Diversity Coding for Transparent Self-Healing and Fault-Tolerant Communication Networks

Ender Ayanoglu [1], *Chih-Lin I* [1], *R. D. Gitlin* [1], *and J. E. Mazo* [2]
AT&T Bell Laboratories

ABSTRACT

In this paper, a channel coding approach called *diversity coding* is introduced for self-healing and fault-tolerance in digital communication networks for nearly instantaneous recovery from link failures. To achieve this goal, the problem of link failures is treated as an erasure channel problem. Implementation details of this technique in existing and future communication networks are discussed.

[1] Communication Systems Research Laboratory, Crawfords Corner Road, Holmdel, NJ 07733-1988.
[2] Mathematical Sciences Research Center, 600 Mountain Avenue, Murray Hill, NJ 07974-2070.

1 Introduction

The need for rapid self-healing communication networks is increasing in importance as the backbone network becomes concentrated into fewer high-capacity links. For example, failure of an high capacity, such as 1.7 Gbps, link must be rapidly restored in order to minimize the disruption to tens of thousands of customers. The current approach to self-healing networks is to provide redundant network facilities for traffic rerouting. This class of self-healing networks is generally not capable of providing transparent recovery.

In this paper we present an error control based approach, called *diversity coding*, so that if M diverse links are available, then as long as any combination of $N \geq M$ information bearing and coded links survive, the network is transparently self-healing. The availability of diverse channels is not restricted to point-to-point configurations, but is implicit in most network topologies in existence today. The concept can also be applied to related applications in a single physical link: consider a wavelength-division multiplexed system where a small number of extra wavelengths are dedicated to coded information. If a transceiver fails, or if impairments (such as polarization dispersion) render a link inoperative, then because of the diversity coding, reliable non-stop transmission is possible. Further extensions are possible to systems that use error-detecting codes in the link, but which use diversity coding for error correction.

1.1 Existing Approaches to Self-Healing Networks

Most of the existing approaches for network self-healing or fault-tolerance fall within one or more of the following categories [1], [2]:

1. Protection switching for transmitter and receiver failures.
2. Dual-feeding, i.e., using 100% active extra capacity, all the time.
3. Restoration via an intelligent switch, using either dedicated restoration capacity or preemption of lower priority traffic.
4. Use of a central system, to detect failures, determine spare capacity, calculate new routes, and transmit cross-connect messages.
5. Use of distributed algorithms, such as precalculated routes for potential failures.
6. Traffic restoration by rerouting to unaffected trunks. This requires recalculating routes for every new call in a dynamic fashion.

In comparison with the above approaches, the diversity coding scheme proposed in this paper has, among others, the advantages of using extra capacity very efficiently, not requiring rerouting, and being nearly instantaneous. We illustrate the basic idea with a simple example in the next subsection.

1.2 1-for-N Diversity Coding

Consider N data lines that transmit binary data as in Figure 1. Let d_j be the information-bearing bits transmitted on the jth line for $1 \leq j \leq N$. Assume that an extra physically diverse line is available for protection against line failures, say due to physical disconnects, fading, polarization dispersion in fibers, etc. If we form

$$c_1 = d_1 \oplus d_2 \oplus \cdots \oplus d_N, \tag{1}$$

where $\oplus$ represents logical EXOR operation, i.e., modulo 2 addition, then the checksum c_1 can be sent on the $N+1^{\text{st}}$ link. In the case of a single line failure, say for the ith link, the receiver detects failure by carrier loss, and instantaneously generates

$$\hat{d}_i = c_1 \oplus \bigoplus_{\substack{j=1 \\ j \neq i}}^{N} d_j \tag{2}$$

since it has all d_j, except d_i, available. We used the symbol $\bigoplus_{j=1}^{N} d_j$ to denote the logical EXOR of the variables $d_1, d_2, \ldots, d_N$. By expanding c_1 as in (1), we get $\hat{d}_i = d_i$, since $d_j \oplus d_j = 0$, and recovery from the failure on the ith channel is achieved.

