Consecutive Switch Codes

Sarit Buzaglo*, **Eitan Yaakobi**†, **Yuval Cassuto**‡, and **Paul H. Siegel***§
*Center for Memory and Recording Research, University of California, San Diego, La Jolla, CA 92093-0401 USA
†Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel
‡Electrical Engineering Department, Technion – Israel Institute of Technology, Haifa 32000, Israel
§Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 USA
{*sbuzaglo, psiegel*}*@ucsd.edu*, *yaakobi@cs.technion.ac.il*, *ycassuto@ee.technion.ac.il*

***Abstract*—Switch codes, first proposed by Wang et al., are codes that are designed to increase the parallelism of data writing and reading processes in network switches. A network switch consists of n input ports, k output ports, and m *banks* which store new arriving packets from the input ports in each time slot, called a *generation*. The objective is to store the packets in the banks such that every request of k packets by the output ports, which can be from previous generations, can be handled by reading at most one packet from every bank.**

In this paper we study a new type of switch codes that can simultaneously deliver large symbol requests and good coding rate. These attractive features are achieved by relaxing the request model to a natural sub-class we call *consecutive requests*. For this new request model we define a new type of codes called *consecutive switch codes*. These codes are studied in both the computational and combinatorial models, corresponding to whether the data can be encoded or not. We present several code constructions and prove the optimality of one family of these codes by providing the corresponding lower bound. Lastly, we introduce a construction of switch codes for the case $n = k$, which improves upon the best known results for this case.

I. Introduction

Switch codes were first studied a few years ago by Wang *et al.* in [14] for networking applications. A network switch is a device used to connect between a computer network and external devices. The main task of the network switch is to process and forward packets from the input ports to their designated output ports. Assume that in each time slot, called a **generation**, each input port writes one data packet and each output port can read one packet. Upon arrival, the packets from the input ports are stored in a switch fabric comprising multiple memory **banks**. Then, the output ports read packets from these banks, while each bank can serve exactly one output port. Since the output ports can request packets from previous generations it is common to increase the number of banks in the network switch in order to increase the parallelism in the data writing and reading processes.

Switch codes are a coding scheme which enables one to encode the input packets into the banks such that the packet requests by the output ports can be answered efficiently. Since the requests of the output ports are arbitrary, it is intuitively required that each packet, after encoded to the banks, will have multiple options to be read from the banks. Mathematically speaking, a switch code is required to satisfy the following property. Assume there are n input ports, k output ports, and m banks. In each generation the n packets from the input ports are encoded into m packets which are stored in the banks. Then, in each generation, every request from the output ports for k packets, which may come from previous generations, has to be answered by reading at most one packet from each bank.

Switch codes are associated with the family of codes called **batch codes**. Batch codes were first studied in the previous decade by Ishai *et al.* [8] and recently in [10], [11]. A batch code encodes n information symbols into m buckets such that any request for k information symbols can be answered by reading at most one, and more generally t, symbols from each bucket. If the set of k symbols can have repetitions and each bucket stores a single symbol, then the batch code is called a **multi-set primitive batch code**.

In the original definition of batch codes the packet requests from the output ports are not constrained and can be from any previous generation. However, from a practical point of view it is reasonable to assume that packet requests in each generation are restricted to some ℓ previous consecutive generations. This motivates us to study a new family of switch codes, called **consecutive switch codes**, which follow this restriction on the packet requests. A related family of codes was studied in [4]. We study two classes of these codes, namely, **combinatorial** and **computational** consecutive switch codes. In the combinatorial class, it is assumed that the packets stored in the banks are simply copies of the input packets (that is, they are not coded), while in the computational class the packets can be encoded. We note that the combinatorial model follows the corresponding one for batch codes which was extensively studied in the literature, see e.g [1], [2], [3], [12], as well as the combinatorial model of switch codes which was explored when switch codes were first proposed in [14].

The rest of the paper is organized as follows. In Section II, we formally define switch codes and batch codes and show the equivalence between switch codes and multi-set primitive batch codes. In Section III, we formally define consecutive switch codes and report on constructions of computational consecutive switch codes. In Section IV, we give constructions and a bound for combinatorial consecutive switch codes. Lastly, in Section V we give a construction of switch codes (and hence also of batch codes) for the case $n = k$, which improves upon the state of the art results for this case. Due to the lack of space, some of the proofs of the results in the paper are omitted.

II. Preliminaries

In this section we present some of the definitions and notation used throughout this paper. In particular, we formally define switch codes and describe their connection to batch codes [8].

978-1-5090-1806-2/16/$31.00 ©2016 IEEE

For a positive integer n, denote by $[n]$ the set of n integers $\{1, 2, \ldots, n\}$. For two integers a, b, where $a < b$, denote by $[a, b]$ the set of $b-a+1$ integers $\{a, a+1, \ldots, b\}$. A **multi-set** $\mathcal{M} = \langle i_1, i_2, \ldots, i_k \rangle$ over $[n]$ of size k is a collection of k elements of $[n]$ with repetition, i.e., an element can appear in $\mathcal{M}$ multiple times.

