An Analysis of Blocking Switches Using Error Control Codes

Can Emre Koksal, *Member, IEEE*

***Abstract*—We study the relationship between the degree of blocking and the amount of resource speedup necessary for blocking switches to possess the capabilities of nonblocking switches. We construct an analogy between switch configurations and the codewords of certain error control codes for which we use space covering ideas to derive relations between speedup and number of switch configurations. To derive the necessary speedup for nonblocking, we use two sphere packing bounds: the Hamming bound and the Gilbert–Varshamov bound. To construct nonblocking switches with a given speedup we use maximum distance separable codes. We consider both multicast and point to point scenarios.**

***Index Terms*—Coding theory, covering codes, error control codes, maximum distance separable codes, multicast, nonblocking switches, switching theory, unicast.**

I. Introduction and Motivation

At the core of every switch lies a switching interconnection fabric. The function of the interconnection fabric, or simply the interconnection, is to set up connections between the input and output units of the switch. A switch can be classified as blocking or nonblocking depending on whether a connection can always be made between any free input unit and free output unit through its interconnection.

In general, blocking in a switch is defined as the failure to satisfy a connection requirement because all paths for that connection conflict with paths inside the interconnection already existing for connections between other input–output (I-O) pairs: An $N \times N$ switch is said to be blocking if there exists a set of distinct inputs $\{i_1, \ldots, i_N\}$ and a set of distinct outputs $\{j_1, \ldots, j_N\}$ such that connections $(i_1, j_1), \ldots, (i_N, j_N)$ cannot be made simultaneously. In other words, if the *configuration* $\{(i_1, j_1), \ldots, (i_N, j_N)\}$ is not feasible.

Nonblocking switches are appealing in that they can provide better quality of service with simpler scheduling algorithms relative to blocking switches. However, they may become increasingly complex as the size of the switches grow large. For instance, the crosspoint complexity of a crossbar[1] is $O(N^2)$, and it gets harder to manufacture the crossbar for larger switches and high speed applications. There are many architectures such as the ring, the bus or some multistage switches (e.g., Banyan) that are blocking. In many cases, some *resource speedup* is introduced to achieve the nonblocking property in such switches. Speedup enables multiple connections to be made between I-O pairs simultaneously. For instance, in an optical ring network, more than a single wavelength is used so that multiple nodes can communicate simultaneously. Also, note that a blocking switch architecture may support speedup easily due to its simplicity. Thus, it may be preferable to build a nonblocking switch using a blocking switch along with some speedup.

Manuscript received November 2, 2004; revised October 19, 2005. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Chicago, IL, June 2004.

The author is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: koksal@ece.osu.edu).

Communicated by G. Sasaki, Associate Editor for Communication Networks.

Digital Object Identifier 10.1109/TIT.2007.901191

[1]For a detailed treatment, see [1].

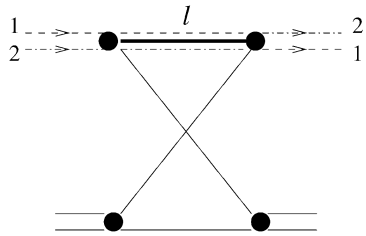

Fig. 1. Link l must be shared by two complete paths for the connections $(1, 2)$ and $(2, 1)$ to be made simultaneously.

While the problem of providing guaranteed quality of service has been explored for nonblocking switches extensively (e.g., [2], [3]), it has not been considered for blocking switches in general. In this correspondence, we study the relation between the degree of blocking and the amount of resource speedup necessary for such switches to possess the capabilities of nonblocking switches. We consider both point to point and multicast scenarios.

More precisely, we call a switch $\mathcal{N}$ to be "nonblocking with speedup S" if the switch can be operated in a nonblocking fashion if S identical parallel copies were available. In this correspondence, we find a lower bound to the size of the set of configurations of a switch $\mathcal{N}$ for it to be nonblocking with speedup S. To derive this bound, we use space covering properties of error control codes. We also give a method to construct configuration sets with sizes close to the lower bound. This construction method relies heavily on the existence of codes with certain properties.

The main contribution of this correspondence is combining switching theory and coding theory, which we believe has not been addressed before. Next, we give examples where our analyzes are relevant, but the main purpose of this correspondence is not to analyze any switching system per se nor to design one. Rather, we focus on giving a different perspective in switching theory. We believe this perspective may be fruitful in not only understanding some fundamental properties of switches, but also providing ways to come up with new architectures and systems involving blocking switches.

Example 1: **Banyan Switch:** In the Banyan switch (see [1] for an extensive treatment) there is only one path between any I-O pair. The Banyan switch is blocking. For instance, consider the 4×4 Banyan switch composed of 2×2 crossbars given in Fig. 1. Given that each link inside the network can handle one connection (one I-O pair), the two connections between I-O pairs $(1, 2)$ and $(2, 1)$ cannot be made simultaneously since the path between the first pair and the path between the second pair share the same link 1. With a closer examination, one can see that if each link can carry two connections simultaneously and if each 2×2 crossbar can support two configurations at a given time, this switch becomes nonblocking. Hence, we say the switch given in Fig. 1 can be made nonblocking with a speedup 2.

