Efficient Address Decoding for
Fast Packet Switching Systems

by

Y. Ben-Asher, A. Cohen, A. Schuster

Technical Report #752
September 1992
Efficient Address Decoding for Fast Packet Switching Systems

Yosi Ben-Asher
Mathematics Dept.
Haifa University, Haifa
yosi@mathcs.haifa.ac.il

Aviad Cohen
Comp. Sci. Dept.
Hebrew Univ., Jerusalem
aviadc@iil.intel.com

Assaf Schuster
Comp. Sci. Dept.
Technion IIT, Haifa
assaf@cs.technion.ac.il

Abstract

We consider the theory and structure of a free space optical message switching systems with \( N \) sources and \( N \) destinations. Each source uses only \( O(\log N) \) address bits of light, compared to \( N \) address bits used in conventional optical crossbars [GJ81a, GJ81b, Ne85, Au85, DKGS88]. It is shown that a non-linear part must be included in the scheme and how to achieve high utilization of this part.

The key problem is how to direct a light beam to \( N \) different destinations, having no more than \( O(\log N) \) address bits of light. By using simple dark/light transparencies and some threshold devices we can construct such an optical system.

The novelty of this paper is the use of a randomly constructed set systems with bounded intersection as address spaces. In addition, such a system is capable of broadcasting, so that communication is not restricted to the “point to point” mode.
1 Introduction

Optical realization of digital computers, has been attracting attention recently. In particular, optical techniques seem suitable for the realization of fast parallel computers [Fe88, Lo86]. This is so, since the most crucial issue in parallel computation, is the bottlenecks that arise in the communication mechanism between the processors, and free space optic avoids this problem. These bottlenecks are due to the time needed to relay a message in the network, collisions of messages, and the complicated topologies which imply long wires. The main advantage of free space optics, is that light beams are used to communicate messages to their destinations, and these can cross each other without any interaction. In this way the routing of one message becomes oblivious to the routing of any other message.

The general problem that is considered is the parallel communication of \( N \) processors. We view it as \( N \) sources of messages and \( N \) destinations for these messages. In order to realize such interconnections optically, every source needs a mechanism that is capable of directing a light beam to all the desired destinations. This mechanism may be viewed as an optical analogue of a multiplexor, i.e. a box having an address of at least \( \log N \) bits as its input. The output of the multiplexor is a light beam directed towards one of \( N \) distinct destinations. Such an optical device can also serve as a basic component also in the realization of optical memory, truth-tables and ALU operations [Fe88].

The collection of all \( N \) addresses is called the address space used by the system. The number of bits constituting an address is called the address size. We shall later formulate the notion of an address space as a set system [BF88].

Previous suggestions for the realization of optical multiplexors, have used either Acousto-optic devices or vector - matrix product systems. Acousto-optic oriented systems experience some severe restrictions [FRS90, Fe88]. In this paper we focus on systems using operations similar to vector - matrix products. Vector - matrix product systems are well known, see [GJ81a, GJ81b, Ne85, Au85, DKGS88]. The use of such systems in realizations of communication typically employ spatial light modulators (SLMs) to construct optical crossbars, i.e. a collection of \( N \) multiplexors which share the same set of \( N \) destinations (see also [MJR89, Re90] and [Fe88] Sec. 3.6.4 and 7.6.2). The crossbar multiplexors use \( N \) address bits each. The routing is performed as follows: Every bit is associated with one destination, and it is controlled by a different SLM. By setting a single bit to 1, the processor blocks all other destinations and hence only one light beam is transmitted.

The above construction uses an amount of address bits which greatly exceeds the information theoretic lower bound (i.e. \( \log N \)). However the complexity of the electronic circuitry that controls the SLMs, and amount of light energy needed to illuminate them, increase with the number of address bits. The method, thus, tends to “waste” energy and time [YCH88], so it becomes important to minimize the number of address bits (address size).

The problem of reducing the number of address bits (used to control the optical multiplexor), lies in the difficulty of achieving a good “separation” of two address words that are encoded by a short description. Clearly, if the system can not distinguish between two addresses, a message to one of the destinations associated with these addresses may mistakenly reach the other destination as well. This problem was posed by Maniloff et. al. in [MJR89]. They introduced a quantity that measures this difficulty of separation, namely the crosstalk. The crosstalk is defined as the maximal number of bits set to '1' which two different addresses have in common (or simply: the intersection size). The crosstalk varies between two extreme values: at one extreme, the crosstalk may be vanish to zero. This situation, easily achieved when the address size is \( N \), appears when every two addresses, viewed as binary vectors, are orthogonal. On the other hand, we shall see
in section 4 that when the address size is asymptotically optimal, $\Theta(\log N)$, then the crosstalk is $\Omega(\log N)$. Maniloff et. al. presented an architecture that uses $\sqrt{N}$ address bits with crosstalk of at most one [MJR89].