Note that, in this system, the transmitter always forms c_1 whether a line failure occurs or not. The receiver can detect a line failure and form $\hat{d}_i$ instantaneously. The system is optimum in the sense that it requires only one extra line to protect single line failures, it operates instantaneously in the case of a single line failure, the failure detection and recovery are accomplished by the receiver alone without communication with the transmitter and therefore a feedback channel is not needed.

This idea was proposed by Falconer and Gitlin for seasonal fading in microwave line-of-sight communications [3]. In what follows, we extend this technique to multiple line failures under the same optimality conditions as above. Similar ideas were employed in other fields such as magnetic recording or burst error correction [4], [5], [6]. In the context of the erasure channel, Wolf et al. used Reed-Solomon codes in an explicit construction to achieve channel capacity in their "postal" channel [7]. In packet-switched communication networks, the technique of *dispersity routing* introduced by

Maxemchuk [8] offers the advantages of a significantly smaller average delay and variance, less sensitivity to link utilization variations, and smaller buffer sizes for a given message loss probability due to buffer overflow. A similar idea has been suggested for lost packet recovery in high-speed networks [9], [10]. These applications also fall into the category of erasure channel coding.

2 Diversity Coding

We assume that the reader is familiar with the theory of finite fields. For more information on this subject we refer the reader to one of the standard texts on the subject such as [11] or [12], or to [13] for a summary. In Section 2.1 we present a minimum set of information on linear coding theory to introduce notation and nomenclature, and to enable the reader to follow the discussion in the rest of the section. Then, using this background, we describe two methods of diversity coding for failure protection in Section 2.2.

2.1 Review of Linear Coding Theory

Many implementable codes are linear, i.e., operations are performed using linear algebra over $GF(2^m)$ where $m \geq 1$. Linearity is desired because it makes the design, analysis, and implementation of codes easier, $m > 1$ is needed because of the very limited set of possibilities for encoding that $GF(2)$ provides. As m gets larger, the field $GF(2^m)$ becomes bigger, and the code designer has greater degrees of freedom in generating the code. Since it is harder to implement a code that requires a large m, however, the code designer tries to find the smallest m that satisfies the design requirements.

Channel coding is the controlled addition of redundancy into symbols to be transmitted over a noisy channel in order to combat noise. In $GF(2^m)$, the encoder of a block channel coding system forms the vector $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ with N consecutive m-bit data symbols $d_1, d_2, \ldots, d_N$, and generates the *channel codeword* $\mathbf{e} = (e_1, e_2, \ldots, e_K)$ from $\mathbf{d}$, where each e_i is an m-bit symbol, $K > N$. Then, $\mathbf{e}$ is transmitted over the channel, which is received as $\tilde{\mathbf{e}}$ where, due to channel noise, $\tilde{\mathbf{e}}$ may not be equal to $\mathbf{e}$. The decoder performs an inverse operation to generate $\hat{\mathbf{d}} = (\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_N)$ where each $\hat{d}_i$ is also an m-bit symbol. The encoder-decoder pair is designed in such a way that for sufficiently small number of errors in the channel, *error correction* is accomplished i.e., $\hat{\mathbf{d}} = \mathbf{d}$ even though $\tilde{\mathbf{e}} \neq \mathbf{e}$.

In a *linear block code,* $\mathbf{e}$ is a linear transformation of $\mathbf{d}$:

$$\mathbf{e} = \mathbf{dG} \tag{3}$$

where $\mathbf{G}$ is an $N \times K$ matrix of rank N whose entries are from $GF(2^m)$. In (3), and in the sequel, the addition and the multiplication operations are performed in $GF(2^m)$. $\mathbf{G}$ is called the *generator matrix* of the linear code. Let $M = K - N$. The decoder employs $\mathbf{H}$, an $M \times K$ matrix of rank M, called the *parity check matrix*, which has the property that

$$\mathbf{GH}^T = \mathbf{0}. \tag{4}$$

In expanding $(d_1, d_2, \ldots, d_N)$ into $(e_1, e_2, \ldots, e_K)$, it is often desirable that $e_i = d_i$ for the first N channel symbols. Then, when no channel errors are detected, the first N channel symbols can be delivered as decoded data symbols without extra processing. A code satisfying this property is called a *systematic code.* The generator matrix of a systematic code is of the form

$$\mathbf{G} = [\ \mathbf{I} \vdots \mathbf{P}\ ] \tag{5}$$

where $\mathbf{P}$ is an $N \times M$ matrix, called the *parity generator matrix.* Then, if we have

$$\mathbf{H} = [\ \mathbf{P}^T \vdots \mathbf{I}\ ] \tag{6}$$

(4) will be satisfied, since in $GF(2^m)$ the additive inverse of an element is itself.

