Necessary and Sufficient Conditions on Block Codes Correcting/Detecting Errors of Various Types

J. H. Weber, C. de Vroedt, and D. E. Boekee

Abstract—**Necessary and sufficient conditions are given for a block code to be capable of correcting up to t_1 symmetric errors, up to t_2 unidirectional errors, and up to t_3 asymmetric errors, as well as detecting from $t_1 + 1$ up to d_1 symmetric errors that are not of the unidirectional type, from $t_2 + 1$ up to d_2 unidirectional errors that are not of the asymmetric type, and from $t_3 + 1$ up to d_3 asymmetric errors. Many known conditions on block codes concerning error correction and/or detection appear as special cases of this general result. Further, some codes turn out to have stronger error correcting/detecting capabilities than they were originally designed for.**

Index Terms—**Asymmetric errors, error correction, error detection, symmetric errors, unidirectional errors.**

I. Introduction

We consider discrete channels with an (input and output) alphabet $\mathcal{A} = \{0, 1, \cdots, a-1\}(a \geq 2)$. For reliable transmission of data over such a channel we can use a block code C of length n over the alphabet $\mathcal{A}$, i.e., $C \subseteq \mathcal{A}^n$. Most block codes have been designed to correct and/or detect errors with a symmetric nature, such as the errors caused by the *a-ary symmetric channel* (cf. [7]), on which $P(y \mid x) = \epsilon$ if $x \neq y$ and $P(y \mid x) = 1 - (a-1)\epsilon$ if $x = y$ (with $0 \leq \epsilon \leq 1/(a-1)$), where $P(y|x)$ denotes the probability of receiving the symbol $y \in \mathcal{A}$ when the symbol $x \in \mathcal{A}$ is sent. However, in some applications, such as optical communications, the errors have a highly asymmetric nature. Channels causing this kind of error can often be modeled by the *a-ary asymmetric channel* (cf. [4]), on which $P(y|x) = 0$ for all $y > x$. Further, in some recently developed memory systems, the errors appear to be of a unidirectional nature. These memory systems can be modeled by an *a-ary unidirectional channel* (cf. [1]), that behaves for a certain codeword either like the a-ary asymmetric channel or like the inverted a-ary asymmetric channel, on which $P(y|x) = 0$ for all $y < x$.

Based on the preceding statements, we shall now formally define the error types that will be considered in this paper. First we define

$$N(\underline{u}, \underline{v}) = |\{i | u_i < v_i\}|,$$
$$d(\underline{u}, \underline{v}) = N(\underline{u}, \underline{v}) + N(\underline{v}, \underline{u}) = |\{i | u_i \neq v_i\}| \text{ (Hamming distance)},$$

for $\underline{u} = (u_1, u_2, \cdots, u_n), \underline{v} = (v_1, v_2, \cdots, v_n) \in \mathcal{A}^n$. The vector $\underline{u}$ is said to cover the vector $\underline{v}$ ($\underline{u} \geq \underline{v}$) if $N(\underline{u}, \underline{v}) = 0$. When sending a codeword $\underline{c} \in C$ and receiving a vector $\underline{y} \in \mathcal{A}^n$, we say that $\underline{c}$ has suffered t *asymmetric errors* if $\underline{c}$ covers $\underline{y}$ and $d(\underline{c}, \underline{y}) = t$, that $\underline{c}$ has suffered t *unidirectional errors* if $\underline{c}$ covers or is covered by $\underline{y}$ and $d(\underline{c}, \underline{y}) = t$, and that $\underline{u}$ has suffered t *symmetric errors* if $d(\underline{c}, \underline{y}) = t$.

In accordance with the three error types, we define three kinds of spheres with radius r for each $\underline{x} \in \mathcal{A}^n$:

$$S_{Sy}(\underline{x}, r) = \{\underline{y} \in \mathcal{A}^n \mid d(\underline{x}, \underline{y}) \leq r\},$$
$$S_U(\underline{x}, r) = \{\underline{y} \in \mathcal{A}^n \mid d(\underline{x}, \underline{y}) \leq r \wedge (\underline{x} \geq \underline{y} \vee \underline{y} \geq \underline{x})\},$$
$$S_{As}(\underline{x}, r) = \{\underline{y} \in \mathcal{A}^n \mid d(\underline{x}, \underline{y}) \leq r \wedge \underline{x} \geq \underline{y}\}.$$