How the crosstalk is related to the complexity and the performance of a free space implementation of an optical multiplexor? Free-space optical systems may contain linear and non-linear components. A linear component outputs a linear combination of its inputs, while a non-linear component outputs a non-linear function of its inputs. Linear components (such as lenses or mirrors) are ideal for free-space optical system, since they are passive elements (i.e., they do not use external source of energy), and hence do not slow down the light beams passing through them. On the other hand non-linear components, such as SLMs, or bistable devices require an external source of energy and adjustment time. Since non-linear components tend to complicate and slow-down free-space optical systems, we wish to reduce the use of non-linear in the realization of the multiplexor. When the crosstalk is high the optical system has to perform more non-linear operations (such as AND gates) to separate two different address words. The problem then, is to design a free-space optical multiplexor with high crosstalk and minimal use of non-linear components. However an optical multiplexor has an inherent non-linearity, as we shall see in section 1.1.

Consider the non-linear components of a message switching system. Typically, these would consist of some threshold element with fixed sensitivity. In other words, the problem is to design a message switching system with any number $N$ of processors, and a given, fixed parameter of sensitivity $D_{th}$ of the non-linear components. For example, consider some threshold device which can distinguish between a level $X$ of beam intensity and level $X + D_{th}$, say it outputs a 0 when the beam intensity is $\leq X$ and a 1 when the beam intensity is $\geq X + D_{th}$. Once the system grows ($N$ increases), it is customary to assume that $X$ increases too (that is, since $X$ usually stands for the “noise” in the system, which is actually the crosstalk in our setting). However once the input needs scaling down, the threshold gap $D_{th}$ is scaled down, too. This would typically require this gap to grow with the crosstalk.

Reif [Re90] considered address spaces with address size $2\log N$ and crosstalk $\log N - 2$. In Reif’s construction, the crosstalk may get up to $\log N - 2$, however $D_{th} = 2$ stays constant. Here we suggest a method to construct an optical multiplexor which requires $O(\log N)$ address bits. The constructions we consider do not assume a constant $D_{th}$ on the one hand, and have crosstalk which does not exceed a constant fraction of the address size. That is, the sensitivity parameter $D_{th}$ increases together with the crosstalk to allow for the separation of any two distinct address words, regardless of the size of the system, $N$.

The rest of the paper is organized as follows: In section 1.1 a formal definition is given for linear systems. It is shown that multiplexors are not a linear system. Section 2 describes the model of message switching systems. In section 3 gives a probabilistic construction of multiplexors, that is, address spaces of size $O(\log N)$. Section 4 shows that the crosstalk of address spaces having address size of $O(\log N)$ is $\Omega(\log N)$. Section 5 gives an explicit construction of addresses of size $O(\log^2 N)$ bits. Section 6 describes a schematic structure of an optical multiplexor. A general proposal of how to pack $N^2$ optical multiplexors to form a message switching system is included as well.

### 1.1 Non-Linearity in Multiplexors

**Definition 1.1** A linear system (device) is a system (device) where the intensity of light at every output beam is a linear combination of the intensities of the input beams.

Clearly, any combination of linear systems (devices) is a linear system.
We shall think an optical multiplexor as having two possible light intensities for an input beam, \( \alpha \) and \( \beta \), representing the '1' and '0' binary values, correspondingly (obviously, \( \alpha, \beta > 0 \)). The first theorem concerns the impossibility of realizing an optical multiplexer by a linear system, with a "small" number of address bits:

**Theorem 1.1** A multiplexor with \( m < N \) address bits and \( N \) destinations cannot be realized by a linear system.

**Proof:** Let \( m < N \). We view the multiplexor as a system having input vectors (addresses) of length \( m \) and output vectors of length \( N \). The \( N \) possible addresses have entries from \{\( \alpha, \beta \)\}. Each address produces an \( \alpha \) value in a single direction and \( \beta \) values in all other \( N - 1 \) directions. Thus, the multiplexor having the \( j \)'th address as input, produces the output vector having \( \alpha \) at the \( j \)'th entry and \( \beta \) elsewhere.

Assume by contradiction, that the multiplexer may be realized by a linear system. Then, by definition, the multiplexor outputs consists of \( N \) linear combinations of the address entries. Let us denote the \( k \)’th output entry \( O(k) \), and let \( X_i \) denote the \( i \)'th entry of the vector \( X \), then

\[
O(k)(X) = \sum_{i=1}^{m} \lambda_i^{(k)} X_i
\]

where the \( \lambda \)’s coefficients are real constants. A linear multiplexor is thus a linear operator, so that its operation is a multiplication of the address by an \( m \times N \) matrix of coefficients, having \( \lambda_i^{(k)} \) at its \([i,k]^{th}\) entry.

