Design of Wavelength Converting Switches for Optical Burst Switching

Jeyashankher Ramamirtham, Jonathan Turner, Joel Friedman

***Abstract*— Optical Burst Switching (OBS) is an experimental network technology that enables the construction of very high capacity routers, using optical data paths and electronic control. In this paper, we study *wavelength converting switches* using tunable lasers and wavelength grating routers, that are suitable for use in optical burst switching systems and evaluate their performance. We show how the routing problem for these switches can be formulated as a combinatorial puzzle or game, in which the design of the game board corresponds to the pattern of permutation used at the input sections of the switch. We use this to show how the permutation pattern affects the performance of the switch, and to facilitate the design of interconnection patterns that yield the best performance. We give upper bounds on the number of different wavelength channels that can be routed through such switches (regardless of the interconnection pattern), and show that for 2×2 switches, there is a simple permutation pattern that achieves these bounds. For larger switches, randomized permutation patterns produce the best results. We study the performance of optical burst switches using wavelength converting switches based on several different permutation patterns. We also present a novel routing algorithm called the *most available wavelength* assignment and evaluate its benefits in improving the switch throughput. Our results show that for a typical configuration, the switch with the best permutation pattern has more than 87% of the throughput of a fully nonblocking switch.**

***Index Terms*—Optical Burst Switching, Wavelength Converting Switches, Wavelength Routers**

I. INTRODUCTION

The transmission capacity of optical fibers has been increasing at a tremendous rate as a result of DWDM technology. Although terabit capacity IP routers based on electronics are now starting to appear, there remains a serious mismatch between the transmission capacity of DWDM fibers and the switching capacity of electronic routers. Since DWDM links are capable of supporting hundreds of channels operating at rates of 10 Gb/s each, it can take 5-10 equipment racks to hold the electronic line cards needed to terminate the channels from just a single fiber. Optical burst switching seeks to reduce the cost and complexity of these systems by replacing much of this electronics with optical components. OBS is a hybrid switching technology that uses electronics to control routing decisions, but keeps data in optical form as it passes through each OBS router. By exploiting the high channel counts of advanced WDM systems, it achieves excellent statistical multiplexing performance with little or no buffering. Further details on OBS can be found in [1], [2], [3], [4], [5], [6].

In this paper we focus on the design of the basic switch elements that are used to construct large OBS routers. In particular, we study the design of the wavelength converting switches that are the key building block needed to implement these systems. Although, there have been a number of studies of optical packet switching in recent years [7], [8], [9], [10], [11], [12], [13], it is not yet clear how the required optical components can be implemented to make them cost-competitive with electronic alternatives.

Recent dramatic advances in tunable lasers have created new architectural options for wavelength converting switches and appear to hold considerable promise for the design of practical optical switching systems. This paper examines a specific type of wavelength converting switch that uses tunable wavelength converters and passive Wavelength Grating Routers (also known as Arrayed Waveguide Grating Multiplexors or AWGMs). This type of switch is attractive because WGRs are relatively simple to fabricate, are inexpensive and consume no power. Unfortunately, the use of wavelength routers introduces the possibility of blocking. In this paper we evaluate the impact of this blocking on the statistical multiplexing performance of a switch element in an OBS router. Our results show that the WGR-based switch can achieve more than 87% of the throughput obtained with a fully nonblocking switch.

We show that the performance of a WGR-based switch is strongly dependent on the pattern of permutation used at the input sections of the switch. This is done by first formulating the routing problem in the WGR switch as a combinatorial puzzle or game, in which the design of the game board corresponds to the permutation patterns. We use this correspondence to explore alternative designs and evaluate the performance of the resulting system, using simulation. Also, we show that the puzzle can be viewed as a bipartite matching problem. This leads directly to a method for rearranging existing connections in a switch to accommodate new connections that might otherwise have to be rejected. We also present a novel routing algorithm to forward the bursts through the switch without rearrangement and evaluate the performance of the algorithm through simulation. We find that the routing algorithm is helpful in improving performance for regular permutation patterns but is not that beneficial for random game boards.

The rest of the paper is organized as follows. In Section II, we give a brief overview of the optical burst switching concept and explain where this work fits in the OBS context. We present the design of the WGR-based wavelength converting switch in

J. Ramamirtham and J. Turner are with Applied Research Laboratory, Department of Computer Science and Engineering, Washington University in St. Louis, MO, USA. E-mail: `{jai,jst}` `@arl.wustl.edu`. This work is supported by the Advanced Research Projects Agency and Rome Laboratory (contract F30602-97-1-0273)

J. Friedman is with the departments of Computer Science and Mathematics, University of British Columbia, Vancouver, BC V6T 1Z4 (V6T 1Z2 for Mathematics), CANADA, `jf@cs.ubc.ca`. Research supported in part by an NSERC grant.

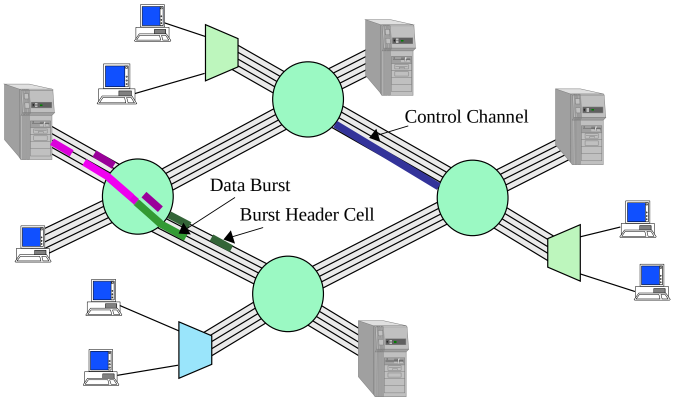


Fig. 1. Burst Switching Concept

Section III. In Section IV, we show how to model blocking in the WGR-based switch as a combinatorial puzzle and show how the problem of solving the puzzle can be reformulated as a matching problem in a bipartite graph. We study the design of game boards and the "solvability" of puzzles on these game boards in Section V. In Section VI, we present a game formulation that models the routing of bursts, as they arrive in an actual burst switch. We present simulation results showing how the blocking characteristics of the switch designs affect the statistical multiplexing performance of a switch element in an OBS router. We also describe a routing algorithm for forwarding bursts and evaluate its performance through simulation. Finally, we present some concluding remarks in Section VII.

II. Burst Switching Architecture

The basic burst switching concept is illustrated in Fig. 1. The transmission links carry data on tens or hundreds of wavelength channels and user data bursts can be dynamically assigned to any of these channels by the OBS routers. One (or possibly several) channel on each link is reserved for control information that is used to control the dynamic assignment of the remaining channels to user data bursts. When an end system has a burst of data to send, an idle channel on the access link is selected and the data burst is sent on that channel. Shortly before the burst transmission begins, a *Burst Header Cell* (BHC) is sent on the control channel, specifying the channel on which the burst is being transmitted and the destination of the burst. The OBS router, on receiving the BHC, assigns the incoming burst to an available channel on the outgoing link leading toward the desired destination and establishes a path between the specified channel on the access link and the channel selected to carry the burst. It also forwards the BHC on the control channel of the selected link, after modifying the cell to specify the channel on which the burst is being forwarded. This process is repeated at every router along the path to the destination. The BHC also includes an *Offset* field which contains the time between the transmission of the first bit of the BHC and the first bit of the burst, and a *Length* field specifying the time duration of the burst. The offset and length fields are used to time the switching operations in the OBS routers, and the offset field is adjusted by the routers to reflect variations in the processing delays encountered in the routers' control subsystems. If a router does not have idle channels available at the output port, the burst can be stored in a buffer, or in a bufferless system, is discarded.

