Coding for Tolerance and Detection of Skew in Parallel Asynchronous Communications *

Mario Blaum
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120, USA
blaum@almaden.ibm.com

Jehoshua Bruck †
California Institute of Technology
Mail Stop 136-93
Pasadena, CA 91125, USA
bruck@paradise.caltech.edu

Abstract

We provide a new definition for the concept of skew in parallel asynchronous communications introduced in [2]. The new definition extends and strengthens previously known results on skew. We give necessary and sufficient conditions for codes that can tolerate a certain amount of skew under the new definition. We also extend the results to codes that can tolerate a certain amount of skew and detect a larger amount of skew when the tolerating threshold is exceeded.

1 Introduction

In [2], the concept of skew in parallel asynchronous communications was introduced. We repeat here some of the basic definitions and the problem description. For details, the reader is referred to [2].

Assume that we have n parallel channels, and we can transmit transitions in each one of them. A transition represents a one, while the absence of a transition represents a zero. When the sender transmits a vector of length n, he sends transitions in the channels corresponding to ones in the vector. The transitions arrive at the other end randomly and asynchronously. The receiver sees a sequence $\hat{Z} = x_1, x_2, x_3, \ldots$, where $x_i \in \{1, 2, \ldots, n\}$ indicates that the i-th transition has arrived in channel x_i. For instance, assume that $n = 5$, the transmitter sends the vector 11100 (possibly followed by other vectors) and the receiver sees the sequence

*Partially presented at the IEEE International Symposium on Information Theory (ISIT'98), MIT, Cambridge, MA USA, August 1998

†Supported in part by the NSF Young Investigator Award CCR-9457811, by an IBM Partnership award and by a Sloan Research Fellowship.

$\hat{Z} = 2, 3, 1, \ldots$. This means that the first transition arrived in channel 2, the second one in channel 3, and the third one in channel 1. At that point, the receiver decides that vector 11100 was transmitted. How does he know that he does not need to wait for more transitions to arrive? This problem is solved by using unordered codes [6]. For instance, constant weight codes are unordered codes. Two binary vectors of length n are unordered when their supports (i.e., sets of non-zero coordinates) are unordered. A code is unordered when every pair of distinct codewords is unordered.

Unordered codes are a good solution to the problem of parallel asynchronous communications when there is no *skew* between the transmitted codewords. We say that there is skew in the transmission when transitions not belonging to the current codeword arrive before transmission of the current codeword has been completed. For instance, like in the previous example, assume that 11100 was transmitted, but the receiver sees the sequence $\hat{Z} = 2, 3, 4, 1, \ldots$. Therefore, the third transition arrives in channel 4 and does not belong in the current codeword, whose support is $\{1, 2, 3\}$. Another manifestation of skew is through repeated arrivals. For instance, in the same example, if the receiver sees the sequence $\hat{Z} = 2, 3, 2, 1, \ldots$, the third transition gives a repeated arrival in channel 2. Again, skew has occurred, since the third transition does not belong in the current codeword.

How to avoid skew? The obvious solution is, based on the probabilistic arrival model, to space enough in time the transmission intervals between unordered codewords such that the probability of skew is extremely low. Alternatively, a back channel between receiver and sender may be implemented such that an acknowledgment is sent from receiver to sender once the current codeword has been received. As soon as the sender sees the acknowledgment, he transmits the next codeword. The advantage of these two solutions is that they are easy to implement and unordered codes have fairly low complexity. The disadvantage is that waiting times may be long, and acknowledgment does not allow for pipelined transmission.

An alternative to long transmission intervals and to acknowledgment is to allow a certain amount of skew to occur, and to use codes (and a decoding algorithm) that are capable of tolerating that amount of skew. This has the advantage of reducing the interval of transmission and to allow for pipelined utilization of the channel. However, the codes will be more complex than mere unordered codes, and also the decoding algorithm will have to make more checks than verifying if, after arrival of a transition, the resulting vector belongs in the code or not. But before finding adequate codes, we need a precise definition of skew. In [2], we used two parameters, t_1 and t_2, to characterize the skew of a codeword X with respect to a received sequence $\hat{Z}$. The parameter t_1 represents the number of transitions remaining in X when a transition not in X arrives (by a transition not in X, we include repeated arrivals). The parameter t_2 represents the number of transitions not in X arriving before reception of X is complete. In this case, we say that the skew of X with respect to $\hat{Z}$ is (t_1, t_2) (a (0,0)-skew indicates no skew).

For example, assume that $n = 5$, as above, $X = 11100$ is the transmitted codeword, and the receiver sees the sequence $\hat{Z} = 1, 4, 3, 2, \ldots$. As we can see, the second transition arrives in channel 4, which does not belong to X. Thus, there are two transitions left in X when this transition not belonging in X arrives, therefore, $t_1 = 2$. Similarly, since only one transition

not belonging in X has arrived before reception of X is completed (and this occurs when the fourth transition in channel 2 arrives), then $t_2 = 1$. So, we say that the skew of X with respect to $\hat{Z}$ is (2,1).
This brings us to the following question: is there something we can do to tolerate skew? Can we characterize in terms of distance, codes that can tolerate skew not exceeding (t_1, t_2)? The answer is yes, and the problem was solved in [2].
Before stating the main theorem in [2], we need some notation. We say that a code $\mathcal{C}$ is (t_1, t_2)-skew tolerant (ST) if, given a transmitted codeword X (possibly followed by other codewords) and a received sequence $\hat{Z}$, such that the skew of X with respect to $\hat{Z}$ does not exceed the parameters t_1 and t_2, then, by examining $\hat{Z}$, the receiver can correctly conclude that X was the transmitted codeword.
Given two vectors X and Y, we denote by $N(X, Y)$ the number of locations where X is 1 and Y is 0. For instance, if $X = 11100$ and $Y = 00110$, we have, $N(X, Y) = 2$ and $N(Y, X) = 1$. Notice that $N(X, Y) \geq 0$ and $X = Y$ if and only if $N(X, Y) = N(Y, X) = 0$. Clearly, $N(X, Y) + N(Y, X) = d_H(X, Y)$, where d_H denotes Hamming distance. Also, notice that if $N(X, Y) = 0$, then the support of X is contained in the support of Y, and we say that X is contained in Y. A pair of vectors X and Y is unordered if and only if $N(X, Y) > 0$ and $N(Y, X) > 0$. The main result in [2] is the following:

Theorem 1.1 Let $\mathcal{C}$ be a code, then $\mathcal{C}$ is (t_1, t_2)-ST if and only if, for every pair of distinct codewords X and Y with $N(X, Y) \leq N(Y, X)$, at least one of the following two conditions occurs:

1. $N(X, Y) \geq 1 + \min\{t_1, t_2\}$.
2. $N(X, Y) \geq 1$ and $N(Y, X) \geq t_1 + t_2 + 1$.

For a proof of this result together with a decoding algorithm, see [2]. Observe that the characterization of ST codes given by Theorem 1.1 is symmetric on t_1 and t_2.
One of the purposes of this paper is presenting a new definition of skew, that depends on one parameter only. We then will give and prove necessary and sufficient conditions for codes tolerating the single parameter skew. We will show that the new definition of skew is more natural than the one depending on two parameters, and we will show the connection between the new results and the old ones.
The new definition of skew is roughly as follows: assume, as usual, that a vector X is transmitted and the sequence $\hat{Z}$ is received. Once the last transition in X arrives, we look at previous arrivals in $\hat{Z}$ and we count how many transitions have arrived since the first transition not belonging in X has arrived. This number will be called the skew of X with respect to $\hat{Z}$, and will be denoted by $\mathcal{S}(X; \hat{Z})$. For instance, in our canonical example, we had that $X = 11100$ and $\hat{Z} = 1, 4, 3, 2, \ldots$ was received. We see that the second transition arrives in channel 4, thus, it does not belong in X. Reception of X is completed when the fourth transition arrives in channel 2. Looking back, there were a total of two transitions

since the first transition not in X has arrived, so, the skew of X with respect to $\hat{Z}$ is 2. With the new notation, $\mathcal{S}(X;\hat{Z}) = 2$.
In general, we will say that a code $\mathcal{C}$ is t-skew-tolerant (ST) if, whenever $X \in \mathcal{C}$ is transmitted, possibly followed by other codewords, and $\hat{Z}$ is received such that $\mathcal{S}(X;\hat{Z}) \leq t$, then, by examining $\hat{Z}$, the receiver can correctly determine that X was the transmitted codeword.
In the next section we give precise mathematical definitions of the new concept of skew, and we prove the necessary and sufficient conditions for codes to be t-ST. In Section 3, we extend the concept to codes that can tolerate skew up to t, and detect skew when the skew exceeds t but not $t+s$. We end the paper with some conclusions.

2 Skew-Tolerant Codes

In this section, we present formally the concepts discussed in the previous section.
Notice that binary vectors can be represented by their supports. In the sequel, we will represent binary vectors either by their natural vector representation or by their supports, without an explicit distinction between the two.
Let X be a subset of $\{1, 2, \ldots, n\}$. Let $\hat{Z} = x_1, x_2, x_3, \ldots$ be a sequence, where each $x_i \in \{1, 2, \ldots, n\}$. Let $Z_m \subseteq \{1, 2, \ldots, n\}$ be the set $\{x_1, x_2, \ldots, x_m\}$ (remember that in a set, repeated entries are equivalent to single entries). Let

$$r(X;\hat{Z}) \quad = \quad \min\{m : X \subseteq Z_m\} \tag{1}$$

In words, $r(X;\hat{Z})$ denotes the index of the arrival of the last transition in X. For instance, in our canonical example, we had that $X = 11100$ and $\hat{Z} = 1, 4, 3, 2, \ldots$ was received. Therefore, since the last transition in X to arrive is the fourth transition, we have that $r(X;\hat{Z}) = 4$.
When the context is clear, we denote $r = r(X;\hat{Z})$. If $Z_m \neq X$ for all m, we define $r = \infty$.
Next we give a formal definition of skew.

Definition 2.1 Assume that X is the transmitted vector and $\hat{Z}$ the received sequence. We define the skew of X with respect to $\hat{Z}$ as t, denoted $\mathcal{S}(X;\hat{Z}) = t$, as follows:

1. If $r = |X|$, then $t = 0$ (i.e., no skew).
2. If $|X| < r < \infty$, then t is the largest such that:
   (a) $Z_{r-(t+1)} \subseteq X$; and
   (b) either $x_{r-t} \notin X$ or $x_{r-t} \in Z_{r-(t+1)}$.
3. If $r = \infty$, then $t = \infty$.

Let us look more closely at Definition 2.1. The first condition says that $Z_r = X$ and, since $|X| = r$, there are no repeated arrivals. Thus, no skew has occurred. The second condition reveals the presence of skew: since $|X| < r$, then either some transition not in X has arrived before reception of X is completed, or we had a repeated arrival. Notice that transitions $1, 2, \ldots, r-(t+1)$ are all in X, while transition $r-t$ is either not in X or it is a repeated arrival. When transition $r-t$ arrives, it is either a repeated arrival or it does not belong in X. Skew has occurred and the magnitude of this skew is given by the number t. The third condition simply defines the skew as infinite when X is never received.
The next example illustrates Definition 2.1.