An (m, n)**-code**, $\mathcal{C}$, over $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^m$ of size q^n. An **encoder** for $\mathcal{C}$ is an injection from $\mathbb{F}_q^n$ to $\mathcal{C}$. Throughout this paper, we assume that a code $\mathcal{C}$ is equipped with an encoder, which will be denoted by $\mathcal{E}_\mathcal{C}$. For a string $\mathbf{x} \in \mathbb{F}_q^n$ and for an integer d, let $\mathcal{R}_d : \mathbb{F}_q^n \to \mathbb{F}_q^{dn}$ be the encoder of the d-repetition code which encodes $\mathbf{x}$ to the concatenation of d copies of $\mathbf{x}$.

Definition 1. *An* $(n, k, m, t)_q$***-switch code*** *is an infinite sequence* $\{\mathcal{C}_T\}_{T \geq 1}$ *of* (m, n)*-codes over* $\mathbb{F}_q$ *such that the following hold.*

1) *For every* $T \geq 1$*, a string* $\mathbf{x}^{(T)} \in \mathbb{F}_q^n$ *is encoded by* $\mathcal{E}_{\mathcal{C}_T}$ *to a string* $\mathbf{c}^{(T)} \in \mathbb{F}_q^m$*.*
2) *For every set of* k *pairs* $I = \{(i_1, T_1), (i_2, T_2), \ldots, (i_k, T_k)\} \subset [k] \times \mathbb{N}$*, there exists a multi-set of indices*

$$J = \left\langle \begin{array}{c} j_{1,1}, j_{1,2}, \ldots, j_{1,d_1}, \\ j_{2,1}, j_{2,2}, \ldots, j_{2,d_2}, \\ \vdots \\ j_{k,1}, j_{k,2}, \ldots, j_{k,d_k} \end{array} \right\rangle$$

over $[m]$*, depending only on* I*, such that for every* $1 \leq r \leq k$*, the symbol* $x_{i_r}^{(T_r)}$ *can be recovered from* $c_{j_{r,1}}^{(T_r)}, c_{j_{r,2}}^{(T_r)}, \ldots, c_{j_{r,d_r}}^{(T_r)}$*, and every* $1 \leq j \leq m$ *appears at most* t *times in* J*.*

The set I *is called the* ***request set****, whereas the multi-set* J *is called the* ***recovery set*** *for the request set* I*.*

Concretely, a switch code encodes row-vector inputs into rows of a semi-infinite matrix and is able to recover any k symbols from all inputs by accessing at most t symbols from each of the columns $\{c_j^{(T)}\}_{T \geq 1}$, $1 \leq j \leq m$. The **rate** of the switch code is defined by $R = n/m$. By definition, a switch code is specified by an infinite sequence of codes and hence it might be very complicated to construct good codes, i.e., codes with high rate. Fortunately, as the next lemma states, it is enough to consider only switch codes that are obtained by repeating the same code, for every time instance $T \geq 1$. More precisely, for any set of parameters for which a switch code exists, there also exists a switch code of the same parameters, $\{\mathcal{C}_T\}_{T \geq 1}$, such that $\mathcal{C}_T = \mathcal{C}$, for all $T \geq 1$.

Lemma 1. *If there exists an* $(n, k, m, t)_q$*-switch code* $\{\mathcal{C}_T\}_{T \geq 1}$ *then there exists an* (m, n)*-code* $\mathcal{C}$ *over* $\mathbb{F}_q$ *such that the infinite sequence of codes* $\{\widetilde{\mathcal{C}}_T\}_{T \geq 1}$*, where* $\widetilde{\mathcal{C}}_T = \mathcal{C}$*, for all* $T \geq 1$*, forms an* $(n, k, m, t)_q$*-switch code.*

Two subsequences $\mathbf{u} = w_{i_1} w_{i_2} \ldots w_{i_r}$ and $\mathbf{v} = w_{j_1} w_{j_2} \ldots w_{j_s}$ of a string $\mathbf{w} \in \mathbb{F}_q^m$ are called **disjoint** if $\{i_1, i_2, \ldots, i_r\} \cap \{j_1, j_2, \ldots, j_s\} = \emptyset$. In [8], Ishai et al. proposed multi-set batch codes.