Let \( A^{(1)}, A^{(2)}, \ldots, A^{(N)} \) be the \( N \) different addresses. The functions \( O(k) \) satisfy

\[
O(k)(A^{(i)}) = \begin{cases} 
\alpha & i = k \\
\beta & i \neq k 
\end{cases}
\]

Thus the input of all the addresses to the multiplexor is characterized by a matrix multiplication, producing the following system of equations:

\[
\begin{pmatrix}
A^{(1)}_1 & A^{(2)}_1 & \cdots & A^{(m)}_1 \\
A^{(1)}_2 & A^{(2)}_2 & \cdots & A^{(m)}_2 \\
\vdots & \vdots & \ddots & \vdots \\
A^{(1)}_{N} & A^{(2)}_{N} & \cdots & A^{(m)}_{N}
\end{pmatrix}
\begin{pmatrix}
\lambda_1^{(1)} & \lambda_2^{(1)} & \cdots & \lambda_m^{(1)} \\
\lambda_1^{(2)} & \lambda_2^{(2)} & \cdots & \lambda_m^{(2)} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_1^{(N)} & \lambda_2^{(N)} & \cdots & \lambda_m^{(N)}
\end{pmatrix}
= \begin{pmatrix}
\alpha & \beta & \beta & \beta & \beta & \beta \\
\beta & \alpha & \beta & \beta & \beta & \beta \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\beta & \beta & \beta & \beta & \beta & \alpha
\end{pmatrix}
\]

However, since \( \alpha \neq \beta \) and \( \alpha, \beta > 0 \), the \( N \times N \) matrix to the right of the equality sign has rank \( N \) [BF88]. By elementary linear algebra, the two matrices to the left of the equality sign have rank \( N \), too. Thus \( m = N \), contradicting the assumptions of the theorem.

Since linear systems (such as convolution, or Fourier transform) do not suffice to realize the optical multiplexor, a possible solution is to use threshold bistable devices ([Fe88] Sec. 6.2,7.2.2) which perform non-linear operations.
2 The model

We now describe the model for message switching of $N$ messages, initially residing at $N$ processors, to $N$ destinations. Such a communication model is sometimes referred to as an optical crossbar. Conceptually, an optical crossbar is defined as an optical matrix-vector multiplication $v' = M \times v$ [YCH88]. For a vector $v = v_1 \ldots v_n$ where $v_i$ is the message sent by processor $i$ and $M$ is the communication matrix $M_{i,j} = 1$ if processor $i$ wants to send a message to processor $j$. This is valid for the case of “full address" $m = N$, for $m \leq N$ we need a different formal model:

**Processors**- A collection of $N$ processors. Each processor uses an “optical multiplexor" as a mechanism to communicate with $N$ destinations.

**Optical Multiplexor**- An optical multiplexor transfers light from an input-entry to one of the $N$ possible output destinations. The input entry contains two value bits, one for sending '0' and one for sending '1'.

**Input vector**- The optical multiplexor has an address input vector, that holds the output address. This input vector has $O(\log N)$ address bits, where each bit can be set by the processor.

**Destination**- A destination has two detectors for the two '0' and '1' bits, if both bits are set then it is a collision. In the case of a collision, the message is ignored by the destination.

**Organization of the various components**- We require that the $N$ optical multiplexors and $N$ destinations be arranged in the 3-D space in such a way that every processor can communicate with the other processors in 3 different modes:

- **Point to point messages**- Processor $P_i$ can send a message to some destination by setting the input vector of its multiplexor to the address of the desired destination.
- **Broadcast to all**- A processor can broadcast a message to all the destinations simultaneously. We will designate a special address word to indicate the broadcast to all mode.
- **Broadcast to some**- A processor can send one message to a subset of up to $t$ processors ($t > 1$ some fixed parameter). Again we will designate a special address word to indicate this mode. We will also allocate one address word to indicate the $t$ destinations. To do this, we will need an unambiguous way of representing $t$ destinations in one address. In fact we will construct an encoding scheme in which the union of the addresses determines uniquely its constituents.

Note that the broadcast to all and broadcast to some modes should be maintained without any additional hardware.

3 $O(\log N)$-bit Address Spaces Exist

We are going to build an address space for a machine with $N$ origins and destinations for routing permutations of messages. Each destination will will be represented by an address vector of $C \log N$ bits. Such a vector may also be thought of as the characteristic function of a subset of a domain of $C \log N$ elements and therefore we shall sometimes use the term “set system" as a synonym for the term “address space". We shall seek an address space with the property that each address vector has at least $C \log N \cdot a \, 1s$ (for some fraction $a$). The threshold of the address space $D_{th}$ is defined as the difference between the minimal size of an address vector (viewed as a subset), and the
maximal size of the intersection of any pair of distinct address vectors. It will be convenient to set
$D_{th} = C \log N \cdot a \cdot d$ and use the parameter $d$ as a measure of the threshold. In our implementation
$D_{th}$ corresponds to the sensitivity of the optical threshold detectors. The question to be dealt with
here is whether there exists an address space of $O(\log N)$ bits, so that the number of 1s which is
common to any two different addresses is relatively small, compared to the number of 1s at any
one of the addresses?