For the sake of convenience we also define a super-sphere

$$S(\underline{x}, r_1, r_2, r_3) = S_{Sy}(\underline{x}, r_1) \cup S_U(\underline{x}, r_2) \cup S_{As}(\underline{x}, r_3)$$

for each $\underline{x} \in \mathcal{A}^n$ and $0 \leq r_1 \leq r_2 \leq r_3$. Each sphere $S_X(\underline{c}, t)$ contains the vectors that can be received when codeword $\underline{c}$ is sent suffering t or less errors of type X (with X =Sy(mmetric), X =U(nidirectional), or X =As(ymmetric), respectively). Hence we say that a code C can *correct* up to t errors of type X if the spheres $S_X(\underline{c}, t)$ are disjoint for any two distinct codewords. On the other hand, we say that a code can *detect* up to d errors of type X if the sphere $S_X(\underline{c}, d)$ does not contain other codewords than $\underline{c}$ for all $\underline{c} \in C$. Necessary and sufficient conditions are known for a code to be capable of correcting or detecting errors of each of the three types. But sometimes a combination of correction and detection is required, or even simultaneous correction and/or detection of errors of various types. For example, some authors (see, e.g., [1], [2], [9]) have considered codes correcting up to t symmetric errors and detecting all ($t + 1$ or more) unidirectional errors, since it was observed that in some memory systems the number of unidirectional errors can be unlimited, whereas the number of symmetric errors is limited with high probability. A necessary and sufficient condition for this case was derived in [2]. To be able to deal with such cases it is interesting to look for necessary and sufficient conditions for all combinations of correction and detection for the three error types considered here.

We call a code t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED $(0 \leq t_1 \leq t_2 \leq t_3, 0 \leq d_1 \leq d_2 \leq d_3, t_i \leq d_i)$ if it can correct up to t_1 symmetric errors, up to t_2 unidirectional errors, and up to t_3 asymmetric errors, as well as detect from $t_1 + 1$ up to d_1 symmetric errors that are not of the unidirectional type, from $t_2 + 1$ up to d_2 unidirectional errors that are not of the asymmetric type, and from $t_3 + 1$ up to d_3 asymmetric errors. In the context of the spheres this means that

$$S(\underline{x}, t_1, t_2, t_3) \cap S(\underline{y}, d_1, d_2, d_3) = \varnothing$$

for any two distinct codewords $\underline{x}$ and $\underline{y}$.

In Section II we derive necessary and sufficient conditions for a code to be t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED. Hence we can obtain necessary and sufficient conditions for correction and/or detection of any combination of symmetric/unidirectional/asymmetric errors by making appropriate choices for t_i and d_i. Some important cases are considered in Section III. Many existing results appear as special cases of the general result. In Section IV we pay attention to another error criterion. Finally, concluding remarks are found in Section V.

II. General Conditions

In literature (see, e.g., [2], [3], [5], [6], [8], [10]) various necessary and sufficient conditions were derived on block codes to have certain error correcting and/or detecting capabilities. Since in each derivation the same kinds of techniques were used, we have tried to obtain general conditions covering all combinations of symmetric, unidirectional, and asymmetric errors. The result is given in Theorem 1.

Manuscript received September 10, 1990; revised September 16, 1991.

J. H. Weber and D. E. Boekee are with the Department of Electrical Engineering, Information Theory Group, Delft University of Technology, 2600 GA Delft, The Netherlands.

C. de Vroedt is with the Department of Mathematics and Informatics, Delft University of Technology, 2600 AJ Delft, The Netherlands.

IEEE Log Number 9103100.

0018–9340/92$03.00 © 1992 IEEE

Theorem 1: A code C is t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED (with $0 \le t_1 \le t_2 \le t_3, 0 \le d_1 \le d_2 \le d_3, t_i \le d_i$) if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a},\underline{b}) \ge N(\underline{b},\underline{a})$ satisfy

$$\begin{cases} d(\underline{a},\underline{b}) \ge t_2 + d_3 + 1 \wedge d(\underline{a},\underline{b}) \\ \quad \ge t_3 + d_2 + 1 & \text{if } N(\underline{b},\underline{a}) = 0 \\ d(\underline{a},\underline{b}) \ge t_3 + d_1 + 1 \wedge d(\underline{a},\underline{b}) \\ \quad \ge t_1 + d_3 + 1 \wedge N(\underline{a},\underline{b}) \ge d_3 + 1 & \text{if } 1 \le N(\underline{b},\underline{a}) \le t_3 \\ d(\underline{a},\underline{b}) \ge t_3 + d_1 + 1 & \text{if } N(\underline{b},\underline{a}) \ge t_3 + 1. \end{cases}$$

Proof: From the definition of a t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED code it follows that we have to prove that for all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a},\underline{b}) \ge N(\underline{b},\underline{a})$:

$$S(\underline{a}, t_1, t_2, t_3) \cap S(\underline{b}, d_1, d_2, d_3) = \varnothing$$

$$\text{and } S(\underline{b}, t_1, t_2, t_3) \cap S(\underline{a}, d_1, d_2, d_3) = \varnothing$$

$$\Leftrightarrow$$

$$\begin{cases} d(\underline{a},\underline{b}) \ge t_2 + d_3 + 1 \wedge d(\underline{a},\underline{b}) \\ \quad \ge t_3 + d_2 + 1 & \text{if } N(\underline{b},\underline{a}) = 0 \\ d(\underline{a},\underline{b}) \ge t_3 + d_1 + 1 \wedge d(\underline{a},\underline{b}) \\ \quad \ge t_1 + d_3 + 1 \wedge N(\underline{a},\underline{b}) \ge d_3 + 1 & \text{if } 1 \le N(\underline{b},\underline{a}) \le t_3 \\ d(\underline{a},\underline{b}) \ge t_3 + d_1 + 1 & \text{if } N(\underline{b},\underline{a}) \ge t_3 + 1. \end{cases}$$

Without loss of generality we may assume that $\underline{a}$ and $\underline{b}$ satisfy

$$\begin{cases} a_i = b_i & \text{for } 1 \le i \le \alpha \\ a_i > b_i & \text{for } \alpha + 1 \le i \le \alpha + \beta \\ a_i < b_i & \text{for } \alpha + \beta + 1 \le i \le \alpha + \beta + \gamma = n \end{cases}$$

with $\alpha = |\{i \mid a_i = b_i\}|$, $\beta = |\{i \mid a_i > b_i\}|$, and $\gamma = |\{i \mid a_i < b_i\}|$.