Definition 2. *An* $(n, N, k, m, t)_q$***-multi-set batch code*** *encodes a string* $\mathbf{x} \in \mathbb{F}_q^n$ *into the concatenation of some* m*-strings* $\mathbf{y} = y_1 y_2 \ldots y_m$*,* $y_i \in \mathbb{F}_q^*$*, for all* $1 \leq i \leq m$*, of total length* N*, such that for every multi-set* $\mathcal{M} = \langle i_1, i_2, \ldots, i_k \rangle$ *over* $[n]$ *of size* k*, the* k *symbols* $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ *can be recovered from* $\mathbf{y}$*, where the following conditions hold.*

1) *For every* $1 \leq j \leq m$*, at most* t *symbols from each of the strings* y_i *are accessed.*
2) *For every* $1 \leq r < s \leq k$*, the two subsequences of* $\mathbf{y} = y_1 y_2 \ldots y_m$ *that are used to recover* x_{i_r} *and* x_{i_s}*, respectively, are disjoint.*
3) *The the* k *position sets of the subsequences of* $\mathbf{y}$ *that are accessed depend only on* $\mathcal{M}$*.*

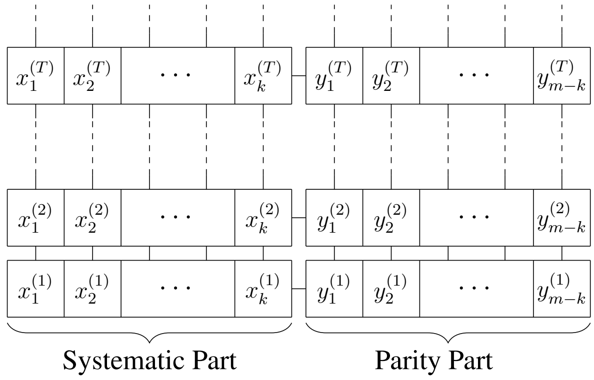

Fig. 1: Systematic $(k, m)_q$-switch code. For every k pairs $(i_1, T_1), (i_2, T_2), \ldots, (i_k, T_k)$, the entries $x_{i_1}^{(T_1)}, x_{i_2}^{(T_2)}, \ldots, x_{i_k}^{(T_k)}$ can be recovered by accessing at most one symbol from each column.

In [8] the authors defined the general concept of batch codes, which are by default not multi-set batch codes. However, we will only consider multi-set batch codes, and henceforth we refer to multi-set batch codes as **batch codes** for short. In [8] the authors also consider the concept of **primitive batch code**, in which each of the m strings to which the input is encoded is of length one, i.e. $y_i \in \mathbb{F}_q$, for all $1 \leq i \leq m$, and hence, $N = m$. Even though the following connection between batch codes and switch codes is somewhat known, we state it here for the completeness of the results in the paper.

Lemma 2. *A code* $\mathcal{C}$ *is an* $(n, N = m, k, m, t = 1)_q$*-primitive batch code if and only if it is an* $(n, k, m, t = 1)_q$*-switch code.*

In this paper we will consider only switch codes for which $n = k$ and $t = 1$, and we denote these codes by $(k, m)_q$-switch codes. This case was also studied in [8], [13], [14], however most of the constructions in [8] (and also all the constructions in [11]) apply to cases in which k is much smaller than n. The case $n = k$ is motivated by the need to equate the switch write and read rates, and $t = 1$ models a simple memory delivering one data packet per time unit. Note that in this case the recovery sets become sets rather than multi-sets. By Lemma 1, we can restrict our discussion to switch codes that are formed by only one code $\mathcal{C}$. Henceforth, an (m, k)-code over $\mathbb{F}_q$ will be called a $(k, m)_q$-switch code if the infinite sequence of codes $\{\mathcal{C}_T = \mathcal{C}\}_{T \geq 1}$ is a $(k, m)_q$-switch code. Furthermore, we mostly consider systematic switch codes, i.e. we assume that for all $\mathbf{x} \in \mathbb{F}_q^k$, $\mathcal{E}_\mathcal{C}(\mathbf{x}) = \mathbf{x}\mathbf{y}$, for some $\mathbf{y} \in \mathbb{F}_q^{m-k}$ (see Figure 1).

III. Consecutive Switch Codes

Primitive batch codes and switch codes are equivalent concepts as Lemma 2 states, yet there is a significant difference between these two concepts. Unlike batch codes, switch codes introduce a time-dimension which motivates us to define a variation of switch codes, which we refer to as **consecutive switch codes**. Consecutive switch codes are designed for a natural sub-class of the request set I in Definition 1. As for switch codes, these codes are capable of retrieving k information symbols, from different time instances, by accessing at most t symbols from each column. However, for ℓ-consecutive switch codes the k information symbols must belong to ℓ-consecutive time instances. The motivation for this variation of switch codes is that in practice two input vectors that were encoded in a short time interval store correlated data, and therefore are likely to be of interest to the same user. Restricting the switch codes to this natural sub-class of requests allows us to increase the rate dramatically, and thus to design practical-rate codes that behave like switch codes for the more common queries of information symbols.