Example 2: **Waveguide Grating Router (WGR):** A WGR is an optical switch. It takes in input signals and distributes them according to the wavelength of the signal and the state of another independent controller (see [4] for an extensive treatment). The grating is built such that if a signal at a certain wavelength is inserted in all the input ports, it will be distributed over different output ports as illustrated in Fig. 2. If control signal (or the operating conditions of the grating) is changed, routes of the same set of signals shift cyclically as shown in the same figure. Since, under the same control signal there exists only one route that a signal follows, the switch blocks any request for a different configuration. For the single wavelength scenario, a WGR can support the

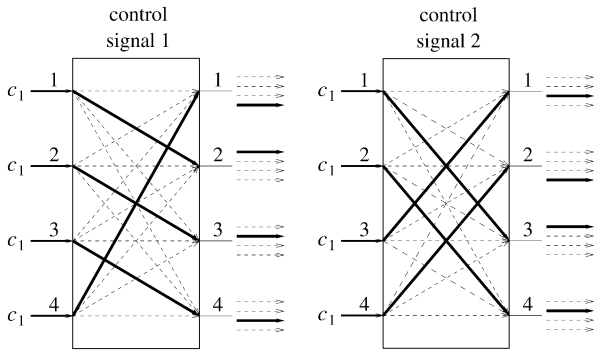

Fig. 2. Signal at a wavelength, c_1 is switched as shown in the first part. If control signal is changed, the route of the same set of signals change as shown in the second part.

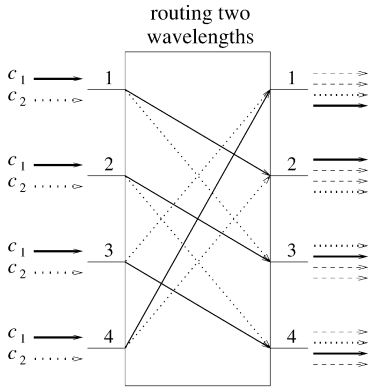

Fig. 3. The signal with c_2 is switched to a different output fiber than the one with c_1. Thus each new wavelength introduces an extra degree of freedom.

trivial configuration ($\{(1,1),(2,2),\ldots,(N,N)\}$) and all the cyclic shifts of it.

To make this switch nonblocking, resource speedup is necessary and it can be introduced by increasing the number of wavelengths. The impact of a second wavelength is illustrated in Fig. 3. The signal at wavelength c_2 is switched to a different output fiber than the one at wavelength c_1. Thus each new wavelength introduces an extra degree of freedom which scales the number of configurations by a factor N. For nonblocking, a speedup (the number of wavelengths) of N is necessary where N is the number input links. A routing table specifies the wavelength that is used to setup connection between an input and an output fiber. The coordinate of the element in this matrix represents the input–output pair. Following is a routing table for the 4×4 router.

$\downarrow$ **I-O** $\rightarrow$	**1**	**2**	**3**	**4**
1	c_1	c_2	c_3	c_4
2	c_4	c_1	c_2	c_3
3	c_3	c_4	c_1	c_2
4	c_2	c_3	c_4	c_1

Example 3: The ring network is one of the most popular architectures in optical networks. Today, most of the physical layer infrastructure is built around rings. A ring network is illustrated in Fig. 4.

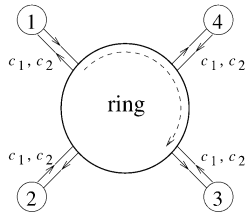

Fig. 4. A four user unidirectional optical ring which supports two wavelengths. Each node can add and drop the same pair.

If a single wavelength is used in a unidirectional ring, only one node can achieve full duplex communication[2] with another node at a time. Using wavelength multiplexers enable multiple nodes to communicate simultaneously. In a sense, multiple rings can be supported over the same infrastructure, which can be viewed as introducing speedup.

Let us define a symmetric configuration, λ, as the one for which if $(i,j) \in \lambda$ then $(j,i) \in \lambda$. The system illustrated in Fig. 4 can support only symmetric configurations, i.e., it is blocking. If we add two more wavelengths on the already existing two, all possible configurations can be supported. Hence we say the system can be made nonblocking with a speedup 2.

The rest of this correspondence is organized as follows. In Section II, we give the problem statement. Next, we give the necessary and sufficient conditions for a blocking switch to mimic a nonblocking switch in Sections III and IV respectively. In Section V, we compare the necessary and sufficient conditions and discuss some of the results further. We finalize the correspondence with conclusions and some directions for future research.

II. Problem Statement

In this section, we give some definitions and present a mathematical description of the problem.

A. Definitions

A configuration is a set, λ, which is composed of I-O pairs, (v_i, v_o). We assume that the number, N, of inputs and outputs of a switch is identical, and the cardinality of any configuration, $|\lambda| = N$. We also assume that exactly one connection terminates at each $v_o \in \{1,\ldots,N\}$, i.e., each one of the N pairs in λ will have a distinct output. On the other hand, an input can be found in more than one of these pairs depending on whether the switch has the multicast capability. Without the multicast capability, exactly one connection initiates at each $v_i \in \{1,\ldots,N\}$. We call a switch *multicast* or *unicast* accordingly.