Example 2.1 Let $X = \{1, 2, 3\} \subseteq \{1, 2, 3, 4, 5, 6\}$.
If $\hat{Z} = 3, 1, 2, 3, 4, 5, \ldots$, then, since $|X| = 3 = r$, $\mathcal{S}(X; \hat{Z}) = 0$.
If $\hat{Z} = 1, 2, 4, 3, 4, 3, \ldots$, then $|X| = 3 < 4 = r$. Since $Z_{4-2} = \{1, 2\} \subseteq X$ and $x_{4-1} = 4 \notin X$, then $\mathcal{S}(X; \hat{Z}) = 1$.
If $\hat{Z} = 1, 1, 2, 4, 3, 6, \ldots$, then $|X| = 3 < 5 = r$. Since $Z_{5-4} = \{1\} \subseteq X$ and $x_{5-3} = 1 \in Z_{5-4}$, then $\mathcal{S}(X; \hat{Z}) = 3$.

Next, we define skew-tolerant codes.

Definition 2.2 Let $\mathcal{C} = \{X, Y, \ldots\}$ be a code. We say that $\mathcal{C}$ is t-skew tolerant (ST) if, for each $X \in \mathcal{C}$ and sequence $\hat{Z}$ such that $\mathcal{S}(X; \hat{Z}) \leq t$, then $\mathcal{S}(Y; \hat{Z}) > t$ for all $Y \in \mathcal{C}$, $Y \neq X$.

Definition 2.2 means that if X is a transmitted codeword followed by other codewords, $\hat{Z}$ is the received sequence and $\mathcal{S}(X; \hat{Z}) \leq t$, then, by examining $\hat{Z}$, the receiver can correctly conclude that X was the transmitted codeword. This correct identification occurs when the r-th transition arrives, i.e., when Z_r is examined.
Now, what is the relationship between t-skew as given by Definition 2.1 and (t_1, t_2)-skew as defined in [2]? Observe that if X is a transmitted codeword followed by other codewords, $\hat{Z}$ is the received sequence and the skew of X with respect to $\hat{Z}$ when defined by two parameters is equal to (t_1, t_2), then $\mathcal{S}(X; \hat{Z}) = t_1 + t_2 - 1$. Explicitly,

Corollary 2.1 Let $\mathcal{C}$ be a t-ST code. Then, $\mathcal{C}$ is (t_1, t_2)-ST for each t_1, t_2 such that $t_1 + t_2 \leq t - 1$. In particular, if a code is $(2t-1)$-ST, it is also (t, t)-ST, and a code is 1-ST if and only if it is (1,1)-ST.

The following theorem is one of our main results. It gives necessary and sufficient conditions characterizing t-ST codes. Its proof will be a special case of the proof of the forthcoming Theorem 3.2, which gives more general conditions for codes combining skew-tolerance and skew-detection.

Theorem 2.1 A code $\mathcal{C}$ is t-ST if and only if, for every $X, Y \in \mathcal{C}$, $X \neq Y$, with $N(X, Y) \leq N(Y, X)$, the following condition is satisfied:

$$N(X, Y) \geq 1 \quad \text{and} \quad 2N(X, Y) + N(Y, X) \geq 2t + 3 \tag{2}$$

Below we give some examples of the conditions that codewords in a t-ST code $\mathcal{C}$ satisfy for $1 \leq t \leq 3$ according to Theorem 2.1. In all cases, we assume that $X, Y \in \mathcal{C}$, $X \neq Y$ and $N(X,Y) \leq N(Y,X)$.

$\mathcal{C}$ is 1-ST if and only if

1. $N(X,Y) = 1$ and $N(Y,X) \geq 3$; or
2. $N(X,Y) \geq 2$.

$\mathcal{C}$ is 2-ST if and only if

1. $N(X,Y) = 1$ and $N(Y,X) \geq 5$; or
2. $N(X,Y) \geq 2$ and $N(Y,X) \geq 3$.

$\mathcal{C}$ is 3-ST if and only if

1. $N(X,Y) = 1$ and $N(Y,X) \geq 7$; or
2. $N(X,Y) = 2$ and $N(Y,X) \geq 5$; or
3. $N(X,Y) \geq 3$.

Connecting Theorems 1.1 and 2.1, we can obtain a stronger version of Corollary 2.1 when $t_1 \neq t_2$. Explicitly:

Corollary 2.2 Let $t_1, t_2 \geq 1$ and $\max\{t_1, t_2\} + 3\min\{t_1, t_2\} - 2 \leq 2t$. Then, if a code $\mathcal{C}$ is t-ST, it is also (t_1, t_2)-ST.

Proof: Assume that $\mathcal{C}$ is t-ST, and $t_1, t_2 \geq 1$, $\max\{t_1, t_2\} + 3\min\{t_1, t_2\} - 2 \leq 2t$. Since Theorem 1.1 is symmetrical on t_1 and t_2, we may assume that $t_1 \geq t_2$, therefore

$$t_1 + 3t_2 - 2 \quad \leq \quad 2t. \tag{3}$$

Let $X, Y \in \mathcal{C}$ with $N(X,Y) \leq N(Y,X)$. We have to show that at least one of the two conditions in Theorem 1.1 is satisfied. If $N(X,Y) \geq t_2 + 1$, then condition 1 in Theorem 1.1 is satisfied, so assume that

$$N(X,Y) \leq t_2. \tag{4}$$

Thus,

$$\begin{aligned} N(Y,X) &\geq 2t - 2N(X,Y) + 3 \\ &\geq (t_1 + 3t_2 - 2) - 2t_2 + 3 \quad \text{(by (3) and (4))} \\ &\geq t_1 + t_2 + 1. \end{aligned}$$

Thus, X and Y satisfy condition 2 in Theorem 1.1, completing the proof. □

For instance, a 3-ST code is (1,5)-ST, (2,2)-ST and (5,1)-ST. This looks like a paradox. How can a 3-ST code be (1,5)-ST if a skew equal to (1,5) corresponds to a skew equal to 5? The answer is, a (1,5)-ST code can tolerate skew not exceeding (1,5), but not other types of skew, like a skew equal to (2,2), which corresponds to a skew equal to 3.
We next give a decoding algorithm for t-ST codes. We will prove in the Appendix that the algorithm correctly decodes the transmitted codeword when condition (2) is satisfied for every pair of distinct codewords in the code.