Reference [3] describes a scalable OBS router architecture consisting of a set of *Input/Output Modules* (IOM) that interface to external links and a multistage interconnection network of *Burst Switch Elements* (BSE). The interconnection network uses a Benes topology, which provides parallel paths between any input and output port. A three stage configuration comprising d port switch elements can support up to d^2 external links (each carrying many WDM channels). The topology can be extended to 5, 7 or more stages. In general, a $2k-1$ stage configuration can support up to d^k ports. For example, a 5 stage network constructed from 8 port BSEs would support 512 ports. If each port carried 256 channels at 10 Gb/s each, the aggregate system capacity would be $1,310$ Tb/s.

Input IOMs process the arriving BHCs, performing routing lookups and inserting the number of the output IOM into BHCs before passing them on. The BSEs use the output port number to switch the burst through to the proper output. Each of the components that does electronic processing on the cell keeps track of the time spent and updates the offset field in the BHC to maintain synchronization with the burst. Additional details can be found in [3].

III. WGR-Based Switch Design

Each BSE in a burst switch requires a wavelength converting switch, capable of switching an optical signal from any of the BSE's d input fibers to any of its d output fibers (Fig. 2). A BSE with $d = 8$ and $h = 256$ wavelengths would have an aggregate throughput of 2 Tb/s, assuming 10 Gb/s per wavelength. Since bursts can arrive at unpredictable times, a BSE must be able to switch bursts to different wavelengths on the output fibers, in order to provide acceptable statistical multiplexing performance at typical traffic intensities. Wavelength conversion technologies are discussed in [14], [15].

The WGR-based switch design we are interested in is shown in Fig. 3. Each of the d input sections consists of four components, an *optical demultiplexor*, a bank of *tunable wavelength converters*, a *wavelength grating router* and a bank of *optical multiplexors*. The router and the multiplexors are joined through a fixed *permutation pattern*. We will see that the blocking characteristics of the switch depend critically on the choice of this permutation pattern. Each output section consists of a

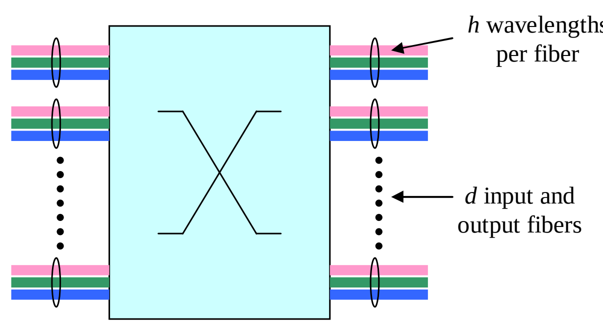


Fig. 2. Wavelength converting switch with d input/output fibers and h wavelength channels per fiber

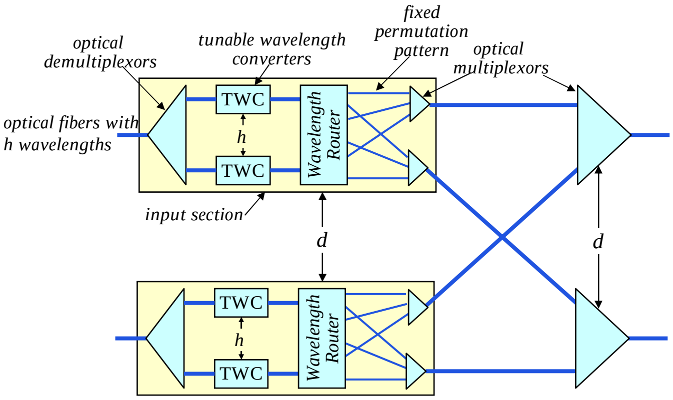


Fig. 3. Wavelength switch using Tunable Wavelength Converters (TWC) and Passive Wavelength Grating Routers (WGR)

	λ_0	λ_1	λ_2	λ_3
I_0	O_0	O_1	O_2	O_3
I_1	O_1	O_2	O_3	O_0
I_2	O_2	O_3	O_0	O_1
I_3	O_3	O_0	O_1	O_2

Fig. 4. Routing matrix of a 4×4 WGR

single optical multiplexor. Each input section is connected to each output section by an optical fiber carrying up to h optical signals on different optical wavelengths. Since each wavelength can be used only once on each output fiber, the signals arriving at an output section from different input sections must use distinct wavelengths.

In high performance optical networks, hundreds of different optical signals may be carried on a single fiber, using different wavelengths of light. The optical demultiplexor in each input section separates these signals, so that they can be individually switched to different output fibers. The tunable wavelength converters use a tunable laser and an optical modulator to transfer the information carried on one input wavelength to a different (and dynamically selectable) output wavelength. This wavelength conversion is needed to allow input signals on different input fibers to be switched to the same output fiber, even if the input signals are carried on the same wavelength. The wavelength grating router is a passive optical device that switches optical signals based on their wavelength. Specifically, an optical signal carried on wavelength i at input j is switched to output $(j+i) \bmod h$ where h is the number of wavelengths. Thus, the choice of wavelength used for a given signal determines which output section the signal is forwarded to. In general, there are h/d different wavelengths that can be used by a given input signal to reach a different output, but different inputs may use different sets of wavelengths to reach the same output. The permutation patterns used in each of the input sections determine these wavelength sets. Since a given input channel is not able to use any wavelength to reach a given output fiber, blocking can occur. That is, there may be situations where all of the wavelengths that can be used to get to a desired output are in use at the output causing blocking to occur, even when there are free wavelengths available on the outgoing link.

There is an alternative nonblocking design that replaces the WGR with a crossbar. In a system using such a switch, a signal from an input can be routed to any output if there are fewer than h signals already being sent to the output. This design is significantly more complex than the WGR-based design because WGRs are much simpler devices than crossbars. We believe that WGR-based switches may offer a more cost-effective solution. We show that the performance of switches using WGRs is close to that achieved with nonblocking switches. A demonstration of a combination of tunable wavelength converters and wavelength routers has been discussed in Reference [16].

IV. Design of WGR-based switches

In this section we study the routing problem in WGR-based switches and show how the blocking performance of these switches is affected by the permutation pattern used within each of the input sections of the switch. First however, we briefly review the routing properties of wavelength grating routers.

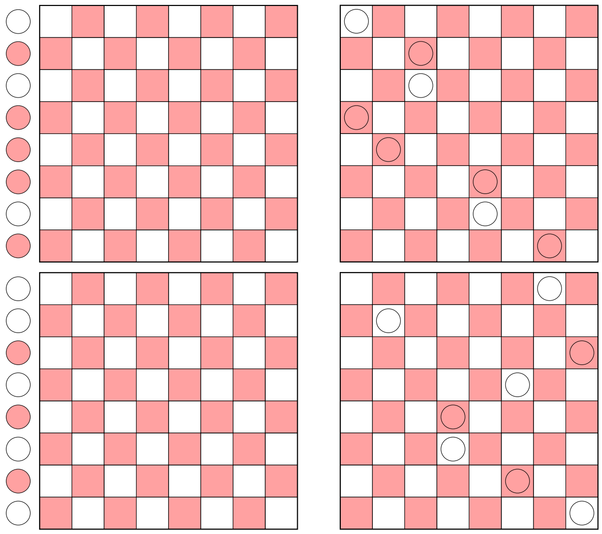
A. *Wavelength Grating Router properties*

An $h \times h$ WGR is a passive static wavelength-routing device that provides complete connectivity between its inputs and outputs, by passively routing h^2 optical connections on h wavelengths [17]. The use of a WGR has several advantages including easy fabrication, commercial availability and relatively low cost. A WGR has a fixed cyclical-permutation-based routing pattern between its input and output ports. A connection at input i using wavelength k gets routed to the same wavelength on output $(i+k) \bmod h$, $\forall\, i\,, k \in [0, h-1]$. The routing pattern for a 4×4 WGR is shown in Fig. 4. A connection at input I_2 using wavelength λ_3 gets routed to output O_1 and a connection at input I_3 using wavelength λ_0 gets routed to output O_3.