**Theorem 3.1** Let $N$ be a natural number and $0 < d < 1$. For any

$$C \geq \frac{54(1 - \frac{d}{2})^2}{\log e (1 - d)^3}$$

there exists an address space of size $C \log N$ with the following property: Each address, viewed as a
binary vector, has at least $Ca \log N$ set bits (for some constant $a$) and intersection of size at most
$Ca(1 - d) \log N$, with any other address.

Proof Let $C$ and $0 < \beta < \alpha$ be constants whose values will be specified later and set $n = C \log N$. The proof is divided into two steps:

**step 1**

The first step is to give a probabilistic construction of a set system $\Psi$ for which each set is of
size at least $(\alpha - \beta)n$ and each pairwise intersection is of size at most $(\alpha^2 + \beta)n$. To this end
consider an algorithm, which picks at random $N$ vectors of $n$ bits each, by $N^2$ independent coin
tosses. Assume that each coin is biased so that the probability of a coin toss yielding 1 is $\alpha$ and
define the following events:

**event A:** There is at least one vector having less than $(\alpha - \beta)n$ 1s.

**event B:** There is at least one pair of vectors with at least $(\alpha^2 + \beta)n$ 1s in common.

**event C:** either $A$ or $B$ happens.

Thus by definition, the set system $\Psi$ exists, iff $Pr\{C\} < 1$. In order to establish the later, it suffices
to show that $Pr\{A\} < \frac{1}{2}$ and $Pr\{B\} < \frac{1}{2}$ (since $Pr\{C\} \leq Pr\{A\} + Pr\{B\}$).

To give an upper bound on $Pr\{A\}$, consider the selection of a single random vector, $x$. Call $x$
bad if it has at most $(\alpha - \beta)n$ 1s. Let $S = \sum_{i=1}^{n} x_i$, where $x_i$ is a random variable corresponding
to the result of the choice of the $i$th entry of $x$. Denote $m = an$, so $m$ is the expected number of
1s in the vector. Since $x_i$ are independent 0-1 random variables, we can use the following Chernoff
inequality [1, 2]:

$$Pr(S \leq (1 - \epsilon)m) \leq e^{-\epsilon^2 m/2}, \quad 0 < \epsilon < 1$$

(2)

Putting $m = an$, $n = C \log N$ and $\epsilon = \frac{\beta}{\alpha}$ we get

$$Pr\{x \text{ is bad} \} = Pr(S \leq (\alpha - \beta)n) \leq e^{\frac{\beta^2 an}{2\alpha}} \leq N^{-\frac{C R^2 \log e}{2\alpha}}$$

(3)
Thus, with a judicious choice of \( \alpha, \beta, C \), the probability that \( \bar{x} \) is bad, can be made rather small. The probability that any of the \( N \) random vectors is bad can be estimated by:

\[
\Pr\{A\} < N \cdot N^{\frac{-C\beta^2 \log \varepsilon}{3\alpha^2}}
\]  

(4)

To give an upper bound on \( \Pr\{B\} \), consider the random choice of a pair of random vectors, \( \bar{x}, \bar{y} \). Let \( S = \sum_{i=1}^{n} z_i \), where \( z_i \) is a random variable corresponding to the result of the choice of the \( i \)th entry of \( \bar{x} \) and \( \bar{y} \):

\[
z_i = \begin{cases} 
1 & \text{if } x_i = y_i = 1 \\
0 & \text{otherwise}
\end{cases}
\]  

(5)

Denote \( m = \alpha^2 n \), so \( m \) is the expected number of 1s which are common to \( \bar{x} \) and \( \bar{y} \). Since \( z_i \) are independent 0-1 random variables, we can use the following Chernoff inequality [1, 2]:

\[
\Pr\{S \geq (1 + \varepsilon) m\} \leq e^{-\varepsilon^2 m/3}, \quad 0 < \varepsilon < 1
\]  

(6)

Putting \( \varepsilon = \beta/\alpha^2 \), \( m = \alpha^2 n \) and \( n = C \log N \) we get

\[
\Pr\{\bar{x}, \bar{y} \text{ is a bad pair}\} \leq \Pr\{S \geq (\alpha^2 + \beta) m\} \\
\leq e^{-\frac{\beta^2 \alpha^2 n}{3\alpha^2}} \\
= N^{\frac{-C\beta^2 \log \varepsilon}{3\alpha^2}}
\]  

(7)