"$\Rightarrow$" Define $\underline{z} \in \mathcal{A}^n$ as

$$\begin{cases} z_i = a_i & \text{for } 1 \le i \le \alpha \\ z_i = b_i & \text{for } \alpha + 1 \le i \le \alpha + \beta \\ z_i = a_i & \text{for } \alpha + \beta + 1 \le i \le \alpha + \beta + \mu \\ z_i = b_i & \text{for } \alpha + \beta + \mu + 1 \le i \le \alpha + \beta + \gamma = n \end{cases}$$

where μ will be filled in in accordance with the case under consideration. We next consider three different cases.

1) The case $N(\underline{b},\underline{a}) = 0$. Suppose $d(\underline{a},\underline{b}) \le t_2 + d_3$ or $d(\underline{a},\underline{b}) \le t_3 + d_2$.

   i) If $d(\underline{a},\underline{b}) \le t_2 + d_3$, then $\underline{z}$(with $\mu = \min\{N(\underline{a},\underline{b}), d_3\}$) $\in S_U(\underline{a}, t_2) \cap S_{As}(\underline{b}, d_3)$.

   ii) If $d(\underline{a},\underline{b}) \le t_3 + d_2$, then $\underline{z}$(with $\mu = \min\{N(\underline{a},\underline{b}), t_3\}$) $\in S_U(\underline{a}, d_2) \cap S_{As}(\underline{b}, t_3)$.

2) The case $1 \le N(\underline{b},\underline{a}) \le t_3$. Suppose $d(\underline{a},\underline{b}) \le t_3 + d_1$ or $d(\underline{a},\underline{b}) \le t_1 + d_3$ or $N(\underline{a},\underline{b}) \le d_3$.

   i) If $d(\underline{a},\underline{b}) \le t_3 + d_1$ and $t_3 \le N(\underline{a},\underline{b})$, then $\underline{z}$(with $\mu = t_3$)$\in S_{Sy}(\underline{a}, d_1) \cap S_{As}(\underline{b}, t_3)$.

   ii) If $d(\underline{a},\underline{b}) \le t_3 + d_1$ and $t_3 > N(\underline{a},\underline{b})$, then $\underline{z}$(with $\mu = N(\underline{a},\underline{b})) \in S_{As}(\underline{a}, t_3) \cap S_{As}(\underline{b}, t_3)$.

   iii) If $d(\underline{a},\underline{b}) \le t_1 + d_3$ and $d_3 \le N(\underline{a},\underline{b})$, then $\underline{z}$(with $\mu = d_3$)$\in S_{Sy}(\underline{a}, t_1) \cap S_{As}(\underline{b}, d_3)$.

   iv) If $d(\underline{a},\underline{b}) \le t_1 + d_3$ and $d_3 > N(\underline{a},\underline{b})$, then $\underline{z}$(with $\mu = N(\underline{a},\underline{b})$)$\in S_{As}(\underline{a}, t_3) \cap S_{As}(\underline{b}, d_3)$.

   v) If $N(\underline{a},\underline{b}) \le d_3$, then $\underline{z}$(with $\mu = N(\underline{a},\underline{b})$)$\in S_{As}(\underline{a}, t_3) \cap S_{As}(\underline{b}, d_3)$.

3) The case $N(\underline{b},\underline{a}) \ge t_3 + 1$. Suppose $d(\underline{a},\underline{b}) \le t_3 + d_1$.

   i) If $d(\underline{a},\underline{b}) \le t_3 + d_1$, then $\underline{z}$(with $\mu = t_3$)$\in S_{Sy}(\underline{a}, d_1) \cap S_{As}(\underline{b}, t_3)$.

Hence we have shown that $\underline{z}$ is in $S(\underline{a}, t_1, t_2, t_3) \cap S(\underline{b}, d_1, d_2, d_3)$ or $S(\underline{b}, t_1, t_2, t_3) \cap S(\underline{a}, d_1, d_2, d_3)$ for each case, which contradicts the assumption that these two intersections of sets are both empty.

"$\Leftarrow$" Suppose there exists a $\underline{z} \in \mathcal{A}^n$ such that $\underline{z} \in S(\underline{a}, t_1, t_2, t_3) \cap S(\underline{b}, d_1, d_2, d_3)$ or $\underline{z} \in S(\underline{b}, t_1, t_2, t_3) \cap S(\underline{a}, d_1, d_2, d_3)$. Again, we shall find a contradiction for each case. This will only be shown for $\underline{z} \in S(\underline{a}, t_1, t_2, t_3) \cap S(\underline{b}, d_1, d_2, d_3)$, since it can be shown in a completely analogous way for $\underline{z} \in S(\underline{b}, t_1, t_2, t_3) \cap S(\underline{a}, d_1, d_2, d_3)$. We again consider three cases.