B. *Routing Multiple Channels Simultaneously*

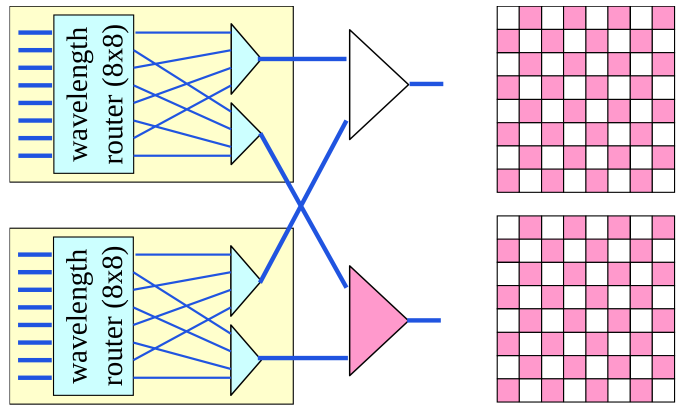
In this section, we show how the problem of simultaneously routing a set of channels through a WGR-based switch can be formulated as a combinatorial puzzle. This formulation makes it easier to understand the intrinsic structure of the problem, yielding insights that are useful in design and analysis.

The puzzle is played on a game board made up of dh^2 squares arranged in h columns and dh rows. The board is divided into d square *blocks* of h rows each. Each square has one of d different colors, with each row containing h/d squares of each color and each column containing h squares of each color. To setup the puzzle we place colored *tokens* beside some or all of the rows. A setup can include at most h tokens of any color. An example of a setup game board with $d = 2$ and $h = 8$ is shown in Fig. 5(a). To solve the puzzle, we must place each token on a square of the same color, in the row where the token was placed. The token placement must also satisfy the constraint that no two tokens of the same color be placed in the same column. An example solution to the puzzle is shown in Fig. 5(b).

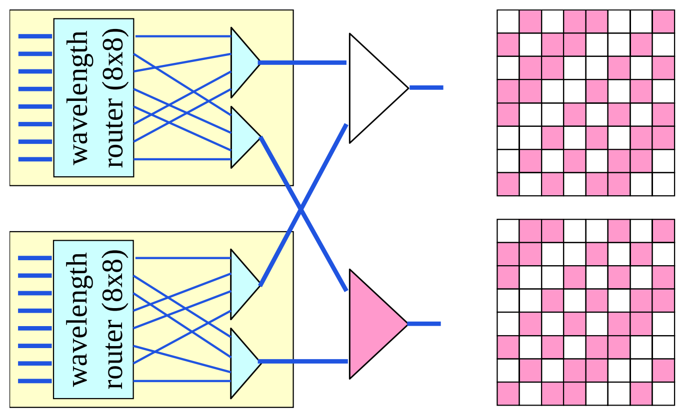


(a) Example puzzle setup (b) Example solution

Fig. 5. An example puzzle setup and solution



(a)



(b)

Fig. 6. Two configurations and the corresponding game boards of a system with $d = 2$ and $h = 8$

Each row in the puzzle corresponds to one of the h input channels on one of the d input fibers. More specifically, row i in block j of the game board corresponds to input channel i of input fiber j. The color of the token that is placed by a row corresponds to the output fiber that the corresponding input channel is to be switched to. More specifically, placing a token of color r on row i of block j corresponds to switching channel i of input fiber j to output fiber r. The columns of the array correspond to different output wavelengths. Placing a token in a particular column corresponds to choosing that output wavelength. The color of each square corresponds to the output that is reached if the wavelength converter for the input channel corresponding to that square's row is tuned to the wavelength corresponding to the column. So, placing a token of color r in column q of row i of block j corresponds to switching channel i of input fiber j to channel q of output fiber r. Note that the puzzle rule requiring that no two tokens of the same color occupy the same column, corresponds to the requirement that no two input signals going to the same output fiber use the same wavelength.

In order to complete the correspondence between the puzzle and the routing problem, we note that within each block, the rows must have closely related color patterns, in order to model the routing characteristics of the WGRs. Specifically, the pattern of colors within each row can be obtained from the previous row's pattern by a cyclic rotation of one column. This relationship only holds within each block. There is no requirement that different blocks have similar color patterns. The color pattern for each block corresponds to the permutation pattern within the input sections of the switch. This is illustrated in Fig. 6 which shows two example configurations of a system with $d = 2$ and $h = 8$ and the corresponding game boards.

Whenever the puzzle has a solution, it means that there is a way to route the input signals to the output channels that are specified by the tokens placed by each row. If the puzzle does not have a solution, then there is no way to route all the channels simultaneously. If, for all possible puzzle setups, there is a solution, the switch is *rearrangeably non-blocking*. It is easy to see that the switch in Fig. 6a is not rearrangeably non-blocking, since the puzzle setup in which tokens of one color are placed in even-numbered rows and tokens of the other color are placed in odd-numbered rows, has no solution. On the other hand, this setup does have a solution when played on the game board in Fig. 6b.

To generalize the problem of routing connections simultaneously, we restrict the number of tokens of a particular color to some value $k \leq h$.

Definition IV.1: A game board is *k-solvable* if every puzzle setup with at most k tokens of each color has a solution.

We show below that no game board is h-solvable. Fortunately, in practice, it can be sufficient to find game boards that are k-solvable for values of k fairly close to h.

C. Routing problem as a bipartite matching problem

The problem of solving the puzzle can be reformulated as a matching problem in a bipartite graph. We start by constructing

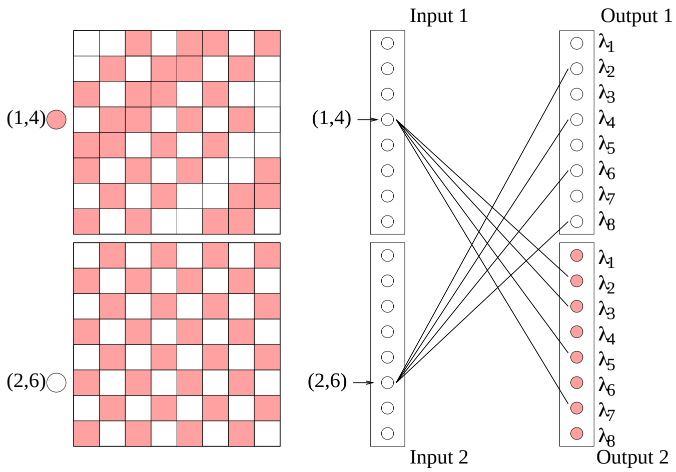


Fig. 7. Example bipartite graph formulation of a puzzle

the bipartite graph. The graph consists of two subsets of vertices; let us call them the "left" and "right" subsets. The "left" subset includes vertex $u(i,j)$ that corresponds to row j in block i of the game board (or channel j of input fiber i). The "right" subset includes a node $v(q,r)$ that corresponds to color q and column r (or channel r of output fiber q). We include the edge $\{u(i,j), v(q,r)\}$ in the graph if the color of the token in row j of block i is q and the color of the square in column r of row i and block j is q. In terms of the WGR switch, you include the edge if the signal on input fiber i, channel j is to be switched to output fiber q, and you can reach output fiber q using wavelength r.