Algorithm 2.1 (Decoding Algorithm for t-ST Codes) Let the received sequence be $\hat{Z} = x_1, x_2, \ldots, x_i, \ldots$. Then:

RESET: Set the initial conditions $i \leftarrow 0$, $X \leftarrow \emptyset$, $B \leftarrow \emptyset$, $W \leftarrow \emptyset$ and $\{x_j\} = \emptyset$ for $j \leq 0$.
START: Set $i \leftarrow i+1$ and $W \leftarrow \left(W - \{x_{i-(t+1)}\}\right) \cup \{x_{i-1}\}$.
If $x_i \in X$ then:
If $|B| = t$ or $x_i \in B$, then declare an uncorrectable error and STOP.
Else, set $B \leftarrow B \cup \{x_i\}$ and go to **START**.
Else, set $X \leftarrow X \cup \{x_i\}$.
If $X - A \notin \mathcal{C}$ for any $A \subseteq W - B$ then go to **START**.
Else, output $X - A$, where $A \subseteq W - B$ and $X - A \in \mathcal{C}$.
If $A \cup B \in \mathcal{C}$, then output $A \cup B$ and set $A \leftarrow \emptyset$ and $B \leftarrow \emptyset$.
Set $\hat{Z} \leftarrow x_{j_1}, x_{j_2}, \ldots, x_{j_s}, x_{i+1}, x_{i+2}, \ldots$, where $\{x_{j_1}, x_{j_2}, \ldots, x_{j_s}\} = A \cup B$, $j_1 < j_2 < \ldots < j_s$, and go to **RESET**.

Let us look more closely at Algorithm 2.1. The index i counts the new arrivals and the set X registers the incremental vectors. The set W represents a buffer storing the last t arrivals, if any. At step **START**, a new arrival x_i is registered. Then, the oldest arrival is discarded from the buffer W and the arrival previous to the current one is incorporated into W. The set B keeps track of the repeated arrivals, if any, while the set A attempts to find those transitions that are not in the current codeword.
The main step of the algorithm checks, for each received x_i, if the set Z_i minus one of the $2^{t-|B|}$ subsets of $\{x_{i-t}, x_{i-(t-1)}, \ldots, x_{i-1}\} - B$ (including the empty set) is in the code $\mathcal{C}$. When and if it does, then the resulting vector is given as output of the algorithm, the set $A \cup B$ is reserved as the initial arrivals of the next codeword and the process is restarted.

As we mentioned above, the proof of the **if** part of Theorem 2.1, is a special case of the **if** part of Theorem 3.2, to be given in Section 3. We end this section with an example of the execution of Algorithm 2.1.

Example 2.2 Consider the code $\mathcal{C} = \{X, Y, Z\}$, where $X = 111110000 = \{1, 2, 3, 4, 5\}$, $Y = 000001000 = \{6\}$ and $Z = 001110111 = \{3, 4, 5, 7, 8, 9\}$. Since $N(X, Y) = 5$ and $N(Y, X) = 1$, $N(X, Z) = 2$ and $N(Z, X) = 3$ and $N(Y, Z) = 1$ and $N(Z, Y) = 6$, code $\mathcal{C}$ is 2-ST. Assume that the following sequence has been received:

$$\hat{Z} = 2, 1, 6, 5, 4, 4, 7, 3, 9, 5, 8, 3, \ldots.$$

The following table gives the execution of Algorithm 2.1:

i	$\hat{Z}$	x_i	X	W	B	A	$X - A$	Output
0	$2, 1, 6, 5, 4, 4, 7,$ $3, 9, 5, 8, 3, \ldots$		$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
1		2	$\{2\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{2\}$	
2		1	$\{2, 1\}$	$\{2\}$	$\emptyset$	$\emptyset$ $\{2\}$	$\{2, 1\}$ $\{1\}$	
3		6	$\{2, 1, 6\}$	$\{2, 1\}$	$\emptyset$	$\emptyset$ $\{1\}$ $\{2\}$ $\{2, 1\}$	$\{2, 1, 6\}$ $\{2, 6\}$ $\{1, 6\}$ $\{6\}$	000001000
0	$2, 1, 5, 4, 4, 7,$ $3, 9, 5, 8, 3, \ldots$		$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
1		2	$\{2\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{2\}$	
2		1	$\{2, 1\}$	$\{2\}$	$\emptyset$	$\emptyset$ $\{2\}$	$\{2, 1\}$ $\{1\}$	
3		5	$\{2, 1, 5\}$	$\{2, 1\}$	$\emptyset$	$\emptyset$ $\{1\}$ $\{2\}$ $\{2, 1\}$	$\{2, 1, 5\}$ $\{2, 5\}$ $\{1, 5\}$ $\{5\}$	
4		4	$\{2, 1, 5, 4\}$	$\{1, 5\}$	$\emptyset$	$\emptyset$ $\{5\}$ $\{1\}$ $\{5, 1\}$	$\{2, 1, 5, 4\}$ $\{2, 1, 4\}$ $\{2, 5, 4\}$ $\{2, 4\}$	
5		4	$\{2, 1, 5, 4\}$	$\{5, 4\}$	$\{4\}$			
6		7	$\{2, 1, 5, 4, 7\}$	$\{4\}$	$\{4\}$	$\emptyset$	$\{2, 1, 5, 4, 7\}$	
7		3	$\{2, 1, 5, 4, 7, 3\}$	$\{4, 7\}$	$\{4\}$	$\emptyset$ $\{7\}$	$\{2, 1, 5, 4, 7, 3\}$ $\{2, 1, 5, 4, 3\}$	111110000