Definition 3. *An $(n,k,m,t)_q$-ℓ-consecutive switch code is an infinite sequence of codes $\{\mathcal{C}_T\}_{T\geq 1}$ of (m,n)-codes over $\mathbb{F}_q$ such that the following hold.*

1) *For every $T \geq 1$, a string $\mathbf{x}^{(T)} \in \mathbb{F}_q^n$ is encoded by $\mathcal{E}_{\mathcal{C}_T}$ to a string $\mathbf{c}^{(T)} \in \mathbb{F}_q^m$.*
2) *For every set of k pairs $I = \{(i_1,T_1),(i_2,T_2),\ldots,(i_k,T_k)\}$, if there exists $\widetilde{T}$ such that $\{T_1,T_2,\ldots,T_k\} \subseteq \{\widetilde{T},\widetilde{T}+1,\ldots,\widetilde{T}+\ell-1\}$, then there exists a recovery set*

$$J = \left\langle \begin{array}{c} j_{1,1}, j_{1,2}, \ldots, j_{1,d_1}, \\ j_{2,1}, j_{2,2}, \ldots, j_{2,d_2}, \\ \vdots \\ j_{k,1}, j_{k,2}, \ldots, j_{k,d_k} \end{array} \right\rangle$$

over $[m]$, depending only on I, such that for every $1 \leq r \leq k$, the symbol $x_{i_r}^{(T_r)}$ can be recovered from $c_{j_{r,1}}^{(T_r)}, c_{j_{r,2}}^{(T_r)}, \ldots, c_{j_{r,d_r}}^{(T_r)}$, and every $1 \leq j \leq m$ appears at most t times in J.

As for general switch codes, we will consider only $(n = k,k,m,t = 1)_q$-ℓ-consecutive switch codes and we denote these codes by $(k,m)_q$-ℓ-consecutive switch codes. Notice that for general switch codes, Lemma 1 states that instead of considering an infinite sequence of codes, it is enough to consider only one code. Unfortunately, the arguments that prove Lemma 1 do not apply to ℓ-consecutive switch codes, since the fact that an infinite sequence of codes $\{\mathcal{C}_T\}_{T\geq 1}$ is an $(n,k,m,t)_q$-ℓ-consecutive switch code does not imply that a subsequence of $\{\mathcal{C}_T\}_{T\geq 1}$ is also an $(n,k,m,t)_q$-ℓ-consecutive switch code. However, for simplicity we will consider only $(k,m)_q$-ℓ-consecutive switch codes that are defined by their first ℓ codes, $\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_\ell$, which are extended periodically, i.e., for every $\widehat{T} \geq 1$, $\mathcal{C}_{\widehat{T}} = \mathcal{C}_T$, where $\widehat{T} \equiv T (\text{mod } \ell)$ and $1 \leq T \leq \ell$. Therefore, throughout this paper, an ℓ-consecutive switch code is a sequence of ℓ codes $\{\mathcal{C}_T\}_{T\in[\ell]}$. Figure 2 (a) illustrates a $(4,6)_2$-2-consecutive switch code in which $\mathcal{C}_1 \neq \mathcal{C}_2$. We remark that this code is optimal, i.e. there does not exist a $(4,5)_2$-2-consecutive code. Moreover, there does not exist a $(4,6)_2$-2-consecutive switch code in which $\mathcal{C}_1 = \mathcal{C}_2$ and $\mathcal{C}_1$ is a linear code.

A $(k,m)_q$-ℓ-consecutive switch code, $\{\mathcal{C}_T\}_{T\in[\ell]}$, is called **combinatorial** if there exists a matrix $F = (F_{T,j})_{T\in[\ell],\, j\in[m]}$ that takes values in $[k]$, such that for every $T \in [\ell]$, $\mathbf{c}^{(T)} = \mathcal{E}_{\mathcal{C}_T}(\mathbf{x}^{(T)}) = x_{F_{T,1}}^{(T)} x_{F_{T,2}}^{(T)} \ldots x_{F_{T,m}}^{(T)}$. The matrix F is called the **index matrix** of the switch code. Note, that $\{\mathcal{C}_T\}_{T\in[\ell]}$ is completely determined by its index matrix F. Intuitively, a combinatorial ℓ-consecutive switch code does not use any coding to encode a string $\mathbf{x}$, only copies of the entries of $\mathbf{x}$ in some order. Therefore, the alphabet size q does not play an important role in the combinatorial case and, henceforth, we omit the subscript q from the notation of such codes and assume that the symbols are taken from some alphabet Σ. A consecutive switch code in which the symbols can be encoded, as opposed to only repeated, is called **computational**; in particular, a combinatorial switch code is by definition also a computational switch code. By default, a consecutive switch code is not combinatorial. For two positive integers k and ℓ, $\ell \leq k$, let $\mathcal{A}(k,\ell)$ be the smallest integer m for which a combinatorial (k,m)-ℓ-consecutive switch code exists. A combinatorial (k,m)-ℓ-consecutive switch code is called **optimal** if $m = \mathcal{A}(k,\ell)$. The simplest combinatorial (k,m)-ℓ-consecutive switch code is the **ℓ-repetition switch code** in which every code $\mathcal{C}^{(T)}$ encodes a string $\mathbf{x} \in \Sigma^k$ into $\mathcal{R}_\ell(\mathbf{x})$, and thus $\mathcal{A}(k,\ell) \leq \ell k$. In Section IV we present a construction of combinatorial consecutive switch codes for every $2 \leq \ell \leq k$. In particular, we show that $\mathcal{A}(k,2) = 2k-1$, while $\mathcal{A}(k,\ell)$ is much smaller than $k\ell$ for $\ell \geq 3$. Figure 2 (b) shows a combinatorial $(4,7)$-2-consecutive switch code for which $m = 7 < 8 = \ell k$ and its index matrix. This code is also optimal. Lastly, for computational 2-consecutive switch codes we have the following theorem.