In each row of the game board, there are h/d squares of the color corresponding to output q or h/d wavelengths that switch signals to output q. Thus, for each token that is placed at row j of block i, there are h/d edges that are drawn from the "left" subset to the "right" subset. This is illustrated in Fig. 7 for a system with 2 input fibers and 8 wavelengths per fiber. The token placed in row 4 of block 1 has four squares of the same color as the token's. This results in 4 edges in the bipartite graph from vertex $(2,4)$ to vertices corresponding to output 2. Similarly, 4 edges are drawn from vertex $(2,6)$ for the token placed at row 6 of input fiber 2.

Now find a maximum size matching in this bipartite graph. If there is an edge for every token, you have a solution. In particular, if $\{u(i,j), v(q,r)\}$ is in the matching, then put the token in block i and row j in column r (equivalently, tune the wavelength converter for channel j of input fiber i to output wavelength r).

One observation about this graph is that it breaks apart into separate subgraphs corresponding to the different outputs. This just corresponds to the fact that the placement of tokens of one color is independent of the placement of tokens of other colors. Using a well-known maximum size matching algorithm based on max flows in unit networks, one can solve the puzzle in $O(h^{5/2})$ time, assuming a solution exists.

The connection with matching also yields an algorithm for rearranging existing connections to accommodate new ones, which corresponds directly to the augmenting path algorithm for bipartite graphs. This can be described in terms of the game board as follows. If it is not possible to place a token of color x in row r_1 (satisfying the required constraints), then find an "x-augmenting path" in the game board starting at some square of color x in row r_1. Call this square (r_1, c_1) where c_1 is the column number. An x-augmenting path in the game board is a sequence of squares $(r_1,c_1),(r_2,c_2),(r_3,c_3),...,(r_m,c_m)$ satisfying the following properties:

- All squares in the sequence have color x.
- Squares $(r_2,c_1),(r_3,c_2),...,(r_m,c_{m-1})$ all contain a token of color x.
- There is no token of color x in column c_m.

Given such a path, we place a token on (r_1,c_1) and for $1 < i \leq m$ we move the token on square (r_i, c_{i-1}) to (r_i, c_i).

We can find such a path by performing a breadth-first search through the game board. We construct the search tree as follows. The nodes, N_i $(1 < i \leq hd)$, of the tree correspond to the rows of the game board. To place a token of color x in row r_1, let N_{r_1} be the root node of the tree. If column c in row r_1 has a square of color x and row r_k has a token of color x in column c, then we add an edge from N_{r_1} to N_{r_k} in the tree. There are h/d nodes adjacent to node N_{r_1}. We can construct the tree by recursively adding nodes at distance 1 from the newly added nodes. The search stops when we have a row that has a column with no token of color x placed in it; that is, we can move the token in the row to this free column.

V. Finding Good Game Boards

The design of a game board has a big influence on our ability to solve the puzzle. Since the game board design corresponds to the permutation pattern of the input section, this means that the permutation pattern affects the likelihood of blocking. The game board in Fig. 6a has many puzzle setups that have no solution, making it a poor design, from the perspective of the puzzle solver. What makes it a poor design is that many rows have exactly the same pattern of colors. This means that if tokens of the same color are placed in these rows, the number of columns they have to choose from is limited, and may be smaller than the number of tokens. This suggests that a good game board design will be one in which different rows have different patterns, and in particular, have as few columns in common as possible with squares of the same color.

A game board is k-solvable if and only if in each of its associated bipartite graphs, there is a matching of size t between any set of $t \leq k$ inputs and all its outputs. By the well-known Marriage Theorem for bipartite matching, such a matching exists if and only if every set of $t \leq k$ nodes in the "left" subset has at least t neighbors in the "right" subset. We can restate this in terms of the puzzle, as follows. For a given game board and a fixed color (call it blue), we say that row i *covers* those columns in which it has a blue square. Similarly, we say that a set of rows covers those columns for which there is a blue square in at least one of the given rows. A game board is k-solvable if and only if for all colors j and all $r \leq k$, all sets of r rows cover at least r columns.

A. Upper bounds on puzzle solvability

We first show that no game board is h-solvable. Consider an arbitrary game board and color blue. There are exactly h blue squares in any column of the game board, meaning that there are $dh - h$ squares that are not blue. If we select any h rows from among the $dh - h$ rows that do not have blue squares in the given column, then any puzzle setup that has blue tokens in these h rows is unsolvable, since none of the tokens can be placed in the selected column, and we must place each of the h tokens in a distinct column. Similarly, if we consider any $i \leq d-1$ columns, there must be at least $(d-i)h$ rows that do not contain blue squares in any of these columns. So, any puzzle setup that has blue tokens in more than $h - i$ of these rows is unsolvable. These results make it clear that we cannot expect to construct a WGR-based switch that will guarantee our ability to place more than $h - d + 1$ tokens of the same color. Fortunately, the value of h is typically much larger than d for configurations of practical interest, which means that the degree of blocking implied by this limitation may be acceptable. This gives us

Theorem V.1: For any k-solvable game board on d colors and h columns, $k \leq h - d + 1$

For larger values of d, we can get a stronger bound using the following theorem.

Theorem V.2: Let G be a game board on d colors and h columns and let s be any integer that satisfies

$$dh\big(h - (h/d)\big)^{\underline{s}}/h^{\underline{s}} \geq h - s + 1$$

where $x^{\underline{r}} = x(x-1)\ldots(x-r+1)$. If G is k-solvable, then $k \leq h - s - 1$.

For $h = 256$ and $d = 8$, 15 is the largest value of s that satisfies the inequality, giving a limit of 240 on the solvability of game boards with $h = 256$ and $d = 8$. If we increase d to 16, the largest s increases to 41 and the limit becomes 214. If we fix a value of d and let $h \to \infty$, the theorem implies that $k \leq h - \lceil \log_{d/(d-1)} d \rceil$. Since $\log_{d/(d-1)} d$ is roughly $d\ln(d)$ for larger values of d, we can use $h - d\ln d$ as an estimate of the bound, which for larger values of d is significantly smaller than the $h-d+1$ implied by *Theorem V.1*. However, for $h >> d$, even this stronger bound does not rule out the existence of practically useful game boards.

To prove *Theorem V.2*, consider an arbitrary game board G. Fix a color (call it blue) and choose s random columns from G. For any row r, the probability that r does not have a blue square in any of the s columns is $(h - h/d)^{\underline{s}}/h^{\underline{s}}$. So the expected number of rows that do not have a blue square in any of the s columns is $dh(h - h/d)^{\underline{s}}/h^{\underline{s}}$. So, there must be some set of s columns, S, for which there are at least $dh(h-h/d)^{\underline{s}}/h^{\underline{s}}$ rows that have no blue squares in S. If this number is $\geq h - s + 1$, then the puzzle cannot be $(h-s)$-solvable.