i	$\hat{Z}$	x_i	X	W	B	A	$X-A$	Output
0	$4,7,9,5,8,3,\ldots$		$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
1		4	$\{4\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{4\}$	
2		7	$\{4,7\}$	$\{4\}$	$\emptyset$	$\emptyset$ $\{4\}$	$\{4,7\}$ $\{7\}$	
3		9	$\{4,7,9\}$	$\{4,7\}$	$\emptyset$	$\emptyset$ $\{7\}$ $\{4\}$ $\{4,7\}$	$\{4,7,9\}$ $\{4,9\}$ $\{7,9\}$ $\{9\}$	
4		5	$\{4,7,9,5\}$	$\{7,9\}$	$\emptyset$	$\emptyset$ $\{9\}$ $\{7\}$ $\{7,9\}$	$\{4,7,9,5\}$ $\{4,7,5\}$ $\{4,9,5\}$ $\{4,5\}$	
5		8	$\{4,7,9,5,8\}$	$\{9,5\}$	$\emptyset$	$\emptyset$ $\{5\}$ $\{9\}$ $\{9,5\}$	$\{4,7,9,5,8\}$ $\{4,7,9,8\}$ $\{4,7,5,8\}$ $\{4,7,8\}$	
6		3	$\{4,7,9,5,8,3\}$	$\{5,8\}$	$\emptyset$	$\emptyset$	$\{4,7,9,5,8,3\}$	001110111

Let us remark that Algorithm 2.1 deals with skew between adjacent codewords only. However, the conditions are more general: the skew can come from any codeword as long as the t constraint is not broken. In order to correct skew coming from non-adjacent codewords, we have to modify slightly Algorithm 2.1, by taking into account repeated arrivals coming from non-adjacent codewords. In its present form, Algorithm 2.1 stops and declares an uncorrectable error when a repeated arrival appears more than once. For simplicity, we omit here the complete algorithm.

3 Skew-Tolerant and Skew-Detecting Codes

In this section we extend the results of the previous section to codes that can detect skew and also to codes that can simultaneously tolerate and detect skew when the threshold of skew-tolerance is exceeded. We start with some definitions.

Definition 3.1 Let $\mathcal{C}=\{X,Y,\ldots\}$ be a code. We say that $\mathcal{C}$ is s-skew detecting (SD) if, for each $X\in\mathcal{C}$ and sequence $\hat{Z}$ such that $\mathcal{S}(X;\hat{Z})\leq s$, then $\mathcal{S}(Y;\hat{Z})>0$ for all $Y\in\mathcal{C}$, $Y\neq X$.

Definition 3.1 states that when a codeword X is transmitted, possibly followed by other codewords, giving a received sequence $\hat{Z}$, then, by examining $\hat{Z}$, if there is no skew of X with respect to $\hat{Z}$, i.e., $\mathcal{S}(X,\hat{Z})=0$, then X will be correctly decoded; but if there is skew not exceeding s, i.e., $0<\mathcal{S}(X,\hat{Z})\leq s$, then there is no codeword different from X that can be given as output of the decoder. In fact, the skew will be detected when we have a repeated arrival.

Codes that are (s_1, s_2)-SD were studied in [2]. Again, here we are replacing both parameters with only one. The next theorem characterizes s-SD codes.

Theorem 3.1 A code $\mathcal{C}$ is s-SD if and only if $\mathcal{C}$ has minimum distance $d \geq s+2$ and it is unordered.

Unordered error-correcting codes were studied in [3]. We refer the reader to that paper for efficient constructions. Theorem 3.1 is a special case of the forthcoming Theorem 3.2.
The next step is studying codes that can tolerate a skew not exceeding t, and that can detect a skew exceeding t but not $t+s$. In [4], codes that are (t_1, t_2)-ST (t_1+s_1, t_2+s_2)-SD were studied. Again, our purpose is obtaining similar results by using our new definition of skew.

Definition 3.2 Let $\mathcal{C} = \{X, Y, \ldots\}$ be a code. We say that $\mathcal{C}$ is t-skew tolerant $(t+s)$-skew detecting (t-ST $(t+s)$-SD) if, for each $X \in \mathcal{C}$ and sequence $\hat{Z}$ such that $\mathcal{S}(X; \hat{Z}) \leq t+s$, then $\mathcal{S}(Y; \hat{Z}) > t$ for all $Y \in \mathcal{C}$, $Y \neq X$.

Notice that Definition 3.2 extends Definitions 2.2 and 3.1. In effect, a t-ST code is a t-ST $(t+s)$-SD code with $s=0$, while an s-SD code is a t-ST $(t+s)$-SD code with $t=0$. The following theorem characterizes t-ST $(t+s)$-SD codes.

Theorem 3.2 A code $\mathcal{C}$ is t-ST $(t+s)$-SD if and only if, for any $X, Y \in \mathcal{C}$, $X \neq Y$, with $N(X,Y) \leq N(Y,X)$, either

$$1 \leq N(X,Y) \leq t+1 \quad \text{and} \quad 2N(X,Y) + N(Y,X) \geq 2t+s+3, \tag{5}$$

or

$$N(X,Y) \geq t+2 \quad \text{and} \quad N(X,Y) + N(Y,X) \geq t+s+2. \tag{6}$$

We prove Theorem 3.2 in the Appendix.
Notice that, when $s=0$, conditions (5) and (6) become

$$1 \leq N(X,Y) \leq t+1 \quad \text{and} \quad 2N(X,Y) + N(Y,X) \geq 2t+3,$$

or

$$N(X,Y) \geq t+2 \quad \text{and} \quad N(X,Y) + N(Y,X) \geq t+2.$$

The second one is equivalent to $2N(X,Y) + N(Y,X) \geq t+2+N(X,Y) \geq 2t+4$, so conditions (5) and (6) reduce to condition (2) in Theorem 2.1.
On the other hand, making $t=0$, they give

$$N(X,Y)=1 \quad \text{and} \quad 2N(X,Y)+N(Y,X)\geq s+3,$$

or

$$N(X,Y)\geq 2 \quad \text{and} \quad d=N(X,Y)+N(Y,X)\geq s+2.$$

The first one of these equations gives $N(X,Y)=1$ and
$d=N(X,Y)+N(Y,X)\geq s+3-N(X,Y)=s+2$, so both of them reduce to the condition in Theorem 3.1. Thus, Theorem 3.2 in fact generalizes Theorems 2.1 and 3.1. Below are two examples of the conditions of t-ST $(t+s)$-SD codes according to Theorem 3.2. In all cases, $X,Y\in\mathcal{C}$, $X\neq Y$ and $N(X,Y)\leq N(Y,X)$.