A switch, $\mathcal{N}$, can be represented with a set, $\Lambda_{\mathcal{N}}$, of *feasible* configurations. A configuration is *feasible* if all the connections in the configuration can be made simultaneously. We will represent each configuration with the sequence of N inputs that the outputs are connected to. For instance, if $\lambda = \{(1,1),(3,2),(1,3),(4,4)\}$, then we simply use $\lambda = (1314)$.

Let the set of all possible configurations be Γ. Hence, for any switch $\mathcal{N}$, $\Lambda_{\mathcal{N}} \subset \Gamma$. For multicast switches, $\Gamma = \{1,\ldots,N\}^N$. Switch $\mathcal{N}$ is called nonblocking if all possible configurations are feasible, i.e., $\Lambda_{\mathcal{N}} \equiv \Gamma$. Thus, for switch $\mathcal{N}$ to be nonblocking, $|\Lambda_{\mathcal{N}}| = |\Gamma| = N^N$ feasible configurations is necessary. For a unicast switch, Γ is identical to the set of all permutations that can be made with the elements of $\{1,\ldots,N\}$. A unicast switch is blocking if the number of its feasible configurations is less than $N!$.

[2]We assume only full wavelength communication is possible between nodes. If *add-drop multiplexers* were used, each node would be capable of inserting and taking out fractions of wavelengths, which would relax the full wavelength communication assumption.

Switch $\mathcal{N}$ is said to have a *speedup* $S \in \mathbb{Z}^+$ if it can support S distinct feasible configurations simultaneously. More precisely, with a speedup S any $\lambda_1, \ldots, \lambda_S$ such that $\lambda_i \in \Lambda_{\mathcal{N}}, 1 \leqslant i \leqslant S$ can be in effect simultaneously. Thus, with a speedup S, switch $\mathcal{N}$ is capable of providing all configurations that another switch $\mathcal{N}'$ can, where $\Lambda_{\mathcal{N}'}$ is composed of elements that are subsets of $\bigcup_{i=1}^{S} \lambda_i$ of cardinality N for every possible $\{\lambda_1, \ldots, \lambda_S\} \subset \Lambda_{\mathcal{N}}$. If $\Lambda_{\mathcal{N}'} = \Gamma$, we say switch $\mathcal{N}$ can be made nonblocking with a speedup S, or a speedup S is sufficient for $\mathcal{N}$ to be nonblocking.

B. Problem Statement

In this correspondence, we study the relation between the speedup and the nonblocking behavior of switches. We focus on two problems. Before we present these problems, consider the following example, which illustrates concepts of the two problems.

For a 4×4 switch, $\mathcal{N}$, suppose the configuration $\lambda_1 = (1234) \notin \Lambda_{\mathcal{N}}$, but $\lambda_2 = (1243)$ and $\lambda_3 = (2134) \in \Lambda_{\mathcal{N}}$. Then, λ_2 and λ_3 can cover for λ_1 if they can be in effect simultaneously since I–O pairs $(1,1)$ and $(2,2)$ can be connected through λ_2, and I–O pairs $(3,3)$ and $(4,4)$ can be connected through λ_3. Despite the absence of λ_1, switch $\mathcal{N}$ may still be nonblocking with the speedup 2. Conversely, with the speedup 2 and given λ_2 and λ_3 are feasible, λ_1 need not be feasible for the switch to be nonblocking.

The two problems we consider are the following.

1) For an $N \times N$ switch, $\mathcal{N}$, we derive relations between S and $|\Lambda_{\mathcal{N}}|$. First we find a lower bound, $L(N,S)$, on $|\Lambda_{\mathcal{N}}|$ for the $N \times N$ switch $\mathcal{N}$ to be nonblocking with a speedup S. In other words, no matter how the set $\Lambda_{\mathcal{N}}$ is constructed, if $|\Lambda_{\mathcal{N}}| < L(N,S)$ then there exists a $\gamma \in \Gamma$ such that for any set of S feasible configurations $\{\lambda_1, \ldots, \lambda_S\}$,
$$\gamma \not\subset \bigcup_{i=1}^{S} \lambda_i.$$
Hence if $|\Lambda_{\mathcal{N}}| < L(N,S)$, the speedup S is not sufficient for nonblocking.
Conversely, we find a lower bound on the speedup, below which a switch $\mathcal{N}$ cannot be made nonblocking for some given $|\Lambda_{\mathcal{N}}|$. We derive these results for both multicast and unicast switches.
2) We present a way to construct the set of configurations $\Lambda_{\mathcal{N}}(S)$, for which a switch $\mathcal{N}$ becomes nonblocking with the given speedup S. Indeed, the speedup S will be sufficient to make the $N \times N$ switch $\mathcal{N}$ nonblocking with the set of feasible configurations $\Lambda_{\mathcal{N}} = \Omega(N,S)$ found by this method. We show that this method is *exponentially efficient* for large N: For a given speedup S,
$$\lim_{N \to \infty} \frac{\log |\Omega(N,S)|}{\log L(N,S)} = 1. \tag{1}$$

III. Number of Configurations Versus Speedup for Nonblocking Switches

In this section, we state theorems for the necessary speedup for nonblocking multicast and unicast switches for a given number of configurations. We also give the inverse relations, i.e., the necessary number of configurations for nonblocking for a given speedup.