B. Contiguous game boards

A *repetitive game board* is a game board whose d blocks are the same. A *contiguous game board* is a repetitive game board in which the first row of each block is divided into d contiguous monochrome blocks of size h/d each. Remarkably, for $d = 2$, contiguous game boards are $h - 1$ solvable. To see this, fix a color and note that any set of i rows in the same block covers at least $(h/d) + i - 1$ columns. Any set of k rows in the game board must have at least $\lceil k/d \rceil$ rows in some block, and so must cover at least $(h/d) + \lceil k/d \rceil - 1$ columns. For $k = h - 1$, this is $2(h/d) - 1$, which is $h - 1$ when $d = 2$. That is, a contiguous game board with $d = 2$ is $(h-1)$-solvable, matching the upper bound in the previous sub-section. For arbitrary values of d, we have the following theorem.

Theorem V.3: A contiguous game board on h columns with d colors is k-solvable if and only if $k - \lceil k/d \rceil \leq h - 1$. The largest value of k that satisfies this condition is

$$k^* = \begin{cases} (h/(d-1)) - 1, \text{if } d-1 \text{ divides } h/d, \\ (h/(d-1)) - (h/d \bmod d-1)/(d-1), \text{otherwise.} \end{cases}$$

The first part of the theorem follows directly from the discussion above. The second part can be shown by substitution.

C. Random Game Boards

Our criterion for a good game board is one in which any set of r rows covers at least r columns, for each color. For larger values of h and d, we can expect random game boards to do well, in this respect. Consider an arbitrary set of r rows within a single block of a random game board. The probability that a particular column is not covered for some fixed color is

$$\frac{h-r}{h}\,\frac{h-r-1}{h-1}\cdots\frac{h-r-h/d+1}{h-h/d+1} = \frac{(h-r)^{\underline{h/d}}}{h^{\underline{h/d}}}$$

Thus, within one block, the expected number of columns not covered by r rows is

$$h\,\frac{(h-r)^{\underline{h/d}}}{h^{\underline{h/d}}}$$

The expected number of columns not covered by r rows selected from d independent random blocks in the game board is

$$\begin{aligned} &= h\,\frac{(h-r_1)^{\underline{h/d}}}{h^{\underline{h/d}}}\,\frac{(h-r_2)^{\underline{h/d}}}{h^{\underline{h/d}}}\,.....\,\frac{(h-r_d)^{\underline{h/d}}}{h^{\underline{h/d}}} \\ &\leq h\left(\frac{(h-r/d)^{\underline{h/d}}}{h^{\underline{h/d}}}\right)^d \end{aligned}$$

where $r = r_1 + r_2 + \ldots + r_d$, and r_i is the number of rows in the ith block of the game board. The expected number of columns not covered is plotted as a function of the size of the row set, r in Fig. 8. The expected number of columns not covered gives us an upper bound on the probability that a given set of rows fails to cover one or more columns. In the figure, the curve labeled "tolerable number of uncovered columns" is h minus the size of the row set.

For the values of d of most practical interest (≤ 16), the number of columns not covered by any random row set is much less than the tolerable number. For $d = 8$, the probability that a set of 140 or more rows fails to cover any column is less than one in a million. Another way to look at this is to note that while a nonblocking switch can implement all mappings of the input channels to output links, the blocking switch can implement all but a minuscule fraction of the set of possible mappings.

It is also possible to use eigenvalue methods to test if a given game board is k-solvable. This makes it possible to search for

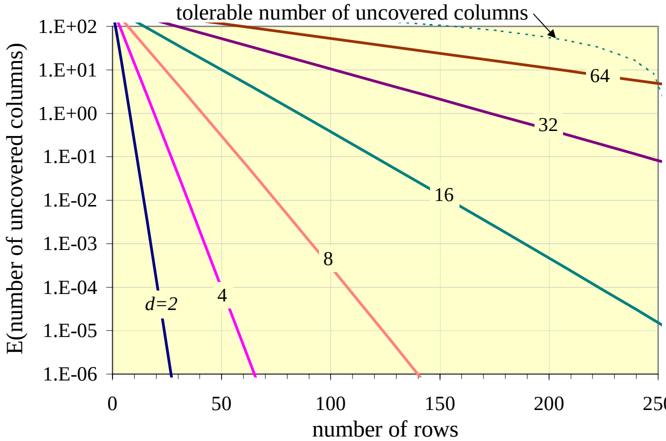


Fig. 8. Number of columns covered by row sets of all sizes ($h = 256$)

good game boards by generating random boards and testing them. We performed such a search of random *regular* game boards, where a regular game board is a balanced game board in which each block has the same pattern, with the exception that the role of the colors is shifted from one block to the next (so the squares colored c in one block are colored $c + 1 \bmod d$ in the next block). In a search of roughly $6,000$ random regular game boards, we found one board that is 171-solvable, one that is 170-solvable, one that is 169-solvable, and five that are 168-solvable.

VI. Routing Connections Online

The puzzle introduced above corresponds to the version of the routing problem in which we are asked to simultaneously route a whole set of connections. More often, we are interested in routing individual connections one-at-a-time, without disturbing connections routed previously. This problem can be formulated as a two player game, played on the same game board as the puzzle.

Let's call the first player the *blocker* and the second player, the *setter*. The blocker is given $k \leq h$ tokens of each of the d different colors. The blocker takes a turn by removing zero or more tokens from the board and placing one token beside some unoccupied row of the board. The setter takes its turn by placing the token put down by the blocker, in a square of the same color as the token in the selected row. When placing the token, the setter must not use any column that already contains a token of the same color. The blocker wins if the setter is not able to place the token on the board without violating the conditions. The blocker loses if the setter is able to keep the game going indefinitely.

The switch is *strictly nonblocking* if no matter how badly the setter plays, there is no way for the blocker to force a win. The switch is *wide-sense nonblocking* if there is a winning strategy for the setter (that is a strategy that will keep the game going forever, regardless of how well the blocker plays).

Since a winning strategy for the setter would imply that the corresponding puzzle always has a solution, we cannot expect a winning strategy in versions of the game where the blocker has more tokens than allowed by the upper bounds in Section V. It's easy to see that the setter has a trivial winning strategy when

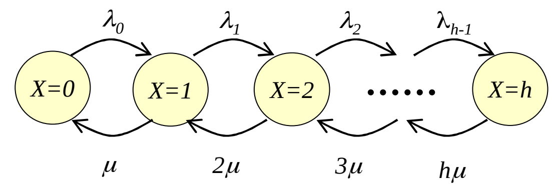


Fig. 9. Birth-death modeling an output of the switch

the number of tokens of each color is limited to $\leq h/d$. Hence, the switch is strictly nonblocking in these cases. It's also easy to see that the blocker can beat a naive setter if the blocker is allowed more than h/d tokens of each color. In other words, the switch is strictly nonblocking if and only if the blocker is limited to $\leq h/d$ tokens of each color.