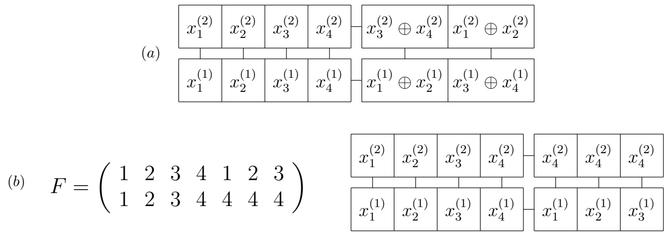

Fig. 2: Examples of 2-consecutive switch codes. (a) A $(4,6)_2$-2-consecutive switch code. (b) A combinatorial $(4,7)$-2-consecutive switch code and its index matrix $F \in [4]^{2\times 7}$.

Theorem 1. *There exists a $(k,m)_q$-2-consecutive switch code, where $m = 1.5k$ and $m < q = O(m)$. Moreover, there exists a $(k,m)_2$-2-consecutive switch code, where $m = 2k - \lfloor \log_2 k \rfloor$.*

IV. Combinatorial Consecutive Switch Codes

In this section we construct combinatorial (k,m)-ℓ-consecutive switch codes, for every $2 \leq \ell \leq k$ and also show a lower bound on $\mathcal{A}(k,2)$. As mentioned above, we construct only combinatorial ℓ-consecutive switch codes of period ℓ. Hence, when constructing such codes, we only specify the first ℓ codes $\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_\ell$, which are used periodically, or equivalently we specify the switch code's index matrix $F \in [k]^{\ell\times m}$.

We start with the simplest case in which $\ell = 2$.

Construction 1. *Let $F \in [k]^{2\times 2k-1}$ be defined by*

$$F = \begin{pmatrix} 1 & 2 & \cdots & k & 1 & 2 & \cdots & k-1 \\ 1 & 2 & \cdots & k & k & k & \cdots & k \end{pmatrix}.$$

Theorem 2. *Let $\{\mathcal{C}_T\}_{T\in[\ell]}$ be the combinatorial switch code whose index matrix is the matrix F from Construction 1. Then*

$\{\mathcal{C}_T\}_{T\in[\ell]}$ *is a* (k,m)*-2-consecutive switch code, with* $m = 2k-1$.

Construction 1 provides us with 2-consecutive switch codes in which $m = 2k-1$, i.e., m is only one less than the length of the trivial 2-repetition switch code. The next theorem states that the code from Construction 1 is optimal, namely $\mathcal{A}(k,2) = 2k-1$.

Theorem 3. *If* $\{\mathcal{C}_T\}_{T\geq 1}$ *is a combinatorial* (k,m)*-2-consecutive switch code, then* $m \geq 2k-1$.

Proof. Let F be the index matrix of $\{\mathcal{C}_T\}_{T\in[2]}$ and let $G(U,V,E)$ be the bipartite graph whose vertex sets are $U=[k]$ and $V=[k]$ and whose edge set E consists of all the edges of the form $e_j = (F_{1,j}, F_{2,j})$, $j\in[m]$. (E may contain parallel edges.) In particular, $|E| = m$. Since $\{\mathcal{C}_T\}_{T\geq 1}$ is a (k,m)-2-consecutive switch code, it follows that for every set of k pairs $I = \{(i_1,T_1),(i_2,T_2),\ldots,(i_k,T_k)\}$, where $i_r\in[k]$ and $T_r\in\{1,2\}$, for all $r\in[k]$, there exist k distinct indices $j_1,j_2,\ldots,j_k$ such that, for all $r\in[k]$, $F_{T_r,j_r} = i_r$. This implies that for all $S_1\subseteq U$ and $S_2\subseteq V$, where $s = |S_1|+|S_2|\leq k$, there exist s edges (u,v), such that $u\in S_1$ or $v\in S_2$.

Assume to the contrary that $m\leq 2k-2$. We will show the existence of $S_1\subseteq U$ and $S_2\subseteq V$, where $s=|S_1|+|S_2|\leq k$, such that the number of edges (u,v) for which $u\in S_1$ or $v\in S_2$ is less than s, and from this we derive a contradiction.