A. Multicast Switches

Theorem 1: If for an $N \times N$ multicast switch $\mathcal{N}$
$$|\Lambda_{\mathcal{N}}| < \frac{N^k}{\binom{N}{k}}$$
for some $k \in \{2, \ldots, N\}$, the speedup $S = \lceil \frac{N}{k-1} \rceil$ is necessary for $\mathcal{N}$ to be nonblocking.

The proof is geometric and it uses the Gilbert–Varshamov bound for codes. The complete proof can be found in the Appendix, but we sketch the main idea here. Suppose there exists a $\gamma \in \Gamma$ such that each $\lambda \in \Lambda_{\mathcal{N}}$ has less than $k, 1 < k \leqslant N$ entries in common with γ. Then, we cannot find $\lfloor N/k \rfloor$ configurations, $\lambda_1, \ldots, \lambda_{\lfloor N/k \rfloor} \in \Lambda_{\mathcal{N}}$ such that
$$\gamma \subset \bigcup_{l=1}^{\lfloor N/k \rfloor} \lambda_l.$$
Therefore, the speedup $S = \lfloor N/k \rfloor$ is not sufficient to make the switch $\mathcal{N}$ nonblocking. In the Appendix, we show the existence of such a $\gamma \in \Gamma$, if $|\Lambda_{\mathcal{N}}|$ is not large enough.

Corollary 1: An $N \times N$ multicast switch $\mathcal{N}$ cannot be nonblocking with a speedup $S \in \{1, \ldots, N\}$ if
$$|\Lambda_{\mathcal{N}}| < \frac{N^{\lceil N/S \rceil}}{\binom{N}{\lceil N/S \rceil}}. \tag{2}$$

Theorem 1 gives the necessary speedup for nonblocking multicast switches for a given number of configurations, whereas Corollary 1 derives a lower bound for the necessary number, $L(N,S)$ of configurations for nonblocking multicast switches for a given speedup. The proof of the corollary is immediate from Theorem 1 as given in the Appendix. Next, we derive similar bounds for unicast switches.

B. Unicast Switches

Theorem 2: If for an $N \times N$ unicast switch $\mathcal{N}^*$
$$|\Lambda_{\mathcal{N}^*}| < k!$$
for some $k \in \{2, \ldots, N\rfloor\}$, then the speedup $S = \lceil \frac{N}{k-1} \rceil$ is necessary for $\mathcal{N}^*$ to be nonblocking.

The proof can be found in the Appendix. The approach is very similar to the one with multicast switches.

Corollary 2: An $N \times N$ unicast switch, $\mathcal{N}^*$ cannot be nonblocking with a speedup $S \in \{1, \ldots, N-1\}$ if
$$|\Lambda_{\mathcal{N}^*}| < \lceil N/S \rceil!\cdot \tag{3}$$

Theorem 2 gives the necessary speedup for nonblocking unicast switches for a given number of configurations, whereas Corollary 2 gives a lower bound for the necessary number of configurations for nonblocking unicast switches for a given speedup. The proof of the corollary is immediate from Theorem 1 as given in the Appendix.

IV. Constructing a Set of Configurations for Nonblocking Switches

In this section, we present a method to construct a set of feasible configurations for a switch in an exponentially efficient manner. We study the space covering properties of certain *maximum distance separable* (MDS) codes, and illustrate that codewords of linear MDS codes cover the space in such a way that, if the elements of $\Lambda_{\mathcal{N}}$ for the switch $\mathcal{N}$ are chosen to be the codewords of a certain linear MDS code, the necessary speedup for nonblocking is minimized. In the Appendix, we give some basic definitions on MDS codes necessary for the development. For a detailed treatment of MDS codes and algebraic coding theory in general, see [5].

A. Multicast Switches

Let us consider MDS codes over $\eta = \{1, \ldots, N\}$. Every subset of $k = N-d+1$ coordinates of the MDS code (N, k, d) is an information set. Thus, for any set of k coordinates, the codewords run through all N^k possible k-tuples. Hence, there exists no two codewords that have the same set of k symbols in any given k coordinates. Note that, for a linear code, $\mathcal{C} : (N, k, d)$, any set of k-coordinates is an information set if and only if $d = N - k + 1$, which is true only for MDS codes. The following theorem illustrates why this property is important in our construction of configuration sets.

Theorem 3: If $\Lambda_{\mathcal{N}} : (N, k, N - k + 1)$, then the speedup $S = \lceil N/k \rceil$ is sufficient to make the multicast switch $\mathcal{N}$ nonblocking.

The proof can be found in the Appendix. Theorem 3 not only gives a way to construct the configuration set for a blocking switch, but also shows that, given a speedup S, $N^{\lceil N/S \rceil}$ configurations are sufficient to make a switch nonblocking.

Corollary 3: A speedup $S \in \{1, \ldots, N\}$ is sufficient to make an N × N multicast switch $\mathcal{N}$ nonblocking, if $\Lambda_{\mathcal{N}} = \Omega(N, S) = (N, \lceil N/S \rceil, N - \lceil N/S \rceil + 1)$.