We now present an approximate analytical model for evaluating the performance of wavelength converting switches in OBS routers. The performance metric used is the fraction of arriving bursts that must be discarded. This is called the *burst rejection probability*. We phrase the model in terms of the game board. Since the tokens of different colors are independent, we focus on tokens of a single color. New tokens arrive at rate λ, and if possible are placed on the game board. Tokens stay on the game board for an average time period of $1/\mu$. If the token interarrival time and the token "dwell time" are exponentially distributed, we can model the system by the birth-death process shown in Fig. 9, where the state index corresponds to the number of tokens on the game board. The transition rate from state i to state $i-1$ is $i\mu$, where $1/\mu$ is the expected time duration for which a token stays on the game board. The transition rate from state i to $i+1$ varies for different states since the probability that an arriving token is actually placed on the game board decreases as the number of tokens on the board increases. The rate, λ_i, is the rate at which tokens are placed on the board, which is equal to λ times the probability that an arriving token is successfully placed. So, for $i < h/d$, $\lambda_i = \lambda$. For $i \geq h/d$ and a random game board, we can approximate λ_i by

$$\lambda_i = \lambda \left(1 - \binom{h-h/d}{i-h/d} \Big/ \binom{h}{i}\right)$$

Note that $\binom{h}{i}$ is the number of sets of columns that can be used by i tokens and $\binom{h-h/d}{i-h/d}$ is the number of column sets used by i tokens that would prevent placement of a new token.

If we let π_i be the steady state probability that the system is in state i, then it can easily shown that

$$\pi_i = \frac{\lambda_0 \lambda_1 \ldots \lambda_{i-1}}{i!\, \mu^i} \pi_0$$

Using $\pi_0 + \pi_1 + \ldots + \pi_h = 1$ and solving for π_0, we can determine the individual steady state probabilities. The burst rejection probability is then given by

$$P_{rejection}(\rho) = \sum_{i=h/d}^{h} \pi_i \binom{h-h/d}{i-h/d} \Big/ \binom{h}{i}$$

where ρ is the offered load to the system given by $\lambda/h\mu$ and $\binom{h-h/d}{i-h/d}/\binom{h}{i}$ is the probability of a burst being rejected in state i.

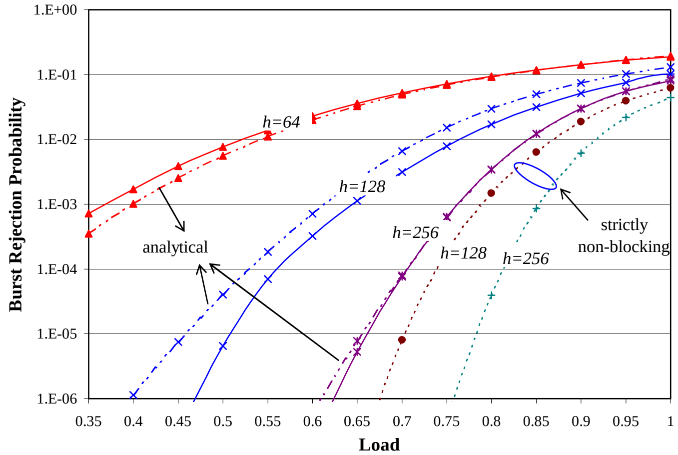


Fig. 10. Error probabilities of different system configurations ($d = 8$)

A. Simulation results for random game boards

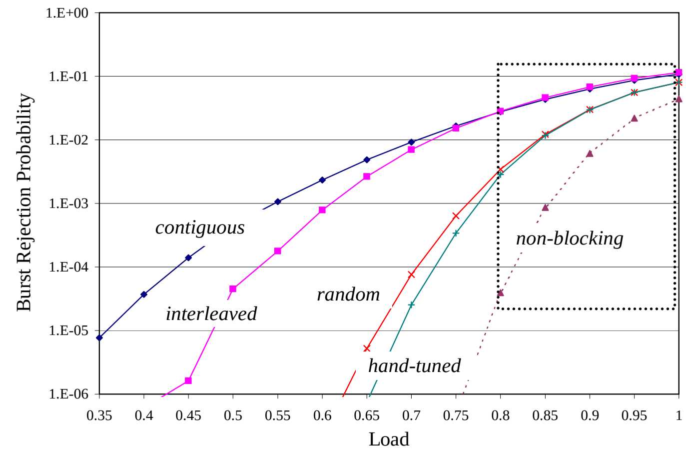
We now study how the blocking characteristics of the WGR-based switch affects the statistical multiplexing performance of an OBS router using simulation and compare the results with the analysis.

Here, we consider only the case of routers in which there are no buffers available to store bursts which can't be routed to the proper output without a wavelength conflict. Burst arrivals on each input channel are independent and each arriving burst is randomly assigned to a different output fiber. Burst lengths and the idle times between successive bursts on the same channel are exponentially distributed. The simulations used random regular permutation patterns at the input sections of the switch. Arriving bursts are assigned to the first wavelength that takes them to the proper output, that is not already in use at that output. In the game formulation, this corresponds to placing a token in the leftmost square of the right color, for which the column does not already contain a token of the same color.

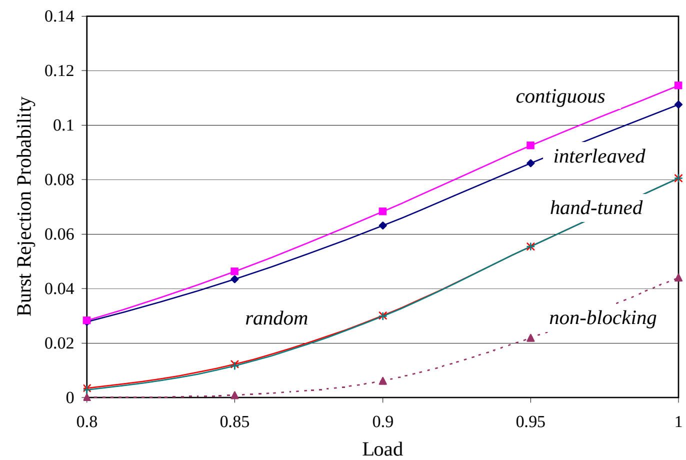
The burst rejection probabilities for systems with different values of d and h and varying loads are shown in Fig. 10. Also shown are the burst rejection probabilities for systems that use strictly nonblocking switches in place of WGR-based switches. For a system with $d = 8$ and $h = 256$, the rejection probability is 10^{-6} at a load of approximately 0.62 for the WGR-based switch and at a load of 0.75 for the strictly nonblocking switch. For systems designed to operate with a burst rejection ratio of 10^{-6}, the WGR-based switch can provide a throughput which is approximately 82% of what the nonblocking switch can provide. We also show the burst rejection probabilities as determined by the analysis of the burst switch.The analysis is accurate for higher values of h/d as can be seen in Fig. 10 and for $h = 256$, the analysis is very close to what is observed through simulation.

B. Effects of game board configurations

The next set of results show how different game board configurations can affect system performance. Fig. 11 shows the result of using four different configurations for a system with $d = 8$ and $h = 256$. The first configuration corresponds to a "contiguous" permutation pattern, wherein, each input block's first h/d outputs are connected to the first output, the next h/d outputs are connected to the second output, and so on. The second configuration corresponds to a perfect shuffle between the wavelength router stage and the input side couplers. The third configuration corresponds to a randomly generated game board. The fourth configuration is a hand generated configuration where colors corresponding to any output in a row are distributed such that two rows have very little overlap and hence, the number of wavelengths available to reach a given output between a set of rows is increased. As can be seen from Fig. 11, the first two configurations perform poorly. This is because they do not try to maximize the number of wavelengths available to a subset of input rows uniformly. The fourth configuration performs slightly better than the random pattern and has a burst rejection probability of 10^{-6} at a load of 0.65. With this design, the WGR-based switch achieves 87% of the throughput that is achieved with the strictly non-blocking switch.