Since $|E| = m < |U|+|V|-1$, it follows that G contains at least $d\geq 2$ connectivity components, $G_1(U_1,V_1,E_1), G_2(U_2,V_2,E_2),\ldots,G_d(U_d,V_d,E_d)$. We claim that at least two of these connectivity components are trees. Indeed, if none of these connectivity components is a tree, then $|E_i|\geq |U_i|+|V_i|$ for all $1\leq i\leq d$ and $|E|\geq 2k$. If only one connectivity component is a tree then $|E|\geq 2k-1$. Assume w.l.o.g. that G_1 and G_2 are trees and $|U_1|+|V_1|\leq |U_2|+|V_2|$. Let $S_1 = U_1$ and $S_2 = V_1$. Then $s = |S_1|+|S_2|\leq (|U|+|V|)/2\leq k$. Notice, that since G_1 is a connectivity component, it follows that for every edge $(u,v)\in E$, $u\in S_1$ if and only if $v\in S_2$, and there exist exactly $|E_1|$ edges that connect an element of S_1 with an element of S_2. Since G_1 is a tree, it follows that $|E_1| = |S_1|+|S_2|-1 < s$ and we derive a contradiction. $\square$

Given a matrix $F\in[k]^{\ell\times m}$ we define the **index graph** of F to be the bipartite graph $G_F(U,V,E)$, with vertex sets $U=[k]$ and $V=[m]$, and an edge set E that consists of all the pairs of the form $(i,j)\in U\times V$, such that $F_{T,j}=i$, for some $T\in[\ell]$. Intuitively, the set V corresponds to the columns of the matrix F, the set U corresponds to all possible entries of F, and an edge (i,j) indicates that i appears in the jth column of F. Note, that E may contain parallel edges if i appears more than once in the jth column of F. Furthermore, the graph G_F has the property that the degree of each vertex in V is exactly ℓ. An example of a matrix $F\in[6]^{3\times 4}$ and its index graph are given in Figure 3. Note, also that if F is the index matrix of a combinatorial ℓ-consecutive switch code with period ℓ, then the edge $(i,j)\in E$ implies that $c_j^{(T)} = x_i^{(T)}$, for some $T\in[\ell]$.

Given a bipartite graph $G(U,V,E)$, for every $S\subseteq U$ we define $N(S)\subseteq V$ to be the set of all vertices in V that are connected by an edge to some vertex in S.

$$F = \begin{pmatrix} 3 & 4 & 5 & 2 \\ 2 & 3 & 4 & 5 \\ 1 & 1 & 6 & 6 \end{pmatrix}$$

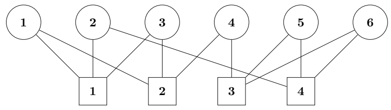

Fig. 3: Example of a matrix $F\in[6]^{3\times 4}$ and its index graph, where the vertex sets U and V are represented by the circles and squares, respectively. Note, that this graph is also a $(6,4,3,3)$-matching graph.

Definition 4. *A bipartite graph* $G(U,V,E)$ *is called a* $(\boldsymbol{k},\boldsymbol{\nu},\boldsymbol{\ell},\boldsymbol{r})$***-matching graph*** *if the following hold.*

1) *Its vertex sets are of sizes* $|U| = k$ *and* $|V| = \nu$.
2) *The degree of each vertex in* V *is* ℓ.
3) *If* $S\subseteq U$ *is of size* $s\leq r$ *then* $|N(S)|\geq s$.

By Hall's theorem [5], condition (3) is equivalent to the condition that for every $S\subseteq U$ of size $s\leq r$ there exists a **matching**, i.e., there exists s disjoint edges from S to V. Figure 3 illustrates a $(6,4,3,3)$-matching graph. Matching graphs are, in a sense, a special type of **expander graphs** [6]. However, we are not aware of any result on expander graphs that fits this description of matching graphs. The study of bipartite expander graphs focuses on the setting in which the degree restriction in item (2) is either omitted or imposed on the vertex set U. Moreover, the neighborhood of $S\subseteq U$ is required to "expand" S, i.e. to be much larger than the set S, and not only to be at least of the same size as S, as required in item (3).

We will show how matching graphs can be useful to construct combinatorial ℓ-consecutive switch codes, but first we need one more definition. The **row cyclic shift mapping** $RS: [k]^{\ell\times m}\to[k]^{\ell\times m}$ is defined by $(RS(F))_{i,j}\stackrel{\text{def}}{=}F_{i-1,j}$, for $i\in[2,\ell]$, and $(RS(F))_{1,j}\stackrel{\text{def}}{=}F_{\ell,j}$, for all $j\in[m]$. Define $RS^0(F)\stackrel{\text{def}}{=}F$ and for $1\leq i\leq \ell-1$, define the ***i*th row cyclic shift** of a matrix F by

$$RS^i(F)\stackrel{\text{def}}{=}\underbrace{RS\circ RS\circ\cdots\circ RS}_{i\text{ times}}(F).$$

Construction 2. *Let* $D\in[k]^{\ell\times\nu}$ *be a matrix whose index graph is a* $(k,\nu,\ell,k/2)$*-matching graph and let* $m = k+(\ell-1)\nu$. *Define the matrix* $F^{(SC)}\stackrel{\text{def}}{=}(F_1|F_2|\ldots|F_\ell)\in[k]^{\ell\times m}$, *where* $F_1\in[k]^{\ell\times k}$ *is the matrix*

$$F_1\stackrel{\text{def}}{=}\begin{pmatrix} 1 & 2 & \ldots & k \\ 1 & 2 & \ldots & k \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & \ldots & k \end{pmatrix}$$

and for all $2\leq b\leq\ell$, $F_b\stackrel{\text{def}}{=}RS^{b-2}(D)$.