Thus one gets:

\[
\Pr\{B\} \leq \left( \frac{N}{2} \right) N^{\frac{-C\beta^2 \log \varepsilon}{3\alpha^2}} < \frac{1}{2} N^{\frac{-C\beta^2 \log \varepsilon}{3\alpha^2}}
\]  

(8)

Finally, if we require that \( \Pr\{A\} < \frac{1}{2} \) and \( \Pr\{B\} < \frac{1}{2} \) and use (4) and (8), we get the following sufficient conditions for the existence of a set system \( \Psi \):

\[
C^2 \beta \log \varepsilon \geq 6\alpha^2 \quad \text{and} \quad C^2 \beta \log \varepsilon \geq (2 + \frac{1}{\log N})\alpha
\]  

(9)

step 2:

The second and last step of the proof is to set \( a = (\alpha - \beta) \) and require that the maximal size of pairwise intersection is at most \( a(1 - d)n \). This imposes the following condition on the maximal possible value of \( d \):

\[
(\alpha - \beta)n - (\alpha^2 + \beta)n = (\alpha - \beta)d \cdot n
\]  

(10)

this shows that for a given value of \( d \) \( \alpha \) must be chosen so as to satisfy

\[
\alpha + d < 1
\]

Rearranging (10) and squaring both sides of the resulting equation, one gets:

\[
\alpha^2 (1 - d - \alpha)^2 = 4\beta^2 (1 - \frac{d}{2})^2
\]  

(11)

There are now two cases to consider:

\[
\alpha \geq \frac{1}{3}.
\]
In this case the two inequalities (9) are satisfied iff the left one is satisfied. Thus we may choose \( \beta \) such that:

\[
\beta^2 = \frac{6\alpha^2}{C \log e}
\] (12)

Upon plugging (12) in (11) and rearranging one obtains:

\[
C = \frac{24(1 - \frac{d}{2})^2}{\log e (1 - d - \alpha)^2}
\] (13)

Since \( \alpha + d < 1 \) and \( \alpha \geq \frac{1}{3} \) \( C \) is minimized for \( \alpha = \frac{1}{3} \) and we get:

\[
C = \frac{24(1 - \frac{d}{2})^2}{\log e (\frac{2}{3} - d)^2}
\] (14)

\[
\alpha \leq \frac{1}{3}
\]

In this case the two inequalities (9) are satisfied iff the right one is satisfied. Thus we may choose any \( f_3 \) such that:

\[
\beta^2 > \frac{2\alpha}{C \log e}
\] (15)

Using (15) in (11) and rearranging we get that a solution can be found for any \( C \) satisfying:

\[
C > \frac{8(1 - \frac{d}{2})^2}{\log e \cdot \alpha (1 - d - \alpha)^2}
\] (16)

The expression in (16) is minimized for \( \alpha = \frac{1-d}{3} \). Thus we get that a solution can be found for any \( C \) satisfying:

\[
C > \frac{54(1 - \frac{d}{2})^2}{\log e \cdot \alpha (1 - d)^3}
\] (17)

Finally we show that (17) gives smaller bounds than (14). To this end it suffices to show that for all allowed values of \( \alpha, \beta, d \), one cannot have:

\[
\frac{4}{(\frac{2}{3} - d)^2} < \frac{9}{\alpha (1 - d)^3}
\] (18)

Setting \( x = 1 - d \) this is equivalent to:

\[
4x^3 - 9x^2 + 6x - 1 < 0
\] (19)

But since \( 4x^3 - 9x^2 + 6x - 1 = 4(x - 1)^2(x - \frac{1}{4}) \) this holds only if \( x < \frac{1}{4} \) i.e if \( d > \frac{3}{4} \). However since one must have \( \alpha < 1 - d \) one has in this case \( \alpha < \frac{1}{3} \) and (14) is not valid.

In order to implement the "broadcast to some" broadcast mode we the following concept:

**Definition 3.1** A \((\gamma, t)\)-robust address space is an address space for \( N \) addresses, of size \( n = C \log N \) bits, such that every vector has at least \( C \alpha \log N \) 1s and each \( t \) vectors satisfy:

\[
| (\bigcup_{i=1}^{t} V_i) \cap V_j | \leq C(1 - \gamma) \alpha \log N \quad \text{for} \quad V_j \neq V_t, \ldots, V_i
\] (20)
Corollary 3.1 Let $N$ be a natural number. For any

\[
C \geq \frac{54(\frac{1}{3} + \frac{1-\gamma}{2\gamma})^2}{\log e(\frac{1-\gamma}{2\gamma})^3}
\]

there exists a $(\gamma, t)$-robust address space of size $C \log N$.

Proof: The corollary follows from Theorem 3.1 and the following upper bound:

\[
| (\bigcup_{i=1}^{t} V_i) \cap V_j | = | \bigcup_{i=1}^{t} V_i \cap V_j | \leq t \cdot \max(V_i \cap V_j)
\]

since by (21) a set system with $d$ satisfying $t(1-d) = 1 - \gamma$ is a $(\gamma, t)$-robust address space.