(a)



(b)

Fig. 11. Error probabilities of different game boards ($d = 8$,$h = 256$)

C. Effect of reconfiguring connections in the switch

The simulations described above were done with the restriction that once assigned to a wavelength channel, a burst occupies that wavelength for its entire duration. We also simulated a situation in which bursts could be dynamically switched to different wavelengths to accommodate newly arriving bursts. Although this is not a realistic scenario, it allows us to separate the effects due to the intrinsic blocking character of the WGR-switches from those due to the restriction on rearrangements. The simulations were done using the rearrangement algorithm described in Section IV-C. The burst rejection probabilities in these simulations matched the strictly non-blocking results almost exactly for the given traffic pattern. This confirms the implications of Fig. 8. Even though the WGR-based switch is not rearrangeably nonblocking, it performs nearly as well as a rearrangeably nonblocking switch, when rearrangements are allowed. This suggests that a better wavelength assignment strategy (the strategy used by the *setter* to place tokens on the game board) may help reduce the burst rejection probabilities further, when rearrangement is not allowed.

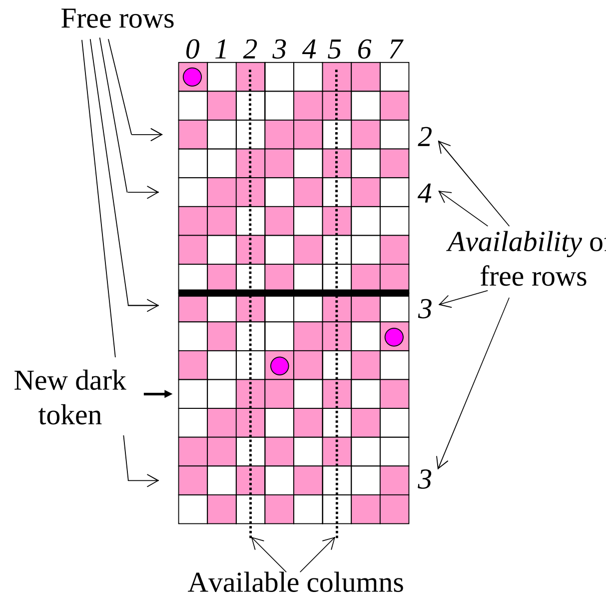


Fig. 12. Switch state for routing example

D. Effect of using most available wavelength assignment algorithm

We now present a routing algorithm that attempts to maximize the number of wavelengths available to free inputs to any output and study the effects of using the algorithm on the performance of the switch.

Define *availability*, a_{ij}, of a row i for color j to be the number of columns that have a j-colored square and are available in the row and we define these columns to be *available* columns. Equivalently, a_{ij} is the number of wavelengths at an input i that take us to output j and are not used by any other input.

Let us suppose a token of color m needs to be placed at row n. If $c_1, c_2, \ldots, c_k$ are the available columns in row n that have an m-colored square (That is, $a_{nm} = k$), we need to make a choice among one of the k available columns. Let $i_1, i_2, \ldots, i_s$ be the rows that are free currently in the game board and $a_{i_1m}, a_{i_2m}, \ldots, a_{i_sm}$ be their availabilities for color m. If we place the token in column c_l, the availability of rows that have c_l as an available column gets decremented by one. Let the new availabilities be $a^l_{i_1m}, a^l_{i_2m}, \ldots, a^l_{i_sm}$, where

$$a^l_{i_km} = \begin{cases} a_{i_km} - 1 & \text{if } c_l \text{ is an available at row } i_k \\ a_{i_km} & \text{otherwise} \end{cases}$$

Now we can define the *availability vector*, A_{c_l}, of a column, c_l, as the *ordered* vector of the resultant availabilities of the free rows if c_l is the column that is chosen for a new token in row n. That is

$$A_{c_l} = \text{Sort}(a^l_{i_1m}, a^l_{i_2m}, \ldots, a^l_{i_sm})$$

Among the columns c_l, $1 \leq l \leq k$, we choose the column that has the *maximum availability*, where the maximum availability is the lexicographic maximum of all the availability vectors. Specifically, if $A = \{a_1, a_2, \ldots, a_x\}$ and $B = \{b_1, b_2, \ldots, b_x\}$ are two availability vectors then

$$A > B \qquad \text{if } \exists i \leq x, \text{ such that } a_1 = b_1, \ldots, a_{i-1} = b_{i-1} \text{ and } a_i > b_i$$

By choosing the column with the maximum availability, we maximize the number of options available to the row with the minimum number of options for placing tokens of color m.

To illustrate the routing algorithm, let us assume that the state of a switch is as represented by the game board with $h = 8$ and $d = 2$ shown in Fig. 12. We consider connections to the output that has colored squares (as opposed to white squares) and let us assume that tokens to the output have already been placed as indicated in the figure. Denote rows as (i, j), where i represents the block and j represents the row within the block. Suppose we have to place a dark token in row $(2, 4)$, we have two columns, column 2 and column 5, that are available. Also suppose inputs $(1, 3)$, $(1, 5)$, $(2, 1)$, and $(2, 7)$ are currently free, then the availabilities of these inputs are 2, 4, 3, and 3 respectively. The rows that are affected when column 2 is used are $(1, 5)$, and $(2, 1)$ and the row that is affected when column 5 is used is $(2, 1)$. Thus the availability vectors for the two columns are given by

$$A_2 = \{2, 2, 2, 3\}, \text{ and } A_5 = \{2, 2, 3, 4\}$$

and we can see that $A_5 > A_2$. Thus, we choose column 5 to place the token in.

The simulation results of using the most available wavelength assignment algorithm for different permutation patterns are shown in Fig. 13. The algorithm gives tremendous improvement in throughput for the contiguous permutation pattern. It gives reasonable improvement for the random and the hand-tuned permutation patterns. Surprisingly, it does poorly for the interleaved permutation pattern. For the hand-tuned permutation, it improves the throughput to about 89% of the non-blocking case at a blocking probability of 10^{-6} with a utilization of 68%.

VII. CONCLUSION

We have studied the performance of wavelength converting switches using tunable wavelength converters and passive