In a systematic code, the channel symbols $e_{N+1}, e_{N+2}, \ldots, e_K$ which we will also denote by $c_1, c_2, \ldots, c_M$ are called the *parity symbols*, $\mathbf{c} = (c_1, c_2, \ldots, c_M)$ is given as

$$\mathbf{c} = \mathbf{dP}. \tag{7}$$

Some channels make erasures as well as errors. For example, a receiver may be designed such that a symbol is declared as erased when it is received ambiguously. In a received data stream, if erasures have occurred, their positions are known, whereas if errors have occurred, their positions are not known. This makes it simpler to correct erasures as compared to errors. In order to correct t errors and s erasures, a code must have at least $2t + s$ parity symbols, i.e.,

$$2t + s \leq M. \tag{8}$$

This standard result of coding theory can be obtained by combining the minimum distance bound [11, p. 11] with the Singleton bound [11, p. 50]. Most codes have considerably more parity symbols than this bound. Systematic codes that satisfy this bound and use $2t+s$ parity symbols to correct t errors and s erasures are known as *maximum distance separable codes* or *MDS codes.* Channels for which error control techniques are applied obviously expect some errors. They sometimes also have erasures. Our application is unique in that we are *only* interested in recoveries from erasures, hence we treat the channel as a pure erasure channel. This restriction is imposed since it is assumed that each channel has its own error protection mechanism. In the case of channels that make errors, a significant part of the decoding effort is spent on locating the errors. In the case of the pure erasure channel, the error location step is not needed and, the decoding operation is simpler.

In the following section, we describe an extension of the 1-for-N coding system of Figure 1 to M-for-N diversity coding systems, shown in Figure 2, for recoveries from M simultaneous line failures. The encoding-decoding operations to be introduced require using codes from $GF(2^m)$, or the processing of m-bit symbols, in order to be optimal in the sense of requiring only M parity links, and to take advantage of the rich linear algebraic properties that $GF(2^m)$ offers.

2.2 M-for-N Diversity Coding

Recall that in $GF(2^m)$ additions correspond to bit-by-bit EXOR operations. Hence, the coding operation in (1) can be represented in the format of (5) where $\mathbf{P} = (1, 1, \ldots, 1)^T$ is an $N \times 1$ column vector of all 1's. This scheme protects a single line failure out of N lines using a single parity line. Note that failures can occur simultaneously on M lines where $M > 1$, and coding theory suggests that we should be able to protect M lines using M parity lines. Our goal in this section is to extend the technique of Section 1.2 to $M > 1$.

Let $d_1, d_2, \ldots, d_N$ represent m-bit blocks from the lines $1, 2, \ldots, N$ respectively. We would like to protect M simultaneous line failures by providing M m-bit parity symbols $c_1, c_2, \ldots, c_M$, $1 \leq M \leq N$, where we restrict $M \leq N$ for practical reasons. This encoding is carried out linearly as

$$c_j = \sum_{i=1}^{N} \beta_{ij} d_i \qquad 1 \leq j \leq M, \tag{9}$$

where multiplication and summation are performed in $GF(2^m)$. In the notation of (5), $\mathbf{P} = [\beta_{ij}]_{N \times M}$. The parity symbols c_j are then transmitted to the receiver along with the data symbols. Consider first the case when n of the N data lines fail ($1 \leq n \leq M$). At the receiver their carrier signals drop, and the receiver detects the failures. Let $k_1, k_2, \ldots, k_n$ be the indices of the links that failed; we generate signals $\tilde{c}_j$ as

$$\tilde{c}_j = c_j + \sum_{\substack{i=1 \\ i \neq k_1, k_2, \ldots, k_n}}^{N} \beta_{ij} d_i \qquad 1 \leq j \leq n. \tag{10}$$