\section{Intersection of Two Addresses is Large}

In section 3 it was shown that there exist sets of $N$ vectors of $O(\log N)$ bits, where each vector has many 1s but much less 1s in common with each of the other vectors in the set. In these constructions, the maximum 1s that two vectors share at the same positions is taken to be $O(\log N)$. A natural question is whether we can do better than this. That is, can we construct an address space of $N$ vectors of $O(\log N)$ bits, but the largest number of common positions for any pair of vectors which contain 1s is $\phi(N)$ where $\frac{\phi(N)}{\log N} \rightarrow \infty$? This section answers this question negatively.

We may reformulate the problem as follows: Let $[n] = \{1, 2, \ldots, n\}$.

Definition 4.1 A set $F$ of subsets of $[n]$ is called a set-system. Let $L$ be a set of nonnegative integers, a set-system $F = \{A_1, A_2, \ldots, A_N\}$ is $L$-intersecting if for each $i, j$ (i $\neq$ j), $|A_i \cap A_j| \in L$.

As an example of such a set system, consider the following 3-bits vectors, where every two vectors have no more than one '1' in common:

\[
\text{SetSystem} = \begin{cases}
0 & 0 & 0, \\
0 & 0 & 1, \\
0 & 1 & 0, \\
0 & 1 & 1, \\
1 & 0 & 0, \\
1 & 0 & 1, \\
1 & 1 & 0
\end{cases}
\]

In this setting, a set of $N$ 0-1 vectors of $N$ entries is actually a set-system of size $N$. Each vector $\vec{x}$ corresponds to a single set $s_\vec{x}$ from the set-system, where $s_\vec{x}$ is the set of indexes of the 1 entries in $\vec{x}$.

There is a rich literature concerning upper bounds on the size of a set-system with restricted intersections. We will use a recent, stronger version of the well known Ray-Chaudhuri - Wilson theorem:

Theorem 4.1 [BF88], Ex. 4.4.1. Let $K = \{k_1, \ldots, k_r\}$ and $L = \{l_1, \ldots, l_s\}$ be two sets of nonnegative integers and assume $k_i > l_j$ for every $i, j$. Let $F$ be an $L$-intersecting family of subsets of a set of $N$ elements. Assume the size of every set in $F$ is a number from $K$. Then

\[
|F| \leq \binom{n}{s} + \binom{n}{s-1} + \cdots + \binom{n}{s-1 + 1}
\]

(22)
Putting \( n = C \log N \), \( s \leq n/2 \) and
\[
H(\delta) = -\delta \log \delta - (1 - \delta) \log(1 - \delta) \quad 0 < \delta < 1
\]
the entropy function, we get
\[
|\mathcal{F}| \leq \left( \begin{array}{c} n \\ s \end{array} \right) + \left( \begin{array}{c} n \\ s - 1 \end{array} \right) + \cdots + \left( \begin{array}{c} n \\ s - 1 + 1 \end{array} \right)
= \left( \begin{array}{c} n - s \\ n - s \end{array} \right) + \left( \begin{array}{c} n \\ n - s + 1 \end{array} \right) + \cdots + \left( \begin{array}{c} n \\ n \end{array} \right)
\leq \left( \frac{n - s}{n} \right)^{-(n-s)} \left( \frac{s}{n} \right)^{-s}
= 2^{nH\left(\frac{s}{n}\right)} = N^{C H\left(\frac{s}{n}\right)}
\]
Thus, for an address space of \( N \) vectors we need \( CH\left(\frac{s}{n}\right) \geq 1 \). However \( C \) is constant, so if
\[
\frac{s}{n} = \frac{\phi(N)}{C \log N} N \rightarrow 0
\]
then \( H\left(\frac{s}{n}\right) \rightarrow 0 \), so we cannot always have \( N \) vectors intersecting in so few entries. This proves that, asymptotically, the result from section 3 is the best possible.

5 \( O(\log^2 N) \)-bit Addresses Construction

We have seen that the result from section 3 is asymptotically optimal. However it has several fundamental flaws: first, it proves the existence of an appropriate address space, rather than give a particular construction. In systems that need to find on-line a new address space (say, for a given \( N \)) the time required for the construction is large. Suppose that the randomized algorithm that is given is used, and suppose that we want high probability of success in small number of iterations. Then \( n \approx C \log N \) becomes larger than the minimum possible, since the probability is dependent on the deviation from the average, which in turn, depends on \( C \).

Even when the set of vectors is already found, it has an irregular structure and need to be stored at the local memory of each of the processors using it. I.e., a local memory of \( CN \log N \) bits is required for each processor.