1) The case $N(\underline{b},\underline{a}) = 0$.

   i) If $N(\underline{a},\underline{z}) = 0$, then $d(\underline{a},\underline{b}) = N(\underline{a},\underline{b}) \le N(\underline{z},\underline{b}) \le d(\underline{z},\underline{b}) \le d_3$.

   ii) If $N(\underline{a},\underline{z}) \ge 1$, then $d(\underline{a},\underline{b}) \le d(\underline{a},\underline{z}) + d(\underline{z},\underline{b}) \le t_2 + d_3$.

2) The case $1 \le N(\underline{b},\underline{a}) \le t_3$.

   i) If $N(\underline{a},\underline{z}) \ge 1$ and $N(\underline{z},\underline{a}) \ge 1$, then $d(\underline{a},\underline{b}) \le d(\underline{a},\underline{z}) + d(\underline{z},\underline{b}) \le t_1 + d_3$.

   ii) If $N(\underline{a},\underline{z}) = 0$, then $N(\underline{a},\underline{b}) \le N(\underline{z},\underline{b}) \le d(\underline{z},\underline{b}) \le d_3$.

   iii) If $N(\underline{z},\underline{a}) = 0$ and $N(\underline{z},\underline{b}) \ge 1$ and $N(\underline{b},\underline{z}) \ge 1$, then $d(\underline{a},\underline{b}) \le d(\underline{a},\underline{z}) + d(\underline{z},\underline{b}) \le t_2 + d_1$.

   iv) If $N(\underline{z},\underline{a}) = 0$ and $N(\underline{z},\underline{b}) = 0$, then $N(\underline{a},\underline{b}) \le N(\underline{a},\underline{z}) \le d(\underline{a},\underline{z}) \le t_2$.

   v) If $N(\underline{z},\underline{a}) = 0$ and $N(\underline{b},\underline{z}) = 0$, then $N(\underline{b},\underline{a}) \le N(\underline{b},\underline{z}) = 0$.

3) The case $N(\underline{b},\underline{a}) \ge t_3 + 1$.

   i) If $N(\underline{b},\underline{z}) \ge 1$ and $N(\underline{z},\underline{b}) \ge 1$, then $d(\underline{a},\underline{b}) \le d(\underline{a},\underline{z}) + d(\underline{z},\underline{b}) \le t_3 + d_1$.

   ii) If $N(\underline{z},\underline{b}) = 0$, then $N(\underline{b},\underline{a}) \le N(\underline{a},\underline{b}) \le N(\underline{a},\underline{z}) \le d(\underline{a},\underline{z}) \le t_3$.

   iii) If $N(\underline{b},\underline{z}) = 0$, then $N(\underline{b},\underline{a}) \le N(\underline{z},\underline{a}) \le d(\underline{z},\underline{a}) \le t_3$. □

Sometimes a code turns out to have stronger error correcting/detecting capabilities than it was originally designed for, as can be seen from the next theorem.

Theorem 2: Any t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED code (with $0 \le t_1 \le t_2 \le t_3, 0 \le d_1 \le d_2 \le d_3, t_i \le d_i$) is also a t_1'-SyEC t_2'-UEC t_3'-AsEC d_1'-SyED d_2'-UED d_3'-AsED code with

$$\begin{aligned} t_1' &= \max\{t_1, t_3 + d_1 - d_3\}, \\ t_2' &= \max\{t_2, t_3 + d_2 - d_3\}, t_3' = t_3, \\ d_1' &= \max\{d_1, \min\{t_3 + 1, t_1 + d_3 - t_3\}\}, \\ d_2' &= \max\{d_2, t_2 + d_3 - t_3\}, d_3' = d_3. \end{aligned}$$

Proof: First, observe that $t_2' + d_3 = t_3 + d_2'$. Next, since

$$\begin{aligned} & 0 \le t_1 \le t_1' = \max\{t_1, t_3 + d_1 - d_3\} \le \max\{t_2, t_2' - d_2' + d_1\} \\ & \quad \le t_2' = \max\{t_2, t_3 + d_2 - d_3\} \le t_3 = t_3', \\ & 0 \le d_1 \le d_1' \le \max\{d_1, d_3 + t_1 - t_3\} \le \max\{d_2, d_2' - t_2' + t_1\} \\ & \quad \le d_2' = \max\{d_2, d_3 + t_2 - t_3\} \le d_3 = d_3', \\ & t_1' = \max\{t_1, t_3 + d_1 - d_3\} \le d_1 \le d_1', t_2' \\ & \quad = \max\{t_2, t_3 + d_2 - d_3\} \le d_2 \le d_2', t_3' = t_3 \le d_3 = d_3', \end{aligned}$$

we may apply Theorem 1 to obtain necessary and sufficient conditions for a code to be t_1'-SyEC t_2'-UEC t_3'-AsEC d_1'-SyED d_2'-UED d_3'-AsED. Finally, we show that these conditions are implied by the necessary and sufficient conditions for a code to be t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED.