This can easily be done since β_{ij} are fixed and known at the receiver, and d_i for $1 \leq i \leq N$, $i \neq k_1, k_2, \ldots, k_n$ are available. Note from (9) and (10) that

$$\tilde{c}_j = \sum_{i = k_1, k_2, \ldots, k_n} \beta_{ij} d_i \qquad 1 \leq j \leq n. \tag{11}$$

The n erased data symbols $d_{k_1}, d_{k_2}, \ldots, d_{k_n}$ can be recovered from $\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n$ via an inverse linear transform, provided β_{ij} are chosen such that the column vectors $(\beta_{k_1 j}, \beta_{k_2 j}, \ldots, \beta_{k_n j})^T$ for $1 \leq j \leq n$, $1 \leq k_1 < k_2 < \cdots < k_n \leq N$, and $1 \leq n \leq M \leq N$ are all linearly independent. This can be checked by considering the determinant of the matrix $\mathbf{B}_{k_1, k_2, \ldots, k_n} = [\beta_{k_i j}]_{n \times n}$.

Let

$$\beta_{ij} = \alpha^{(i-1)(j-1)}, \tag{12}$$

where α is a primitive element of $GF(2^m)$. Let

$$m = \lceil \log_2(N+1) \rceil, \tag{13}$$

where $\lceil x \rceil$ is the smallest integer greater than or equal to x. Note that $\mathbf{B}_{k_1, k_2, \ldots, k_n}$ is a Vandermonde matrix. By using a well-known result from linear algebra, we have [11, p. 170]

$$\det \mathbf{B}_{k_1, k_2, \ldots, k_n} = \prod_{1 \leq i < j \leq n} (\alpha^{k_j - 1} - \alpha^{k_i - 1}). \tag{14}$$

None of the entries in the product in (14) can be zero, since in $GF(2^m)$ the additive inverse of a member, which is unique, is itself; in other words, $\alpha^{k_j - 1} = \alpha^{k_i - 1}$ if and only if $i = j$. Therefore,

$$\det \mathbf{B}_{k_1, k_2, \ldots, k_n} \neq 0 \tag{15}$$

for $1 \leq k_1 < k_2 < \cdots < k_n \leq N$, $1 \leq n \leq M \leq N$, and there exists a linear inverse transform $\mathbf{B}^{-1}_{k_1, k_2, \ldots, k_n}$ to obtain $d_{k_1}, d_{k_2}, \ldots, d_{k_n}$ as

$$(d_{k_1}, d_{k_2}, \ldots, d_{k_n}) = (\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n)\ \mathbf{B}^{-1}_{k_1, k_2, \ldots, k_n}. \tag{16}$$

Matrices of the form of $\mathbf{P}$ above are called *Fourier* matrices since $\mathbf{P}$ is in the form of the discrete Fourier transform matrix. The code we described above can be viewed as taking the discrete Fourier transform of the data in $GF(2^m)$ (also known as the *Fourier-Galois transform*), and transmitting the transform as parity information. This is an important observation, since the discrete Fourier transform operation is well-studied and optimized for implementation by means of fast Fourier transform algorithms. The conventional fast Fourier transform algorithm over the field of complex numbers is a signal flow graph with additions and multiplications by a power of $e^{-j\frac{2\pi}{N}}$ such that the number of operations in the discrete Fourier transform is reduced. When the discrete Fourier transform is over $GF(2^m)$, the same signal flow graph can be used by simply replacing $e^{-j\frac{2\pi}{N}}$ by α, to obtain the corresponding fast Fourier transform in $GF(2^m)$, minimizing the operations in the encoder of the diversity coding system.