Another problem is that for moderate values of \( N \), the \( C \) that we have guaranteed, might be higher than \( \log N \). Thus, for some practical cases, \( \log^2 N \leq C \log N \). One may claim that, since the proof is a simple counting sieve, it is probably possible to find address spaces with "shorter" vectors. Although this is probably true, it is not guaranteed and the algorithm randomly choosing vectors fail.

In this section we show that if the asymptotic requirements of the algorithm are loosened a bit, then all the above problems are immediately solved. We use the formalism of section 4, where an address space corresponds to a set-system, and sets belonging to the set-system correspond to address 0-1 vectors. Only that it is not necessary that sets in the set-system are built of elements of \([n]\). In this section, for example, the world from which elements are taken is the set of points of the integral plane.

**Definition 5.1** A set-system for which all sets contain exactly \( k \) elements, is called \( k \)-uniform.
Theorem 5.1 [BF88], pp. 47 For every \( k \geq s \geq 1 \) and \( n \geq 2k^2 \), there exists a \( k \)-uniform family \( \mathcal{F} \) of size \( > (n/2k)^s \) on a world of \( N \) points, such that \( |E_1 \cap E_2| \leq s - 1 \) for any two distinct sets \( E_1, E_2 \in \mathcal{F} \).

In our setting \( k \geq s - 1 + D_{th} \), so given \( N \) and \( D_{th} \) we are interested in the minimal \( s \) for which \( (s - 1 + D_{th})^s \geq N \). It is easy to see that for any \( D_{th}, 1 \leq D_{th} \leq E \log N, s = O\left(\frac{\log N}{\log \log N}\right) \) is sufficient, so for \( D_{th} = O\left(\frac{\log N}{\log \log N}\right) \)

\[
n = O\left(\frac{\log N}{\log \log N} \right)^2.
\]

The constant is rather small and becomes even smaller when \( N \) grows. For \( D_{th} = D \log N \)

\[
n = O((1 + D)^2 \log^2 N)
\]

which, for many \( N \)s and \( D \)s, is much better than what is guaranteed in section 3.

Next we give the proof of theorem 5.1. The proof is important, as it actually gives a construction for the address space.

**Proof of theorem 5.1:** Let \( p \) be the greatest prime \( \leq n/k \), so \( n/2k < p \leq n/k \). Fix a \( k \)-subset \( A \) of the field \( GF(p) \). Recall that \( n \geq 2k^2 \), so \( p \geq k \).

Let \( X \) be a set of \( N \) points containing \( A \times GF(p) \). For any function \( f : A \rightarrow GF(p) \), the "graph" \( G(f) = \{((\xi, f(\xi)) : \xi \in A\} \) is a \( k \) - points subset of \( X \). Our set - system consists of all the graphs of the polynomials of degree \( \leq s - 1 \) over \( GF(p) \), restricted to \( A \). For two different polynomials of degree \( \leq s - 1 \), their graphs have at most \( s - 1 \) points in common. The number of such polynomials is \( p^s > (n/2k)^s \).

6 Optical Realization of Message Switching Systems

In the previous sections it has been shown that a set-system of \( N \) vectors (each of size \( c \log N \) bits) can be devised, such that the size of the intersection of any two vectors does not exceed a certain threshold. But how can such a set system be used to construct a free space optical multiplexor? It is beyond the scope of this paper to describe a specific and detailed optical realization. Rather, we proceed in this section to show that in principle, such a system can be constructed, using simple optical components.

The set system is used to select one of the \( N \) destinations through the following sequence of operations (see fig. 1):

1. The input vector is set to the destination address. Optically, the input vector should be a source of light beams (one for every bit set to one). For example, laser-diod arrays may be used, which is commercially available.

2. The intersection of the input vector with all the vectors of the set system is carried as follows. The set system is realized by a simple transparency, in which '0' bits are dark and '1' bits are transparent. The intersection operation is a direct illumination of the set-system transparency by light rays corresponding to the '1' bits of the input vector.

3. By the properties of the set-system underlying the address space it is guaranteed that only the intersection of the input vector with destination address, exceeds a prespecified threshold. By "blocking" all outcomes of the previous step whose intensity is less than the threshold, a light beam from the input vector will reach the destination.

The threshold operation can be carried either:
• electrically for each destination, using a detector and a common comparator gate, or
• optically, using non-linear bistable devices ([Fe88] sec. 6.2).

We remark that the above is only a schematic description of an optical realization. As such, it ignores many lower-level implementation details (such as how to focus the light on the threshold devices).

![Diagram of optical multiplexor schema](image)

**Figure 1: Optical multiplexor schema.**

The optical multiplexor described above can be used to implement the three broadcast modes, i.e. point to point, broadcast to some, and broadcast to all. When the input vector matches only one vector of the set system, one message will reach the appropriate destination. This corresponds to the "point to point" communication mode. On the other hand, a processor may "broadcast to all" other processors by using the input vector \( \bar{I} \) whose bits are all set to '1'. In order for a processor to be able to send a message to \( t \) different locations, one of the following approaches may be adopted.