The proof is immediate from Theorem 3. Corollary 3 shows that, given a speedup S, a carefully constructed set of

$$|\Lambda_{\mathcal{N}}| \geq N^{\lceil N/S \rceil}. \tag{4}$$

configurations is sufficient for nonblocking multicast switches.

B. Unicast Switches

An $(N, k, N-k+1)$ Reed Solomon (RS) code uses polynomials of degree less than k to generate the codewords. Such polynomials have at most $k-1$ roots, i.e., any element of $\{1, \ldots, N\}$ can be repeated in at most $k-1$ coordinates. Thus, the configurations generated by this method have connections all of which have a maximum fanout of $k-1$. Therefore, the only linear code we can use for the construction of the set of configurations is the $(N, 2, N-1)$ RS code, since $k > 2$ leads to configurations with fanouts greater than 1 for some connections.

The $(N, 2, N-1)$ RS code over the alphabet[3] $\{1, \ldots, N\}$ is composed of N^2 codewords, N of which have all identical coordinates, i.e., from $(1 \cdots 1)$ to $(N \cdots N)$. If we get rid of these N codewords, we end up with $\binom{N}{2}$ codewords in the desired form.

Unfortunately, this construction does not extend to any other speedup value and currently we do not have any method to construct efficient unicast switches. One idea is to get rid of all codewords of the $(N, k, N-k+1)$ code except the permutations of $\{1, \ldots, N\}$. However, this method eliminates too many configurations and for $kgt2$ we end up with a number of configurations less than the necessary number for nonblocking with the speedup $S = \lceil \frac{N}{k-1} \rceil$.

V. Summary

We showed that, for a given speedup S, $L(S) = N^{\lceil N/S \rceil} / \binom{N}{\lceil N/S \rceil}$ configurations is necessary for a multicast switch $\mathcal{N}$ to be nonblocking. Moreover, choosing $\Lambda_{\mathcal{N}}$ to be the linear MDS code $(N, k, N-k+1)$ is exponentially efficient for multicast switches. Indeed, for any given S

$$\lim_{N\to\infty} \frac{\log |\Omega(N,S)|}{\log L(N,S)} = \lim_{N\to\infty} \frac{\log N^{\lceil N/S \rceil}}{\log \frac{N^{\lceil N/S \rceil}}{\binom{N}{\lceil N/S \rceil}}} = 1,$$

since for large N

$$\binom{N}{N/S} \approx \frac{S^{N/S}}{\sqrt{2\pi \frac{N}{S}} \left(1 - \frac{1}{S}\right)^{(S-1)N/S}} \tag{5}$$

where (5) is shown in [6].

The dominant terms of the number of configurations necessary for nonblocking multicast and nonblocking unicast switches are $N^{\lceil N/S \rceil}$ and $\lceil N/S \rceil!$, respectively. These terms are sketched as a function of S in Fig. 5 for a 50×50 switch.

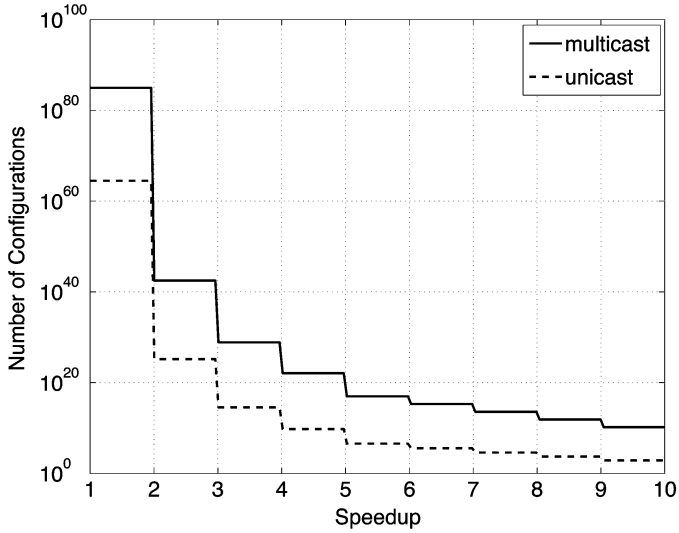

Fig. 5. The number of configurations for nonblocking a multicast and a unicast switch are illustrated as a function of the speedup for $N = 50$.

VI. Conclusion and Future Extensions

In this correspondence, we illustrate a connection between switching theory and coding theory. Using some space covering results in coding theory, we derived a lower bound for the size of the set of configurations for a switch for it to be nonblocking with a speedup S. We also showed that the lower bound can be approached using maximum distance separable codes to construct the switch configurations.

In nonblocking switches, speedup is undesirable most of the time since it is not necessarily feasible to run complex switches faster than the line rates. First, suppose we have a blocking switch that supports the $N^{\lceil N/S \rceil}$ configurations for some $S > 1$ as constructed in Section IV. We just illustrated that the complexity of this blocking switch (in the number of configurations) is significantly smaller than that of a nonblocking switch. The important question is, does it get easier to run the switch faster than the line rate (rate of the links connected to the switch) because it is less complex? For instance, in Fig. 5, the speedup 2 reduces the necessary number of configurations by a factor of $\sim 10^{30}$. If this decrease makes it easier to run the switch twice as fast as the line rates, then it may be preferable to use a blocking switch with some speedup (so that it becomes nonblocking) rather than a nonblocking switch without a speedup. The answer to this question depends on whether it is possible to manufacture such switches and how simple that process is.