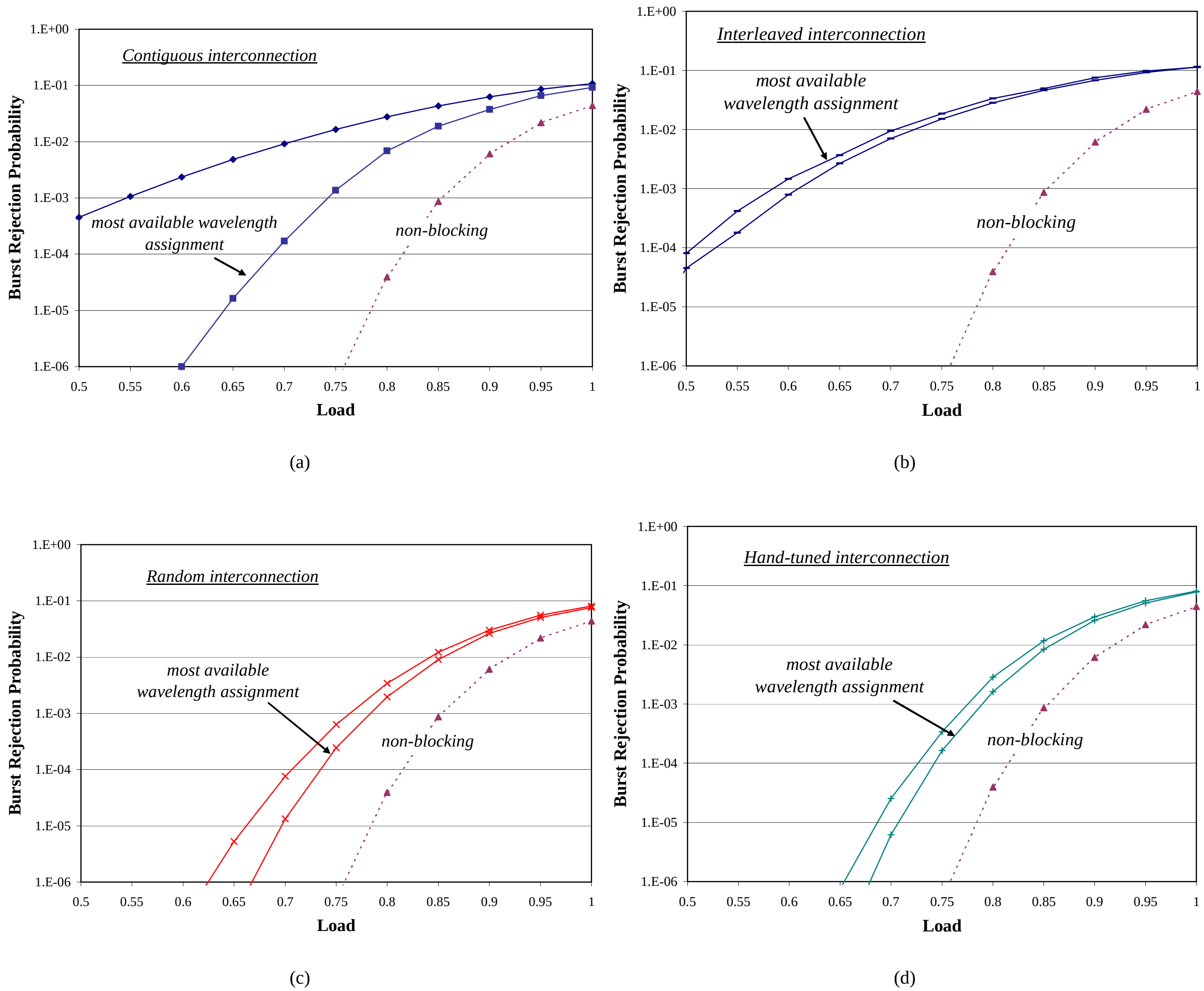


Fig. 13. Burst rejection probabilities of different game boards with the most available wavelength assignment ($d = 8$,$h = 256$)

wavelength grating routers. Such switches are attractive, because the only active element they require are the tunable wavelength converters. Although the blocking nature of these switches results in higher burst rejection probabilities, the performance penalty is small enough to make them a viable alternative.

By formulating the routing problem as a combinatorial puzzle or game, we have been able to develop insights that facilitate the analysis and design of WGR-based switches. We have shown that the problem of solving the puzzle can be reformulated as a matching problem in a bipartite graph. Also, an average case analysis shows us that we can almost always solve the puzzle. Further, we have shown some basic limits to the nonblocking potential of WGR-based switches and have also shown that by selecting the permutation patterns appropriately, one can greatly improve their performance. Simulation results show that in practical switch system configurations, routers using WGR-based switches can achieve more than 87% of the throughput of routers using strictly non-blocking switches.

Also, our simulations showed that if we allow bursts to be reassigned to different wavelength channels during transmission, the performance of the WGR-based switch matches the strictly non-blocking one almost exactly.

We then presented the *most available* wavelength assignment algorithm and studied its performance in the switch. We noticed that this algorithm improved the throughput of the switch using the contiguous permutation pattern. However, it did not yield significantly better performance for the randomized permutations. Using this algorithm, the throughput of WGR-based switches can achieve 89% of the throughput of routers using strictly non-blocking switches.

A design option that needs to be explored is the use of buffering in OBS routers. In general, buffering can be expected to improve the performance, and we expect it to have a larger impact on routers built using WGR-based switches, narrowing the performance gap further.

REFERENCES

[1] C. Qiao, and M. Yoo, “Optical Burst Switching (OBS) - A new paradigm for an optical internet,” *Journal of High Speed Networks*, vol. 8, pp. 69-84, 1999.

[2] C. Qiao, “Labelled optical burst switching for IP-over-WDM integration,” *IEEE Communications Magazine*, vol. 38, no. 9, pp. 104-114, Sep. 2000.

[3] J.S. Turner, “Terabit Burst Switching,” *Journal of High Speed Networks*, 1999.

[4] S. Verman, H. Chaskar, and R. Ravikanth, “Optical burst switching: A viable solution for terabit IP backbone,” *IEEE Network*, vol. 14, no. 6, pp. 48-53, Nov./Dec. 2000.

[5] J.Y. Wei, and R. McFarland, “Just-In-Time signalling for WDM optical burst switching networks,” *IEEE Journal of Lightwave Technology*, vol. 18, no. 12, pp. 2019-2037, Dec. 2000.

[6] Y. Xiong, M. Vandenhoute, and H. Cankaya, “Control architecture in optical burst-switched WDM networks,” *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 10, pp. 1838-1851, Oct. 2000.

[7] D. Blumenthal, P. Prucnal, J. Sauer, “Photonic packet switches: Architectures and experimental implementation,” Proc. IEEE, vol. 82, pp. 1650-1667, Nov. 1994.

[8] P. Gambini, *et al*., “Transparent optical packet switching: Network architecture and demonstrators in the KEOPS project,” *IEEE Journal on Selected Areas in Communication*, vol. 16, pp. 1245-1259, Sep. 1998.

[9] D.K. Hunter, *et. al.*, “WASPNET: A wavelength switched packet network,” *IEEE Communications Magazine*, pp. 120-129, Mar. 1999.

[10] F. Masetti, “System functionalities and architectures in photonic packet switching,” *Photonic Networks*, Giancarlo Prati (editor), Springer Verlag 1997.

[11] L. Xu, H. Perros, and G. Rouskas, “Techniques for optical packet switching and optical burst switching,” *IEEE Communications Magazine*, pp. 136-142, Jan. 2001.

[12] S. Yao, S. Dixit, and B. Mukherjee, “Advances in photonic packet switching: An overview,” *IEEE Communications Magazine*, pp. 84-94, Feb. 2000.

[13] W.D. Zhong, and R.S. Tucker, “Wavelength routing-based photonic packet buffers and their applications in photonic packet switching systems,” *Journal of Lightwave Technology*, vol. 16, no. 10, pp. 1737-1745, Oct. 1998.

[14] T. Durhuus, B. Mikkelsen, C. Joergensen, S.L. Danielsen, K.E. Stubkjaer, “All-optical wavelength conversion by semiconductor optical amplifiers,” *Journal of Lightwave Technology*, vol. 14, pp. 942-954, Jun. 1996.

[15] S.J.B. Yoo, “Wavelength conversion technologies for WDM network applications,” *Journal of Lightwave Technology*, vol. 14, pp. 955-956, Jun. 1996.

[16] I. White, R. Penty, M. Webster, Y.J. Chai, A. Wonfor, and S. Shahkooh, “Wavelength switching components for future photonic networks,” *IEEE Communications Magazine*, vol. 40, no. 9, pp. 74-81, Sep. 2002.

[17] Dragone, C., “An NxN optical multiplexor using a planar arrangement of two star couplers,” *IEEE Photonic Technology Letters*, vol.6, pp. 812-815, Sep. 1991.