1) The case $N(\underline{b}, \underline{a}) = 0$.

   i) If $t_3 + d_2 \geq t_2 + d_3$, then $d(\underline{a}, \underline{b}) \geq t_3 + d_2 + 1 = t'_3 + d'_2 + 1 = t'_2 + d'_3 + 1$.

   ii) If $t_3 + d_2 < t_2 + d_3$, then $d(\underline{a}, \underline{b}) \geq t_2 + d_3 + 1 = t'_2 + d'_3 + 1 = t'_3 + d'_2 + 1$.

2) The case $1 \leq N(\underline{b}, \underline{a}) \leq t_3$.

   i) If $t_3 + d_1 \geq t_1 + d_3$, then $d(\underline{a}, \underline{b}) \geq t_3 + d_1 + 1 = t'_3 + d'_1 + 1 = t'_1 + d'_3 + 1$ and $N(\underline{a}, \underline{b}) \geq d_3 + 1 = d'_3 + 1$.

   ii) If $t_3 + d_1 < t_1 + d_3$, then $d(\underline{a}, \underline{b}) \geq t_1 + d_3 + 1 = t'_1 + d'_3 + 1 \geq t'_3 + d'_1 + 1$ and $N(\underline{a}, \underline{b}) \geq d_3 + 1 = d'_3 + 1$.

3) The case $N(\underline{b}, \underline{a}) \geq t_3 + 1$.

   i) If $t_3 + 1 \leq d_1$ or $t_1 + d_3 \leq t_3 + d_1$, then $d(\underline{a}, \underline{b}) \geq t_3 + d_1 + 1 = t'_3 + d'_1 + 1$.

   ii) If $t_3 + 1 > d_1$ and $t_1 + d_3 > t_3 + d_1$, then $d(\underline{a}, \underline{b}) \geq 2t_3 + 2 \geq t'_3 + d'_1 + 1$. □

III. Special Cases

In this section we consider some important t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED codes (with $0 \leq t_1 \leq t_2 \leq t_3, 0 \leq d_1 \leq d_2 \leq d_3, t_i \leq d_i$). Many known results on error correcting/detecting codes will appear as special cases of the general theorem, and also some interesting new results will show up.

If we want to restrict ourselves to *correction* only, we substitute $d_i = t_i (i = 1, 2, 3)$ into Theorem 1. Hence a code C is t_1-SyEC t_2-UEC t_3-AsEC if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$ satisfy

$$\begin{cases} d(\underline{a}, \underline{b}) \geq t_2 + t_3 + 1 & \text{if } N(\underline{b}, \underline{a}) = 0 \\ d(\underline{a}, \underline{b}) \geq t_1 + t_3 + 1 \wedge N(\underline{a}, \underline{b}) \geq t_3 + 1 & \text{if } N(\underline{b}, \underline{a}) \geq 1. \end{cases}$$

If we want to restrict ourselves to *detection* only, we substitute $t_i = 0 (i = 1, 2, 3)$ into Theorem 1. Hence a code C is d_1-SyED d_2-UED d_3-AsED if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$ satisfy

$$\begin{cases} d(\underline{a}, \underline{b}) \geq d_3 + 1 & \text{if } N(\underline{b}, \underline{a}) = 0 \\ d(\underline{a}, \underline{b}) \geq d_1 + 1 & \text{if } N(\underline{b}, \underline{a}) \geq 1. \end{cases}$$

From this (and also from Theorem 2) it follows that any d_1-SyED d_2-UED d_3-AsED code is also a d_1-SyED d_3-UED d_3-AsED code. Thus we only have to consider d_1-SyED d_2-UED codes.

When considering combinations of *correction and detection*, it is interesting (also from a practical point of view) to look at t_1-SyEC t_2-UEC d_2-UED codes. By substituting $t_3 = t_2, d_1 = t_1$, and $d_3 = d_2$ into Theorem 1, we find that a code C is t_1-SyEC t_2-UEC d_2-UED if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$ satisfy

$$\begin{cases} d(\underline{a}, \underline{b}) \geq t_2 + d_2 + 1 & \text{if } N(\underline{b}, \underline{a}) = 0 \\ d(\underline{a}, \underline{b}) \geq t_1 + d_2 + 1 & \\ \quad \wedge N(\underline{a}, \underline{b}) \geq d_2 + 1 & \text{if } 1 \leq N(\underline{b}, \underline{a}) \leq t_2. \end{cases}$$

By substituting $t_1 = 0, t_2 = t$, and $d_2 = d$ into the previous result, we find that a code C is t-UEC d-UED if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$ satisfy

$$d(\underline{a}, \underline{b}) \geq t + d + 1 \vee (N(\underline{b}, \underline{a}) \geq 1 \wedge N(\underline{a}, \underline{b}) \geq d + 1) \vee N(\underline{b}, \underline{a}) \geq t + 1.$$

Note that this is a correction of a result presented earlier by Lin and Bose in [5] (Theorem 2.1), in which they claim that a binary ($a = 2$) code C is t-UEC d-UED if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$ satisfy

$$d(\underline{a}, \underline{b}) \geq t + d + 1 \vee N(\underline{b}, \underline{a}) \geq t + 1.$$

This condition is indeed sufficient for a code to be t-UEC d-UED, but *not* necessary. The latter can be seen from the code $C = \{10000, 01111\}$ which is 2-UEC 3-UED, but does not satisfy the condition of Lin and Bose.