Now let us briefly discuss how such switches can be implemented. We believe that there may exist systematic ways to design complex (in terms of the number of configurations supported) switches using very simple ones. An example of a simple switch is the one that supports N configurations (e.g., one that supports trivial configuration and all cyclic shifts of it). Such a construction can be of practical interest especially with optical devices. An appropriately chosen set of such

[3]Linear codes are defined over the alphabet that includes 0. For the sake of complete analogy we use $\{1, \ldots, N\}$ instead of $\{0, \ldots, N-1\}$ as the alphabet without loss of generality.

simple switches can be put in cascade so that the number of end to end configurations is equal to the product of the number of configurations of individual switches. An example of such a construction is the Banyan network, which was illustrated in Fig. 1. A possible way of achieving this is breaking the degree $k-1$ generating polynomial of the $(N, k, N-k+1)$ RS code into pieces of smaller degrees using certain factorizations so that each piece can be used as a generating polynomial for one of the switches. The described construction may have a significant advantage in terms of simplicity over single stage nonblocking switches such as the crossbar. Also, such a construction provides the possibility of putting buffering in between stages which can be very appealing in providing quality of service over packet switches. In [7], an algebra for multistage switches is developed, and a method for finding the set of configurations for cascaded switches is shown.

One extension of this work is to find a tighter lower bound for the necessary number of configurations. Even though the current bound is tight in exponent, it may be possible that there exists a bound for which the the ratio of the necessary to sufficient number of configurations for nonblocking converge to a constant. Indeed it may be conjectured that the $(N, k, N-k+1)$ code consists of the necessary number of codewords for nonblocking multicast switches with the speedup $\lceil N/k \rceil$.

Finally, we note that there is also a connection between the construction of the configuration set for unicast switches and the construction of orthogonal Latin squares. In [8], how to use $(N, 2, N-1)$ Reed–Solomon codes to construct orthogonal Latin squares is explained. We believe the connection between Latin squares and MDS codes can be exploited to get further insights into constructing switches.

Appendix
Proof of Theorem 1

Before the proof, we give some definitions. Recall that a configuration is a set of I–O pairs, each of which consists of a distinct output. We can represent each configuration as the list of inputs that the outputs are connected to. Hence, a configuration is an N-tuple whose elements are picked from a finite alphabet $\eta = \{1, \ldots, N\}$, and $\Gamma = \eta^N$ (see [9] for a complete treatment on finite fields).

Definition 1: The Hamming distance, $d_H(\gamma, \gamma')$ between two elements, $\gamma, \gamma' \in \Gamma$ is the number of entries they differ.

Definition 2: The N-ary ball, $B_N(\gamma, r)$ with center $\gamma \in \Gamma$ and radius r consists of all $\gamma' \in \Gamma$ such that $d_H(\gamma, \gamma') \leqslant r$. Note that

$$|B_N(\gamma, r)| \leqslant \binom{N}{r} N^r. \tag{6}$$

The right-hand side of (6) corresponds to all vectors that are created by choosing r elements of γ and then for each of the r elements, choosing one of N possible values. This includes all vectors that have Hamming distance at most r from γ.

Lemma 1: If for some $k, 1 < k \leqslant N, |\Lambda_{\mathcal{N}}| < N^k / \binom{N}{k}$, then there exists a $\gamma \in \Gamma$ such that $d_H(\gamma, \lambda) > N-k$ for all $\lambda \in \Lambda_{\mathcal{N}}$. Thus, all $\lambda \in \Lambda_{\mathcal{N}}$ has less than k entries in common with γ.

Proof: According to this lemma, if $|\Lambda_{\mathcal{N}}|$ is not sufficiently large then, for some $\gamma \in \Gamma$

$$\Lambda_{\mathcal{N}} \cap B_N(\gamma, N-k) = \emptyset.$$

Conversely, no matter how $\Lambda_{\mathcal{N}}$ is constructed, there exists a $\gamma \in \Gamma$ such that

$$\gamma \notin \bigcup_{\lambda \in \Lambda_{\mathcal{N}}} B_N(\lambda, N-k).$$

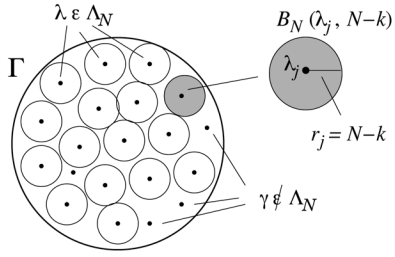

Fig. 6. The union of $B_N(\lambda, N-k)$ for all $\lambda \in \Lambda_{\mathcal{N}}$ is not large enough to cover all the points in Γ.

The geometry of the proof is illustrated in Fig. 6. We will show that the union of $B_N(\lambda, N-k)$ for all $\lambda \in \Lambda_{\mathcal{N}}$ is not large enough to cover all the points in Γ, if the number of points in $\Lambda_{\mathcal{N}}$ is not sufficiently large.