1. Use \( t \) different optical multiplexors for each processor, and hence pay with \( t \times N \) threshold devices.

2. Rather than using the set-system from Theorem 3.1 one may use the multi set-system whose existence is proven by Corollary 3.1. The entries of the input vector are set to the union of the addresses of the \( t \) destinations. Corollary 3.1 guarantees that the receiving destinations are precisely the desired ones.
6.1 General Configuration for a Message Switching System

So far we have discussed the realization of a single optical multiplexor (OM). We now discuss how to assemble \( N \) multiplexors and detectors to form a complete architecture. The following parameters, which are the subject of minimization in this paper, are associated with every system which uses \( N \) optical multiplexors for communication.

1. The area \( A_{\text{system}} \) is proportional to the total number of set-system bits. The this definition is motivated by the following facts:
   - The largest component in every OM is the set-system transparency. Thus the area of an OM corresponds to the transparency size which is \( CN \log N \).
   - In such a system, all OMs must be placed on the surface of the volume that the system occupies (otherwise light beams destined to inner OMs might be blocked by outer ones).

2. \( V_{\text{system}} \) is the volume associated with the system. We define \( V_{\text{system}} \) to be proportional to \( (A_{\text{system}})^{\frac{1}{2}} \). This is motivated by the fact the sphere attains the minimal volume for a fixed surface area.

A trivial OM architecture can be constructed by placing all \( N \) OMs on one face of a cube, and placing all the \( N \) detectors on the opposite face, such that every detector is visible from every OM, see fig. 2.

![Figure 2: The 'cube' architecture: simple layout of \( N \) OMs.](image)

This cube architecture (called 'cube') has sizes \( A_{\text{cube}} \approx CN^2 \log N \) and \( V_{\text{cube}} \approx N^3(C \log N)^{\frac{3}{2}} \). Two observations may be used for reducing both the area and the volume:

- Let each processor use two OMs denoted by \( OM_x \) and \( OM_y \). Each of \( OM_x \) and \( OM_y \) has \( \sqrt{N} \) destinations, thus the processor selects one destination out of \( N \) by tuning both OMs, so that the destination is selected when both \( OM_x \) and \( OM_y \) light beams hit the same detector (See fig 3).

- The detectors can be made transparent. Thus \( N \) of them can be packed in a square \( \sqrt{N} \times \sqrt{N} \) (See fig 3). Such a detector can be realized by a SEED bistable device ([Fe88] sec. 6.2.4). The SEED bistable device begins to absorb light when the intensity exceeds a certain threshold. In our case a SEED detector will absorb light only when it has been hit by both \( OM_x \) and \( OM_y \), thus a light beam can move through \( \sqrt{N} \) such devices without being completely absorbed.
Another way of implementing a transparent detector is by splitting the light beams into two parts (using half mirrors, or holograms [Fe88]). The major part of the light beam will bypass the detector, while a small fraction will trigger the detector (See fig 4).

![Figure 3: Using two OMs to select a given detector.](image)

![Figure 4: bypassing of an opaque detector.](image)

The general configuration (denoted by 'sqrt') for $N$ processors is described in fig. 5. Basically, each destination is activated by a vertical column detector containing $N$ transparent sub-detectors. Every processor has two OMs, $OM_x$ and $OM_y$, each having $\sqrt{N}$ destinations. $OM_x$ and $OM_y$ are placed in a horizontal plane. All $N$ columns cross every horizontal plane which is associated with all possible cross points of $OM_x$ and $OM_y$. When a sub-detector of column $j$ is hit by light beams from both $OM_x$ and $OM_y$, the column detector is activated, so that a message of one bit is being sent from processor $i$ to destination $j$.

The sizes of the 'sqrt' architecture are:

$$A_{sqrt} \approx 2N \cdot \sqrt{N} \cdot C \log \sqrt{N} = CN^{\frac{3}{2}} \log N$$
$$V_{sqrt} \approx N^2 (C \log N)^{\frac{3}{2}}$$

Which is a significant improvement compared to the sizes of 'cube'.

14
Figure 5: The 'sqrt' architecture $OM^{sqrt}$.

7 Conclusion

We have described a new optical method to realize message routing, using set-systems, which reduces the number of address bits to $O(\log N)$, while maintaining the sensitivity of the threshold devices. The method obtains reduced cross-talk by encoding the input addresses in the set-systems. The results show that all three communication modes, namely point to point, broadcast to some, and broadcast to all, may be supported while keeping a low cross-talk.

We gave a brief description of how the optical devices of NOMs may be packed in 3-D space to form a parallel message switching system. In particular, it is suggested that packing detectors in 3-D space (rather than merely using its surface) may result in a substantial reduction of the size of the system.
References