Next we consider codes *detecting all errors* of a certain type. To this end we first investigate t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED AAsED (all asymmetric error detecting) codes. By substituting $d_3 = n$ into Theorem 1, we find that a code C is t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED AAsED if and only if all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$ satisfy

$$d(\underline{a}, \underline{b}) \geq t_3 + d_1 + 1 \wedge N(\underline{b}, \underline{a}) \geq t_3 + 1.$$

From this it follows that any t_1-SyEC t_2-UEC t_3-AsEC d_1-SyED d_2-UED AAsED code is also a t_3-SyEC t_3-UEC t_3-AsEC ($\max\{d_1, t_3 + 1\}$)-SyED AUED AAsED code. Hence we can restrict ourselves to t-SyEC d-SyED AUED codes, which are characterized by the necessary and sufficient condition

$$d(\underline{a}, \underline{b}) \geq t + d + 1 \wedge N(\underline{b}, \underline{a}) \geq t + 1$$

for all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$. Note that any t-SyEC AUED code is also a t-SyEC $(t + 1)$-SyED AUED code.

Finally, the results of this section are summarized in Table I, which also contains a few other interesting results. In this table, all conditions for a code C to have the denoted error correcting/detecting capability must hold for all $\underline{a}, \underline{b} \in C$ with $\underline{a} \neq \underline{b}$ and $N(\underline{a}, \underline{b}) \geq N(\underline{b}, \underline{a})$. The values of t_i and d_i denote the substitutions that must be made to derive the relevant condition from Theorem 1.

IV. Another Error Criterion

In this paper the number of errors when sending a codeword $\underline{c}$ and receiving a vector $\underline{y}$ has been defined to be the number of coordinates in which the two vectors differ:

$$|\{i \mid c_i \neq y_i\}|.$$

How much these coordinates differ is not important in this definition. If one wishes to take into account the magnitude of each symbol error, a suitable and widely used (cf. [4]) definition for the number of errors is

$$\sum |c_i - y_i|.$$

Note that there is no difference between these two definitions in the binary case ($a = 2$).

All the results on t_1-SyEC t_2 -UEC t_3-AsEC d_1-SyED d_2-UED d_3-AsED codes derived in this paper while counting the number of errors as $|\{i|c_i \neq y_i\}|$ are also valid when counting the number of errors as $\sum |c_i - y_i|$, if we adapt the definitions of $N(\underline{a}, \underline{b})$ and $d(\underline{a}, \underline{b})$ for the latter case as follows:

$$N(\underline{a}, \underline{b}) = \sum \max\{b_i - a_i, 0\},$$
$$d(\underline{a}, \underline{b}) = N(\underline{a}, \underline{b}) + N(\underline{b}, \underline{a}) = \sum |b_i - a_i|.$$