$\mathcal{C}$ is 1-ST 5-SD (i.e., $t=1$, $s=4$) if and only if

1. $N(X,Y)=1$ and $N(Y,X)\geq 7$; or
2. $N(X,Y)=2$ and $N(Y,X)\geq 5$; or
3. $N(X,Y)\geq 3$ and $N(Y,X)\geq 4$.

$\mathcal{C}$ is 2-ST 8-SD (i.e., $t=2$, $s=6$) if and only if

1. $N(X,Y)=1$ and $N(Y,X)\geq 11$; or
2. $N(X,Y)=2$ and $N(Y,X)\geq 9$; or
3. $N(X,Y)=3$ and $N(Y,X)\geq 7$; or
4. $N(X,Y)=4$ and $N(Y,X)\geq 6$; or
5. $N(X,Y)\geq 5$.

We can connect t-ST $(t+s)$-SD codes with (t_1,t_2)-ST (t_1+s_1,t_2+s_2)-SD codes similarly to Corollary 2.2. Since the necessary and sufficient conditions for (t_1,t_2)-ST (t_1+s_1,t_2+s_2)-SD codes are quite complicated [4], we omit this result.
In order to construct t-ST and t-ST $(t+d)$-SD codes, we can use unordered error-correcting codes [3], or the more sophisticated results of [5]. However, better constructions might be obtained by fully using the conditions of Theorems 2.1 and 3.2.

4 Conclusions

We studied the problem of pipelined transmission in parallel asynchronous communications allowing a certain amount of skew. We gave a new definition of skew depending on only one parameter, as opposed to previous definitions depending on two parameters. We presented necessary and sufficient conditions for codes tolerating a certain amount of skew under the new definition. We extended the results to simultaneous tolerance and detection of skew. More research is needed to construct efficient skew-tolerant and skew-detecting codes under the new definition of skew.

Acknowledgement: We wish to thank Tom Verhoeff for his numerous comments and suggested improvements.

5 Appendix

Proof of the "only if" part of Theorem 3.2: Assume that code $\mathcal{C}$ is t-ST $(t+s)$-SD and take $X, Y \in \mathcal{C}$, $X \neq Y$, such that $N(X,Y) \leq N(Y,X)$. Given an arbitrary set S, let $\hat{S}$ denote a sequence of elements of S transmitted in some order (with no repeated arrivals). Observe first that X and Y must be unordered (i.e., $N(X,Y) \geq 1$). Otherwise, if $X \subseteq Y$, and the sequence $\hat{Z} = \hat{X}, \hat{Y}$ is received, then, according to Definition 2.1, $\mathcal{S}(X,\hat{Z}) = 0 = \mathcal{S}(Y,\hat{Z})$, therefore the receiver cannot decide which vector was transmitted first, whether X or Y. This contradicts the assumption that $\mathcal{C}$ is t-ST $(t+s)$-SD (Definition 3.2).
Next define the following sets: $C = X \cap Y$, $A = X - Y$, and $B = Y - X$. Notice that $|A| = N(X,Y)$ and $|B| = N(Y,X)$. Observe also that

$$d_H(X,Y) = N(X,Y) + N(Y,X) \geq t + s + 2. \tag{7}$$

In effect, if the sequence $\hat{Z} = \hat{C}, \hat{A}, \hat{B}$ is received, then, according to Definition 2.1, $\mathcal{S}(X,\hat{Z}) = 0$ and $\mathcal{S}(Y,\hat{Z}) = |A| + |B| - 1 = N(X,Y) + N(Y,X) - 1$. Since $\mathcal{C}$ is t-ST $(t+s)$-SD, according to Definition 3.2, we must have $N(X,Y) + N(Y,X) - 1 \geq t + s + 1$, proving (7).
In order to prove condition (5), assume that $1 \leq N(X,Y) \leq t+1$. By (7),
$|B| \geq t + s - N(X,Y) + 2$, thus, we can partition B as $B = B_1 \cup B_2$, where $B_1 \cap B_2 = \emptyset$ and $|B_1| = t + s - N(X,Y) + 1$ (thus, $|B_2| > 0$ and $B_2 \neq \emptyset$). If $B_1 = \emptyset$, then $s = 0$ and $N(X,Y) = t + 1 \leq N(Y,X)$. Thus, $2N(X,Y) + N(Y,X) \geq 2(t+1) + (t+1) > 3t + 2$, satisfying (5) when $s = 0$. So, assume that $B_1 \neq \emptyset$ and the following sequence $\hat{Z}$ is received:

$$\hat{Z} \;=\; \hat{C}, \hat{B}_1, \hat{A}, \hat{B}_2, \ldots \tag{8}$$

According to Definition 2.1 and (8),

$$\mathcal{S}(X;\hat{Z}) = |B_1| + |A| - 1 = (t + s - N(X,Y) + 1) + N(X,Y) - 1 = t + s, \tag{9}$$

while

$$\mathcal{S}(Y;\hat{Z}) = |A| + |B_2| - 1 = N(X,Y) + |B_2| - 1. \tag{10}$$

Since $\mathcal{C}$ is t-ST $(t+s)$-SD, by (9), (10) and Definition 3.2, we must have that $N(X,Y) + |B_2| - 1 \geq t + 1$, or, $|B_2| \geq t - N(X,Y) + 2$. Therefore, using this result together with the cardinality of B_1, we obtain