Our proof is by contradiction. Suppose for a given set of configurations $\Lambda_{\mathcal{N}}$ and a given k, $1 < k \leqslant N$, $|\Lambda_{\mathcal{N}}| < N^k / \binom{N}{k}$, and yet $\bigcup_{\lambda \in \Lambda_{\mathcal{N}}} B_N(\lambda, N-k) \equiv \Gamma$. Then

$$\begin{aligned} N^N &= \left| \bigcup_{\lambda \in \Lambda_{\mathcal{N}}} B_N(\lambda, N-k) \right| \\ &\leqslant \sum_{\lambda \in \Lambda_{\mathcal{N}}} |B_N(\lambda, N-k)| \qquad (7) \\ &\leqslant |\Lambda_{\mathcal{N}}| \cdot \binom{N}{N-k} N^{N-k} \qquad (8) \end{aligned}$$

where (7) follows from the union bound, and (8) follows from (6). Consequently

$$\begin{aligned} |\Lambda_{\mathcal{N}}| &\geqslant \left[\binom{N}{k} N^{N-k} \right]^{-1} N^N \\ &= \frac{N^k}{\binom{N}{k}} \end{aligned} \tag{9}$$

and the proof is complete.

The proof of Theorem 1 is almost immediate from the above derivation. Lemma 1 implies that there is a $\gamma \in \Gamma$ with which all $\lambda \in \Lambda_N$ have at most $k-1$ elements in common, if $|\Lambda_N| < N^k / \binom{N}{k}$. So if the speedup S satisfies $S(k-1) < N$, then the switch is blocking. Hence, $S = \lceil \frac{N}{k-1} \rceil$ is necessary (but not necessarily sufficient) for nonblocking.

Also for Corollary 1, we can rewrite the above inequality to have $k < N/S + 1$ and with this we conclude that a speedup S is not sufficient to make a switch nonblocking if the number of configurations

$$|\Lambda_{\mathcal{N}}| < \frac{N^{\lceil N/S \rceil}}{\binom{N}{\lceil N/S \rceil}}. \tag{10}$$

Proof of Theorem 2

Definition 3: Let Γ^* be the set of permutations that could be made using the elements of $\eta = \{1, \ldots, N\}$. The N-ary ball, $B_N^*(\gamma, r)$ with center $\gamma \in \Gamma^*$ and radius r consists of all $\gamma' \in \Gamma^*$ such that $d_H(\gamma, \gamma') \leqslant r$. Note that

$$|B_N^*(\gamma, r)| \leqslant \binom{N}{r} r!. \tag{11}$$

The right hand side of (11) corresponds to all vectors that are created by choosing r elements of γ and then permuting them. This includes all permutations of γ that have Hamming distance at most r from γ.

Next, we state the version of Lemma 1 for unicast switches.

Lemma 2: If $|\Lambda_{\mathcal{N}*}| < k!$, then there exists a $\gamma \in \Gamma^*$ for which $d_H(\gamma, \lambda) > N - k$ for all $\lambda \in \Lambda_{\mathcal{N}*}$.

Proof: We will show that no matter how $\Lambda_{\mathcal{N}*}$ is constructed, there exists a $\gamma \in \Gamma^*$ such that

$$\gamma \notin \bigcup_{\lambda \in \Lambda_{\mathcal{N}*}} B_N^*(\lambda, N-k).$$

We will show that the union of $B_N^*(\lambda, N-k)$ for all $\lambda \in \Lambda_{\mathcal{N}*}$ is not large enough to cover all the points in Γ^*, if the number of points in $\Lambda_{\mathcal{N}*}$ is not sufficiently large.

Our proof is by contradiction. Suppose for a given set of configurations $\Lambda_{\mathcal{N}*}$ and a given k, $1 < k \leqslant N$, $|\Lambda_{\mathcal{N}*}| < k!$, and yet $\bigcup_{\lambda \in \Lambda_{\mathcal{N}*}} B_N^*(\lambda, N-k) \equiv \Gamma^*$. Then

$$\begin{aligned} N! &= \left| \bigcup_{\lambda \in \Lambda_{\mathcal{N}}} B_N^*(\lambda, N-k) \right| \\ &\leqslant \sum_{\lambda \in \Lambda_{\mathcal{N}*}} |B_N^*(\lambda, N-k)| \qquad (12) \\ &\leqslant |\Lambda_{\mathcal{N}*}| \cdot \binom{N}{N-k} (N-k)! \qquad (13) \end{aligned}$$

where (12) follows from the union bound, and (13) follows from (11). Consequently

$$\begin{aligned} |\Lambda_{\mathcal{N}*}| &\geqslant \left[\binom{N}{n-k} (N-k)! \right]^{-1} N! \\ &= k! \qquad (14) \end{aligned}$$

and the proof is complete.

The proof of Theorem 2 is almost immediate from the above derivation. Lemma 2 implies that there is a $\gamma \in \Gamma$ with which all $\lambda \in \Lambda_N$ have at most $k-1$ elements in common, if $|\Lambda_N^*| < k!$. So if the speedup S satisfies $S(k-1) < N$, then the switch is blocking. Hence, $S = \lceil \frac{N}{k-1} \rceil$ is necessary (but not necessarily sufficient) for nonblocking.