From (10) to (16) we have assumed that all the failures occur in the data lines. This system may be used to recover from $n \leq M$ simultaneous line failures out of $d_1, d_2, \ldots, d_N$ in an environment where the M parity lines never fail. However, we can solve the more general problem where failures are allowed in both data and parity lines by using the $\mathbf{P}$ matrix, and by appropriately choosing the finite field. Let $c_1, c_2, \ldots, c_M$ be generated as in (9) where $\beta_{ij} = \alpha^{(i-1)(j-1)}$ as in $\mathbf{P}$ above. We now assume that any $n \leq M$ lines out of $d_1, d_2, \ldots, d_N$ and $c_1, c_2, \ldots, c_M$ can fail. Assume that, due to line failures, $d_{k_1}, d_{k_2}, \ldots, d_{k_n}$ are not available at the receiver, but $c_{l_1}, c_{l_2}, \ldots, c_{l_n}$ are active, where $1 \leq n \leq M$, $1 \leq k_1 < k_2 < \cdots < k_n \leq N$, and $1 \leq l_1 < l_2 < \cdots < l_n \leq M$. Similarly to (10), generate signals $\tilde{c}_i$

$$\tilde{c}_j = c_{l_j} + \sum_{\substack{i=1 \\ i \neq k_1, k_2, \ldots, k_n}}^{N} \beta_{il_j} d_i = \sum_{i=k_1, k_2, \ldots, k_n} \beta_{il_j} d_i \qquad 1 \leq j \leq n. \tag{17}$$

In this case, for the n erased data symbols $d_{k_1}, d_{k_2}, \ldots, d_{k_n}$ to be recoverable from $\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n$, the matrix $\mathbf{B}_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n} = [\alpha^{(l_i-1)(k_j-1)}]_{n\times n}$, should be invertible. In other words, we would like *any* $n \times n$ square submatrix of $\mathbf{P}$ to be invertible, where $1 \leq n \leq M$ [12, p. 321].

$\mathbf{B}_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n}$ is not a Vandermonde matrix in general. Therefore, it cannot be verified nonsingular with the method we used for $\mathbf{B}_{k_1,k_2,\ldots,k_n}$. However, if the field $GF(2^m)$ is chosen large enough, then $\mathbf{B}_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n}$ must be nonsingular. In [13], we proved that if

$$m > \max_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n} \deg \det \mathbf{B}_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n} = \sum_{i=1}^{M-1} (M-i)(N-i), \tag{18}$$

then

$$\det \mathbf{B}_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n} \neq 0 \tag{19}$$

for $1 \leq k_1 < k_2 < \cdots < k_n \leq N$, $1 \leq l_1 < l_2 < \cdots < l_n \leq M$, and one can always recover the data in the case of a failure in any $(n \leq M)$ lines out of $N+M$ data and parity lines via

$$(d_{k_1}, d_{k_2}, \ldots, d_{k_n}) = (\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n)\ \mathbf{B}^{-1}_{k_1,k_2,\ldots,k_n;l_1,l_2,\ldots,l_n}. \tag{20}$$

Note that the codes defined above correct M erasures with M parity symbols, and are in systematic form, therefore they are MDS codes.

The bound in (18) is usually very pessimistic. For a specific M, one can find a lower value for m that will satisfy the requirement. By carefully looking at some specific values of M, and comparing the required m and the one given by (18), we find [13]

M	Actual Required m	Bound Given by (18)
2	$\lceil \log_2(N+1) \rceil$	N
3	$\lceil \log_2(N+1) \rceil$	$3N-3$
4	N	$6N-7$

By an inequality in [12, p. 321], the smallest m possible when $M=2$ or 3 is achieved by the method above. For $M=2$ and $M=3$, these codes are equal to extended Reed-Solomon codes [12, p. 326], in shortened form when $\log_2(N+1)$ is not an integer. However, in $M=4$, we have lost the logarithmic dependency of m on N. This logarithmic dependency can be recaptured with the techniques to be described next.

