber local area network," *IEEE Journal of Lightwave Technology*, vol. 4, p. 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. Lam, "To spread or not to spread: The myths of optical CDMA," in *IEEE Lasers and Electro-Optics Society Annual Meeting*, vol. 2, 2000, pp. 810–811.

[5] P. Kamath, J. Touch, and J. Bannister, "The need for media access control in Optical CDMA networks," in *IEEE Infocom*, 2004.

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