Also for Corollary 1, we can rewrite the inequality to have $k < N/S + 1$ and with this we conclude that a speedup S is not sufficient to make a switch nonblocking if the number of configurations

$$|\Lambda_{\mathcal{N}*}| < \lceil N/S \rceil !. \qquad (15)$$

Proof of Theorem 3

From the information set property of the MDS codes, every set of k coordinates run through all possible N^k k-tuples. Thus, given any $\gamma \in \Gamma$, there exists a $\lambda_1 \in \Lambda_{\mathcal{N}}$ whose first k coordinates match with those of γ, i.e., $\lambda_{1,m} = \gamma_m$ for $m = 1, \ldots, k$. Similarly, for some $\lambda_2 \in \Lambda_{\mathcal{N}}$, $\lambda_{2,m} = \gamma_m$ for $m = k+1, \ldots, 2k$. In this manner, we can find $\lfloor N/k \rfloor$ elements, in $\Lambda_{\mathcal{N}}$ each of which exactly matches a distinct set of k coordinates of γ; hence, a total of $k\lfloor N/k \rfloor$ coordinates of γ are matched. The remaining (if any) $N - k\lfloor N/k \rfloor$ coordinates (less than k) can be matched using another element $\lambda_{\lceil N/k \rceil} \in \Lambda_{\mathcal{N}}$. Thus, if the elements of $\Lambda_{\mathcal{N}}$ are chosen to be the codewords of the linear MDS code $(N, k, N-k+1)$, then a total of $\lceil N/k \rceil$ elements, $\lambda_1, \ldots, \lambda_{\lceil N/k \rceil}$ of $\Lambda_{\mathcal{N}}$ is sufficient to have

$$\gamma \subset \bigcup_{l=1}^{\lceil N/k \rceil} \lambda_l$$

for any $\gamma \in \Gamma$.

Distance Properties of MDS Codes

An (N, k) linear code, $\mathcal{C}$ over a finite field $\mathcal{F}$ is a k-dimensional subspace of the vector space $\mathcal{F}^N$. Therefore, $\mathcal{C}$ is composed of $|\mathcal{F}|^k$ N-tuples that are called the *codewords*.

In a *linear code* $\mathcal{C}$, the set of Hamming distances between a codeword and all other codewords is independent of the codeword. The *minimum Hamming distance* between codewords is the minimum element of this set. An (N, k) code with a minimum Hamming distance, d, is called an (N, k, d) code.

For example, the following 3-tuples form a $(3, 2, 2)$ linear code over $\eta = \{0, 1, 2\}$. Note that there are $|\eta|^2 = 9$ codewords

$$\mathcal{C} = \{000, 111, 222, 012, 021, 102, 120, 201, 210\}.$$

A fundamental bound on the parameters of an (N, k, d) code is the *Singleton bound*:

$$d \leqslant N - k + 1.$$

A code meeting this bound, i.e., an $(N, k, N-k+1)$ code is called *maximum distance separable*. *Reed–Solomon* codes are the most famous MDS codes.

Let $\mathcal{C}$ be a code over alphabet of size N. A set of k coordinates is called an *information set* if the codewords run through all N^k possible k-tuples in these coordinates. Thus, if a given set of k coordinates form an information set, there corresponds a unique codeword to every possible set of symbol values in that set of coordinates.

Every subset of coordinates of an MDS code, $(N, k, N-k+1)$ is an information set. Conversely, if every k coordinates of a code is an information set, then the code is an MDS code.

Acknowledgment

The author would like to thank Prof. Robert G. Gallager and Prof. İ. Emre Telatar for all their support and many discussions on this work. He would also like to thank the anonymous reviewers for their valuable feedback.

References

[1] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Circuits*. Boston, MA: Kluwer Academic, 1990.

[2] A. Kam and K. Y. Siu, "Linear complexity algorithms for QoS support in input-queued switches with no speedup," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 6, pp. 1040–1056, 1999.

[3] C. E. Koksal, R. G. Gallager, and C. E. Rohrs, "Rate quantization and service quality over single crossbar switches," in *Proc. INFOCOM*, 2004.

[4] C. E. Koksal, "Impacts of Coherent Crosstalk on the Performance and Scalability of WDM AONs," S.M. thesis, MIT, Cambridge, MA, 1998.

[5] E. R. Berlekamp, *Algebraic Coding Theory*. New York, NY: McGraw-Hill, 1968.

[6] Y. Yang and G. M. Masson, "The necessary conditions for clos type nonblocking multicast networks," *IEEE Trans. Comput.*, vol. 48, no. 11, pp. 1214–1227, 1999.

[7] C. E. Koksal, "Providing QoS Over High Speed Electronic and Optical Networks," Ph.D. Dissertation, MIT, Cambridge, MA, 2002.

[8] C. F. Laywine and G. L. Mullen, "Discrete mathematics using latin squares," in *Wiley-Interscience Series in Discrete Mathematics and Optimization*. Hoboken, NJ: Wiley, 1998.

[9] O. Pretzel, *Error Correcting Codes and Finite Fields*. Oxford, U.K.: Clarendon, 1992.