$$\begin{aligned} N(Y,X) &= |B| = |B_1| + |B_2| \\ &\geq (t + s - N(X,Y) + 1) + (t - N(X,Y) + 2) \\ &= 2t + s + 3 - 2N(X,Y), \end{aligned}$$

proving condition (5).
In order to prove condition (6), assume that $|A| = N(X,Y) \geq t + 2$. We can partition A as $A = A_1 \cup A_2$ with $A_1 \cap A_2 = \emptyset$, $|A_1| = t$ and $|A_2| = N(X,Y) - t$ (in particular, $A_1 \neq \emptyset$ and $A_2 \neq \emptyset$). Moreover, let $b \in B$ (notice that $|B| \geq |A| \geq 2$, thus, $B - \{b\} \neq \emptyset$).
Assume that the following sequence $\hat{Z}$ is received:

$$\hat{Z} = \hat{C}, \hat{A}_2, b, \hat{A}_1, \hat{(B - \{b\})}, \ldots \tag{11}$$

According to Definition 2.1 and (11), $\mathcal{S}(X;\hat{Z}) = |A_1| = t$, while

$$\mathcal{S}(Y;\hat{Z}) = |A| + |B| - 1 = N(X,Y) + N(Y,X) - 1 \tag{12}$$

Since $\mathcal{C}$ is t-ST $(t+s)$-SD and $\mathcal{S}(X;\hat{Z}) = t$, by Definition 3.2, $\mathcal{S}(Y;\hat{Z}) \geq t + s + 1$ and, by (12), condition (6) follows, completing the proof of the necessary conditions. □

In order to prove the "if" part of Theorem 3.2, we will assume that we are using Algorithm 2.1 for the decoding. We start with two lemmas and then we use them to prove the theorem itself.

Lemma 5.1 Assume that X and Y are distinct codewords in a code $\mathcal{C}$ and there is a $0 \leq j \leq t$ such that

$$N(X,Y) \leq j + 1 \quad \text{and} \quad N(Y,X) \leq 2(t - j) + s. \tag{13}$$

Then, X and Y do not satisfy conditions (5) or (6).

Proof: Notice that X and Y cannot satisfy (6), since $\min\{N(X,Y), N(Y,X)\} \leq j+1 \leq t+1$. So, assume first that $N(X,Y) \leq N(Y,X)$. Then,

$$2N(X,Y) + N(Y,X) \leq 2(j+1) + 2(t-j) + s = 2t + s + 2,$$

so X and Y cannot satisfy (5).
Assume next that $N(X,Y) > N(Y,X)$, therefore, by (13), in particular $N(Y,X) \leq j$. This fact, together with (13), give

$$\begin{aligned} 2N(Y,X) + N(X,Y) &= N(Y,X) + N(Y,X) + N(X,Y) \\ &\leq (2(t-j)+s) + j + (j+1) \\ &= 2t + s + 1, \end{aligned}$$

so X and Y cannot satisfy (5) in this case either. □

Lemma 5.2 Assume that X and Y are distinct codewords in a code $\mathcal{C}$ and there is a $j > t$ such that

$$N(X,Y) \leq j+1 \quad \text{and} \quad N(Y,X) \leq t+s-j. \tag{14}$$

Then, X and Y do not satisfy conditions (5) or (6).

Proof: Assume first that $\min\{N(X,Y), N(Y,X)\} \geq t+2$. Then, condition (5) cannot occur, and, by (14),

$$N(X,Y) + N(Y,X) \leq (j+1) + (t+s-j) = t + s + 1,$$

so condition (6) cannot occur either.
So, assume that $\min\{N(X,Y), N(Y,X)\} \leq t+1$, then, condition (6) cannot occur. If $N(X,Y) \leq N(Y,X)$, since $t - j < 0$, by (14), we have,

$$2N(X,Y) + N(Y,X) \leq 2(t+1) + (t+s-j) < 2t + s + 2,$$

so condition (5) cannot occur. So, assume that $N(X,Y) > N(Y,X)$, thus $N(Y,X) \leq t+1$. Then, by (14),

$$\begin{aligned} 2N(Y,X) + N(X,Y) &= N(Y,X) + N(Y,X) + N(X,Y) \\ &\leq (t+s-j) + (t+1) + (j+1) \\ &= 2t + s + 2, \end{aligned}$$

so condition (5) cannot occur. □

Proof of the "if" part of Theorem 3.2: Assume that for every pair of distinct codewords X, Y in $\mathcal{C}$ with $N(X,Y) \leq N(Y,X)$ one of conditions (5) or (6) is satisfied. We have to show that the code is t-ST $(t+s)$-SD according to Definition 3.2. Let Y be the transmitted codeword and $\hat{Z}$ the received sequence, and assume that $0 \leq \mathcal{S}(Y;\hat{Z}) \leq t+s$. We will attempt to decode Y using Algorithm 2.1. Let $r = r(Y;\hat{Z})$ be as defined by (1). If $\mathcal{S}(Y;\hat{Z}) \leq t$, there is a subset $A \subseteq \{x_{r-t}, x_{r-(t-1)}, \ldots, x_{r-1}\}$ such that $Z_r - A = Y$. Thus, the algorithm will find Y when x_r arrives. On the other hand, if $t < \mathcal{S}(Y;\hat{Z}) \leq t+s$, Algorithm 2.1 cannot produce Y as output. It remains to be proven that under conditions (5) and (6) no codeword different from Y may result as output of the algorithm.
Assume first that there is an $i \geq 0$ and a D such that $D \subseteq \{x_{r-i-t}, x_{r-i-(t-1)}, \ldots, x_{r-i-1}\}$ and $Z_{r-i} - D \in \mathcal{C}$, with $Z_{r-i} - D \neq Y$. This means, a codeword $Z_{r-i} - D$ is produced as output of the algorithm either when x_r has arrived or before x_r has arrived. Certainly $i \leq t+s$; if not, $Z_{r-i} - D \subseteq Y$ contradicting conditions (5) and (6) (mainly, the code must be unordered).
Let

$$\begin{aligned} D_1 &= D \cap \{x_{r-i-t}, x_{r-i-t+1}, \ldots, x_{r-t-s-1}\} \\ D_2 &= D \cap \{x_{r-t-s}, x_{r-t-s+1}, \ldots, x_{r-i-1}\} \end{aligned}$$