Theorem 4. *Let* $\{\mathcal{C}_T\}_{T\in[\ell]}$ *be the combinatorial switch code whose index matrix is the matrix* $F^{(SC)}$ *from Construction 2. Then* $\{\mathcal{C}_T\}_{T\in[\ell]}$ *is a* (k,m)*-*ℓ*-consecutive switch code.*

Example 1. *Let*

$$D = \begin{pmatrix} 3 & 4 & 5 & 2 \\ 2 & 3 & 4 & 5 \\ 1 & 1 & 6 & 6 \end{pmatrix}$$

be the matrix from Figure 3, whose index graph is a $(6,4,3,3)$-matching graph. Then

$$F=\left(\begin{array}{cccccc|cccc|cccc} 1&2&3&4&5&6&3&4&5&2&2&3&4&5\\ 1&2&3&4&5&6&2&3&4&5&1&1&6&6\\ 1&2&3&4&5&6&1&1&6&6&3&4&5&2 \end{array}\right)$$

is the index matrix of a combinatorial $(6,4)$-3-consecutive switch code.

In order to apply Theorem 4 we must construct a $(k,\nu,\ell,k/2)$-matching graph. To this end we use the matrix $M(\omega,\delta)\in[\omega]^{\delta\times\omega}$, $\delta\le\omega$, defined by

$$\begin{pmatrix} 1 & 2 & \cdots & \delta & \delta+1 & \cdots & \omega-1 & \omega\\ 2 & 3 & \cdots & \delta+1 & \delta+2 & \cdots & \omega & 1\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \delta-1 & \delta & \cdots & 2\delta-2 & 2\delta-1 & \cdots & \delta-3 & \delta-2\\ \delta & \delta+1 & \cdots & 2\delta-1 & 2\delta & \cdots & \delta-2 & \delta-1 \end{pmatrix}$$

Given a matrix $M=(M_{i,j})\in[\omega]^{\delta\times\omega}$ and a positive integer α, define the matrix $M+\alpha\in[1+\alpha,\omega+\alpha]^{\delta\times\omega}$, where $(M+\alpha)_{i,j}=M_{i,j}+\alpha$, for all $i\in[\delta]$ and $j\in[\omega]$.

Construction 3. *Let $k=(\ell-2)f^2+(\ell-2)f$, for some positive integer f, $\omega=(\ell-2)f$, and $\delta=\ell-1$. Let $D\in[k]^{\ell\times(\ell-2)f^2}$ be defined by*

$$D\stackrel{\text{def}}{=}\left(\begin{array}{c|c|c|c} M_1 & M_2 & \cdots & M_f\\ \hline \mathbf{x} & \mathbf{x} & \cdots & \mathbf{x} \end{array}\right),$$

where $\mathbf{x}=(k-\omega+1,k-\omega+2,\dots,k)$ and for all $1\le j\le f$, $M_j=(M(\omega,\delta)+(j-1)\omega)\in[(j-1)\omega+1,j\omega]^{\delta\times\omega}$.

Theorem 5. *The index graph of the matrix D from Construction 3 is a $(k,k-(\ell-2)f,\ell,k/2)$-matching graph.*

Combining Theorems 4 and 5 we conclude the following.

Corollary 1. *If $k=(\ell-2)f^2+(\ell-2)f$, for some positive integer f, then*

$$\mathcal{A}(k,\ell)\le\ell k-(\ell-1)(\ell-2)f\approx\ell k-(\ell-1)\sqrt{(\ell-2)k}.$$

V. Constructions of Binary Switch Codes

In this section we consider the conventional definition of switch codes (which are equivalent to primitive batch codes). As mentioned before, $(k,m)_2$-switch codes were studied in [8], [13], [14]. A construction of $(k,m)_2$-switch codes, where $m=k^2/\log_2 k$ was given in [13]. This construction is optimal in the setting in which each parity check bit is restricted to be the sum of at most $\log_2 k$ information bits. In [8], a construction of $(k,m)_2$-switch code with $m=k^{\log_2 3}$ was presented. Our main result in this section is a construction of $(k,m)_2$-switch codes, where $m\approx 2k^{1.5}$. Our construction significantly improves upon the results in [8], [13] and to the best of our knowledge, it is the best known construction of a $(k,m)_2$-switch code. We achieve this result by using the concept of **one-step majority logic decodable code** [9, pp. 273–275]. The connection between this class of codes and distributed storage was first observed in [7]. We show how such codes can be used to construct $(k,m)_q$-switch codes in general and then we apply this method to a specific type of one-step majority logic decodable code.