Another thing we must adapt is the vector $\underline{z}$ in the "⇒" part of the

TABLE I
NECESSARY AND SUFFICIENT CONDITIONS ON ERROR CORRECTING/DETECTING CODES

EC/ED capability	t_1	t_2	t_3	d_1	d_2	d_3	condition
t-SyEC	t	t	t	t	t	t	$d(\underline{a},\underline{b})\geq 2t+1$
t-UEC	0	t	t	0	t	t	$d(\underline{a},\underline{b})\geq 2t+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge N(\underline{a},\underline{b})\geq t+1)$
t-AsEC	0	0	t	0	0	t	$N(\underline{a},\underline{b})\geq t+1$
t_1-SyEC t_2-UEC	t_1	t_2	t_2	t_1	t_2	t_2	$d(\underline{a},\underline{b})\geq 2t_2+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge N(\underline{a},\underline{b})\geq t_2+1 \wedge$ $d(\underline{a},\underline{b})\geq t_1+t_2+1)$
t_1-SyEC t_3-AsEC	t_1	t_1	t_3	t_1	t_1	t_3	$d(\underline{a},\underline{b})\geq t_1+t_3+1 \wedge N(\underline{a},\underline{b})\geq t_3+1$
t_2-UEC t_3-AsEC	0	t_2	t_3	0	t_2	t_3	$N(\underline{a},\underline{b})\geq t_2+t_3+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge N(\underline{a},\underline{b})\geq t_3+1)$
t_1-SyEC t_2-UEC t_3-AsEC	t_1	t_2	t_3	t_1	t_2	t_3	$N(\underline{a},\underline{b})\geq t_2+t_3+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge N(\underline{a},\underline{b})\geq t_3+1 \wedge$ $d(\underline{a},\underline{b})\geq t_1+t_3+1)$
d-SyED	0	0	0	d	d	d	$d(\underline{a},\underline{b})\geq d+1$
d-UED	0	0	0	0	d	d	$d(\underline{a},\underline{b})\geq d+1 \vee N(\underline{b},\underline{a})\geq 1$
d_1-SyED d_2-UED	0	0	0	d_1	d_2	d_2	$d(\underline{a},\underline{b})\geq d_2+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge d(\underline{a},\underline{b})\geq d_1+1)$
t-SyEC d-SyED	t	t	t	d	d	d	$d(\underline{a},\underline{b})\geq t+d+1$
t-SyEC d-UED	t	t	t	t	d	d	$d(\underline{a},\underline{b})\geq t+d+1 \vee N(\underline{b},\underline{a})\geq t+1$
t-UEC d-UED	0	t	t	0	d	d	$d(\underline{a},\underline{b})\geq t+d+1 \vee N(\underline{b},\underline{a})\geq t+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge N(\underline{a},\underline{b})\geq d+1)$
t_1-SyEC t_2-UEC d-UED	t_1	t_2	t_2	t_1	d	d	$d(\underline{a},\underline{b})\geq t_2+d+1 \vee N(\underline{b},\underline{a})\geq t_2+1 \vee$ $(N(\underline{b},\underline{a})\geq 1 \wedge N(\underline{a},\underline{b})\geq d+1 \wedge$ $d(\underline{a},\underline{b})\geq t_1+d+1)$
t-AsEC d-AsED	0	0	t	0	0	d	$N(\underline{a},\underline{b})\geq d+1 \vee N(\underline{b},\underline{a})\geq t+1$
AUED	0	0	0	0	n	n	$N(\underline{b},\underline{a})\geq 1$
t-SyEC AUED	t	t	t	t	n	n	$N(\underline{b},\underline{a})\geq t+1$
d-SyED AUED	0	0	0	d	n	n	$N(\underline{b},\underline{a})\geq 1 \wedge d(\underline{a},\underline{b})\geq d+1$
t-SyEC d-SyED AUED	t	t	t	d	n	n	$N(\underline{b},\underline{a})\geq t+1 \wedge d(\underline{a},\underline{b})\geq t+d+1$

proof of Theorem 1:

$$\begin{cases} z_i = a_i & \text{for } 1 \leq i \leq \alpha \\ z_i = b_i & \text{for } \alpha+1 \leq i \leq \alpha+\beta \\ z_i = a_i & \text{for } \alpha+\beta+1 \leq i \leq \alpha+\beta+\mu \\ z_i = b_i & \\ \quad -(\lambda - \sum\limits_{k=\alpha+\beta+1}^{\alpha+\beta+\mu} \lvert b_k - a_k \rvert) & \text{for } i = \alpha+\beta+\mu+1 \\ & \quad \text{provided that } \mu \leq \gamma - 1 \\ z_i = b_i & \text{for } \alpha+\beta+\mu+2 \leq \\ & \quad i \leq \alpha+\beta+\gamma = n \end{cases}$$

with

$$\mu = \max\{j \mid 0 \leq j \leq \gamma \wedge \sum_{k=\alpha+\beta+1}^{\alpha+\beta+j} \lvert b_k - a_k \rvert \leq \lambda\}$$

and

$$\lambda = \begin{cases} t_3 & \text{in the cases 1) ii), 2) i), 3) i)} \\ d_3 & \text{in the cases 1) i), 2) iii),} \\ n(a-1) & \text{in the cases 2) ii), 2) iv), 2) v).} \end{cases}$$

Finally, in order to obtain the results on t-SyEC d-SyED AUED codes, d_2 and d_3 must be substituted by $n(a-1)$ in stead of n.

V. CONCLUSION

In this paper we have given general necessary and sufficient conditions for codes to correct and/or detect errors of three different types. This covers earlier results as well as new interesting results. Some codes turn out to have stronger error correcting/detecting capabilities than they were originally designed for. An extension has also been given to another error criterion.

REFERENCES

[1] B. Bose and D. K. Pradhan, "Optimal unidirectional error detecting/correcting codes," *IEEE Trans. Comput.*, vol. C-31, pp. 564–568, June 1982.

[2] B. Bose and T. R. N. Rao, "Theory of unidirectional error correcting/detecting codes," *IEEE Trans. Comput.*, vol. C-31, pp. 521–530, June 1982.

[3] S. D. Constantin and T. R. N. Rao, "On the theory of binary asymmetric error-correcting codes," *Inform. Contr.*, vol. 40, pp. 20–36, 1979.

[4] T. Kløve, "Error correcting codes for the asymmetric channel," Rep. 18-09-07-81, Dep. Mathematics, Univ. Bergen, July 1981.

[5] D. J. Lin and B. Bose, "Theory and design of t-error correcting and d ($d > t$) unidirectional error detecting (t-EC d-UED) codes," *IEEE Trans. Comput.*, vol. C-37, pp. 433–439, Apr. 1988.

[6] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam, The Netherlands: North-Holland, 1977.

[7] R. J. McEliece, *The Theory of Information and Coding.* Reading, MA: Addison-Wesley, 1977.