Therefore, $D_1 = \emptyset$ or $|D_1| \leq i-s$ if $s < i$, and $|D_2| \leq t+s-i$.
Notice that, since $Y - (Z_{r-i} - D) \subseteq D \cup \{x_{r-(i-1)}, x_{r-(i-2)}, \ldots, x_r\}$, then

$$\begin{aligned} N(Y, Z_{r-i} - D) &\leq i + |D_1| + |D_2| \\ &\leq i + \max\{0, i-s\} + |D_2| \end{aligned} \tag{15}$$

and, since $(Z_{r-i} - D) - Y \subseteq \{x_{r-t-s}, x_{r-t-s+1)}, \ldots, x_{r-i}\} - D_2$, we have

$$N(Z_{r-i} - D, Y) \leq 1 + t + s - i - |D_2|. \tag{16}$$

Now, let

$$t + s - i - |D_2| = j. \tag{17}$$

Notice that $j \geq 0$. We have two cases: $0 \leq j \leq t$ and $j > t$.
Assume first that $0 \leq j \leq t$. Then, by (17), $i - s + |D_2| \geq 0$, and by (15) and (17), we have

$$\begin{aligned} N(Y, Z_{r-i} - D) &\leq i + \max\{0, i-s\} + |D_2| \\ &\leq i + (i - s + |D_2|) + |D_2| \\ &= 2(t-j) + s \end{aligned} \tag{18}$$

By (16), (17) and (18), we have $N(Z_{r-i} - D, Y) \leq j + 1$ and $N(Y, Z_{r-i} - D) \leq 2(t-j) + s$, where $0 \leq j \leq t$, contradicting (5) and (6) by Lemma 5.1.
Assume next that $j > t$. In particular, by (17), this implies that $i - s < 0$ and, by (15) and (17),

$$N(Y, Z_{r-i} - D) \leq i + |D_2| = s - (j - t). \tag{19}$$

Now, by (16), (17) and (19), we have $N(Z_{r-i} - D, Y) \leq j+1$ and $N(Y, Z_{r-i} - D) \leq t+s-j$, where $j > t$, contradicting (5) and (6) by Lemma 5.2.
Finally, assume that the algorithm produces a codeword *after* Y has arrived. For this to happen the skew must exceed t, i.e., $t < \mathcal{S}(Y; \hat{Z}) \leq t + s$. Otherwise, Algorithm 2.1 would have produced Y, and, by the first part, would have not produced any other codeword. Thus, there is an $i > 0$ and a D such that $D \subseteq \{x_{r+i-t}, x_{r+i-(t-1)}, \ldots, x_{r+i-1}\}$ and $Z_{r+i} - D \in \mathcal{C}$, with $Z_{r+i} - D \neq Y$. Certainly $i \leq t$; if not, $Y \subseteq Z_{r+i} - D$ contradicting conditions (5) and (6).
Let $D_1 = D \cap \{x_{r+i-t}, x_{r+i-t+1}, \ldots, x_r\}$. In particular, $|D_1| \leq t - i + 1$. Since $Y - (Z_{r+i} - D) \subseteq D_1$, then

$$N(Y, Z_{r+i} - D) \leq |D_1| \tag{20}$$

and, since

$$\begin{aligned} (Z_{r+i} - D) - Y \subseteq\ & \{x_{r-s-t}, x_{r-s-t+1}, \ldots, x_{r+i-t-1}\} \cup (\{x_{r+i-t}, x_{r+i-t+1}, \ldots, x_r\} - D_1) \cup \\ & \{x_{r+1}, x_{r+2}, \ldots, x_{r+i}\}, \end{aligned}$$

we have

$$\begin{aligned} N(Z_{r+i} - D, Y) &\leq (i+s) + (1 + t - i - |D_1|) + i \\ &= 2(t - (|D_1| - 1)) + s - (t - i + 1 - |D_1|) \\ &\leq 2(t - (|D_1| - 1)) + s, \end{aligned} \tag{21}$$

the last inequality since $t - i + 1 - |D_1| \geq 0$. Making $j = |D_1| - 1$ (notice that $0 \leq j \leq t$), we have, by (20) and (21), $N(Y, Z_{r+i} - D) \leq j + 1$ and $N(Z_{r+i} - D, Y) \leq 2(t-j) + s$. By Lemma 5.1, this contradicts conditions (5) and (6), completing the proof. □

References

[1] J. M. Berger, "A note on error detecting codes for asymmetric channels," Information and Control, Vol. 4, pp. 68-73, March 1961.

[2] M. Blaum and J. Bruck, "Coding for Skew-Tolerant Parallel Asynchronous Communications," IEEE Trans. on Information Theory, Vol. IT-39, No. 2, pp. 379-388, March 1993.

[3] M. Blaum and J. Bruck, "Unordered error-correcting codes and their applications," Proceedings FTCS-22, Boston, pp. 486-493, July 1992.

[4] M. Blaum and J. Bruck, "Delay-Insensitive Pipelined Communication on Parallel Buses," IEEE Trans. on Computers, Vol. C-44, No. 5, pp. 660-68, May 1995.

[5] M. Blaum, J. Bruck and L. H. Khachatrian, "Construction of skew-tolerant and skew-detecting codes," IEEE Trans. on Information Theory, vol. IT-39, pp. 1751-1757, Sept. 1993.

[6] T. Verhoeff, "Delay-insensitive codes - an overview," Distributed Computing, 3:1-8, 1988.