A (ν,k)-code $\mathcal{C}$ over $\mathbb{F}_q$ is called a **(ν,k)-one-step majority logic decodable code with availability s** if for every $\mathbf{x}\in\mathbb{F}_q^k$, and for all $i\in[k]$, there exist s disjoint subsequences of $\mathcal{E}_{\mathcal{C}}(\mathbf{x})$ that can each recover the symbol x_i.

Construction 4. *Let $s\in[k]$ and let $\mathcal{C}(s)$ be a (ν,k)-one-step majority logic decodable code over $\mathbb{F}_q$ with availability s. Define the (m,k)-code , $\mathcal{C}$, over $\mathbb{F}_q$, where $m=sk+\lfloor k/s\rfloor\cdot\nu$, as follows. For every $\mathbf{x}\in\mathbb{F}_q^k$,*

$$\mathcal{E}_{\mathcal{C}}(\mathbf{x})=\mathcal{R}_s(\mathbf{x})\mathcal{R}_{\lfloor k/s\rfloor}(\mathcal{E}_{\mathcal{C}(s)}(\mathbf{x})),$$

i.e., $\mathcal{E}_{\mathcal{C}}(\mathbf{x})$ is the concatenation of s copies of $\mathbf{x}$ followed by $\lfloor k/s\rfloor$ copies of $\mathcal{E}_{\mathcal{C}(s)}(\mathbf{x})$.

Theorem 6. *The code $\mathcal{C}$ from Construction 4 is a $(k,m)_q$-switch code or equivalently $\mathcal{C}$ is an $(n=k,N=m,k,m,1)_q$-batch code.*

Note that, for a given k, Construction 4 provides the smallest value of m when the availability s is approximately $\sqrt{k}$. One such code is a binary **cyclic difference-set code**. The proof of the following lemma can be found in [9, p. 293].

Lemma 3. *The binary cyclic $(\nu=2^{2r}+2^r+1,k=2^{2r}+2^r-3^r)$-difference-set code is a (ν,k)-one-step majority logic decodable code with availability $s=2^r+1\approx\sqrt{k}$.*

Combining Theorem 6 and Lemma 3 we have the following corollary.

Corollary 2. *Let $\mathcal{C}$ be the code that is obtained from Construction 4 by setting $\mathcal{C}(s)$ to be the binary cyclic $(\nu=2^{2r}+2^r+1,k=2^{2r}+2^r-3^r)$-difference-set code, with $s=2^r+1$. Then $\mathcal{C}$ is a $(k,m)_2$-switch code, where*

$$m=sk+\left\lfloor\frac{k}{s}\right\rfloor\nu\approx 2k^{1.5}.$$

References

[1] S. Bhattacharya, S. Ruj, and B. Roy, "Combinatorial batch codes: A lower bound and optimal constructions," *Advances in Mathematics of Comm.*, vol. 6, pp. 165–174, 2012.

[2] R.A. Brualdi, K.P. Kiernan, S.A. Meyer, and M.W. Schroeder, "Combinatorial batch codes and transversal matroids," *Advances in Mathematics of Comm.*, vol. 4 pp. 419–431, 2010.

[3] C. Bujtás and Z. Tuza, "Relaxations of Hall's condition: Optimal batch codes with multiple queries," *Appl. Anal. Discrete Math*, vol. 6, no. 1, pp. 72–81, 2012.

[4] Y.M. Chee, F. Gao, S.T.H. Teo, and H. Zhang, "Combinatorial systematic switch codes," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 241–245, Hong Kong, Jun. 2015.

[5] R. Diestel, *Graph Theory*, 2nd ed. New York: Springer, 2000, ch. 2.

[6] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, pp. 439–561, October 2006.

[7] P. Huang, E. Yaakobi, H. Uchikawa, and P. H. Siegel, "Binary liniear locally repairable codes," submitted to *IEEE Trans. Inform. Theory*.

[8] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai,"Batch codes and their applications," *in Proc. of the 36th Annual ACM Symposium on Theory of Computing*, vol. 36, pp. 262–271, 2004.

[9] S. Lin and D. J. Costello, *Error Control Coding*. Prentice Hall, 2004.

[10] H. Lipmaa and V. Skachek, "Linear batch codes," *Coding Theory and Applications, CIM Series*, vol. 3. pp. 245–253, 2015.

[11] A.S. Rawat, Z. Song, A.G. Dimakis, and A. Gál, "Batch codes through dense graphs without short cycles," *IEEE Int. Symp. Inf. Theory*, pp. 1477–1481, Hong Kong, Jun. 2015.

[12] N. Silberstein and A. Gál, "Optimal combinatorial batch codes based on block designs," *Designs, Codes and Cryptography*, pp. 1–16, Sep. 2014.

[13] Z. Wang, H. M. Kiah, and Y. Cassuto, "Optimal binary switch codes with small query size," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 636–640, Hong Kong, Jun. 2015.

[14] Z. Wang, O. Shaked, Y. Cassuto, and J. Bruck, "Codes for network switches," *Proc. IEEE Int. Symp. Inf. Theory*, pp. 1057–1061, Istanbul, Turkey, Jul. 2013.