[8] D. Nikolos, "Theory and design of t-error correcting/d-error detecting ($d > t$) and all unidirectional error detecting codes," *IEEE Trans. Comput.*, vol. C-40, pp. 132–142, Feb. 1991.

[9] D. Nikolos, N. Gaitanis, and G. Philokyprou, "Systematic t-error correcting/all unidirectional error detecting codes," *IEEE Trans. Comput.*, vol. C-35, pp. 394–402, May 1986.

[10] D. K. Pradhan and S. M. Reddy, "Fault-tolerant failsafe logic networks," in *Proc. IEEE COMPCON*, San Francisco, CA, Mar. 1977, p. 363.

Detailed Modeling and Reliability Analysis of Fault-Tolerant Processor Arrays

N. Lopez-Benitez and J. A. B. Fortes

***Abstract*—A method for the generation of detailed models of fault-tolerant processor arrays, based on Stochastic Petri Nets (SPN) is presented in this paper. A compact SPN model of the array associates with each transition a set of attributes that includes a discrete probability distribution. Depending on the type of component and the reconfiguration scheme, these probabilities are determined using simulation or closed-form expressions and correspond to the survival of the array given that a number of components required by the reconfiguration process are faulty.**

Index Terms—**Fault-tolerance, Markov models, processor arrays, reliability, stochastic Petri nets.**

I. Introduction

As is the case with many systems, Markov models can be used to evaluate the reliability of processor arrays. However, reliability estimations are mostly based on the failures of processing elements only [1]. Components other than processing elements become very important in the analysis of fault-tolerant processor arrays because of their susceptibility to faults and the added hardware complexity of the overall array. This fact has played an important role in the derivation of the mathematical framework developed by Koren *et al.* [2] to evaluate yield improvement and performance-related measures of different array architectures. A detailed modeling of fault-tolerant processor arrays, which explicitly takes into consideration the failure statistics of each component as well as their possible interdependencies, entails not only an explosive growth in the model state space but also a difficult model construction process. This paper proposes a systematic method to construct Markov models for evaluating the reliability of processor arrays. The method is based on the premise that the fault behavior of a processor array can be modeled by a Stochastic Petri Net (SPN). However, in order to obtain a more compact and efficient representation, a set of attributes is associated with each transition in the Petri net model. This set of attributes allows the construction of the corresponding Markov model as the reachability graph is being generated. Included in these attributes is a discrete probability distribution such that the effect of faulty spares in the reconfiguration algorithm is captured each time a configuration change occurs. This distribution includes the probabilities of survival given that a number of components required by the reconfiguration process are faulty. Depending on the type of component and the reconfiguration scheme, these distributions are determined using simulation or closed-form expressions. The application of this method is illustrated by the detailed analysis of the SRE reconfiguration scheme [3]; also, analytical expressions to estimate probabilities of survival are derived for this scheme. Other applications that include schemes such as ARCE (Alternate Row-Column Elimination) [3] and DR (Direct Reconfiguration) [4], are reported in [5].

Once the Petri net model and the corresponding reachability graph have been obtained, all the information required to build the transition matrix of the corresponding Markov chain is available. Reliability evaluation tools such as ARIES [6] and SHARPE [7] can be used to evaluate the models developed here.

The second section of this paper discusses some basic notation and concepts which include array configurations and Petri nets; also, the large state space inherent in the models generated is illustrated. The third section discusses MSPN models as an extension of SPN's. Throughout these two sections the SRE reconfiguration scheme is used as an application. In the fourth section a procedure used in generating the reachability graph is described. Finally, results on reliability analysis are reported in Section V.

II. Preliminaries

A. Array Configurations

To analyze a fault-tolerant array architecture with k types of components, the configuration of an array is represented as a k-tuple:

$$C_i = (\eta_{1i}, \eta_{2i}, \cdots, \eta_{ki}) \qquad i = 0, 1, \cdots, |C|$$

where η_{li} denotes the number of elements of component type l and C is the set of all possible configurations of the array. Examples of component types include processing elements, links, switches, spare links, and spare processing elements. The occurrence of faults and the application of the reconfiguration algorithm define a sequence of configurations that begins with C_0 as the *initial configuration*; any other configuration can correspond to the failure state or an operational state of the array. The latter will be referred to as an *operational configuration*.

Upon detection of a faulty component, the reconfiguration algorithm may not send the array to an operational configuration if any of the following happens:

1) The reconfiguration circuitry failed. This possibility can be considered through a *coverage factor* (denoted by c) defined as the probability of successful reconfiguration given that a fault has occurred [8]. This is a measure of the probability of successful operation of all circuitry related to fault detection, isolation, and reconfiguration. The coverage factor is assumed constant and it will be associated with failures of active components only.

Manuscript received September 11, 1990; revised October 21, 1991. This work was supported in part by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization and was administered through the Office of Naval Research under Contracts 00014-85-k-0588 and 00014-88-k-0723.

N. Lopez-Benitez is with the Department of Electrical Engineering, Louisiana Tech University, Ruston, LA 71272.

J. A. B. Fortes is with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

IEEE Log Number 9200308.

0018-9340/92$03.00 © 1992 IEEE