An alternative to making the $\mathbf{P}$ matrix Fourier is to make the parity check matrix of the code equivalent to a Fourier matrix $\tilde{\mathbf{H}} = [\alpha^{(i-1)(j-1)}]_{M\times(M+N)}$. Although an associated $\mathbf{G}$ matrix can be obtained from (4), it is desirable to have the $\mathbf{G}$ matrix in systematic form so as not to corrupt $d_1, d_2, \ldots, d_N$. This can be easily done by elementary row and column operations. In particular, let

$$\mathbf{H} = \left[\begin{array}{cccc} \tilde{\mathbf{h}}_{N+1} & \tilde{\mathbf{h}}_{N+2} & \cdots & \tilde{\mathbf{h}}_{N+M} \end{array}\right]^{-1} \tilde{\mathbf{H}} \tag{21}$$

where $\tilde{\mathbf{h}}_i$ is the ith column vector of $\tilde{\mathbf{H}}$ for $1 \leq i \leq N+M$, and

$$m = \lceil \log_2(N+M+1) \rceil, \tag{22}$$

and define $\mathbf{G}$ via (5) and (6). The inverse in (21) exists, since due to (22), the matrix to be inverted is Vandermonde. This system enables any N of the $N+M$ codeword symbols $e_1, e_2, \ldots, e_{N+M}$ to determine the remaining M. To see this, let $\mathbf{v} = (e_{l_1}, e_{l_2}, \ldots, e_{l_N})$ be the vector of the N known members of $\mathbf{e}$, and let $\mathbf{u} = (e_{k_1}, e_{k_2}, \ldots, e_{k_M})$ be the remaining M members of $\mathbf{e}$, where $1 \leq l_i, k_j \leq N+M$, $1 \leq i \leq N$, and $1 \leq j \leq M$. Some or all members of $\mathbf{u}$ are unknown, whereas all members of $\mathbf{v}$ are known. Let $\mathbf{h}_i$ be the ith column vector of $\mathbf{H}$. In a linear channel code $\mathbf{e}\mathbf{H}^T = \mathbf{0}$. Rearranging this equation, we have

$$\mathbf{u}\mathbf{H}_u^T + \mathbf{v}\mathbf{H}_v^T = \mathbf{0} \tag{23}$$

where we have defined $\mathbf{H}_u = \left[\begin{array}{cccc} \mathbf{h}_{k_1} & \mathbf{h}_{k_2} & \cdots & \mathbf{h}_{k_M} \end{array} \right]$ and $\mathbf{H}_v = \left[\begin{array}{cccc} \mathbf{h}_{l_1} & \mathbf{h}_{l_2} & \cdots & \mathbf{h}_{l_N} \end{array} \right]$. $\mathbf{H}_u$ is an $M \times M$ matrix that can also be expressed, due to (21), as

$$\mathbf{H}_u = \left[\begin{array}{cccc} \tilde{\mathbf{h}}_{N+1} & \tilde{\mathbf{h}}_{N+2} & \cdots & \tilde{\mathbf{h}}_{N+M} \end{array} \right]^{-1} \left[\begin{array}{cccc} \tilde{\mathbf{h}}_{k_1} & \tilde{\mathbf{h}}_{k_2} & \cdots & \tilde{\mathbf{h}}_{k_M} \end{array} \right]. \tag{24}$$

From (24) we have,

$$\det \mathbf{H}_u = \frac{\det \left[\begin{array}{cccc} \tilde{\mathbf{h}}_{k_1} & \tilde{\mathbf{h}}_{k_2} & \cdots & \tilde{\mathbf{h}}_{k_M} \end{array} \right]}{\det \left[\begin{array}{cccc} \tilde{\mathbf{h}}_{N+1} & \tilde{\mathbf{h}}_{N+2} & \cdots & \tilde{\mathbf{h}}_{N+M} \end{array} \right]}. \tag{25}$$

Both of the matrices on the right hand side of (25) are nonsingular since they are both Vandermonde with distinct elements on the second row, due to (22). Therefore $\det \mathbf{H}_u$ is not zero, $\mathbf{H}_u$ is invertible, and the unknown members of $\mathbf{e}$ can be obtained from the known ones.

Since the codes described above can also correct M erasures with M parity symbols, and are in systematic form, they are also MDS codes, as were those described previously. Codes whose parity check matrices are equivalent to a Fourier matrix such as the ones described above belong to the class of Reed-Solomon codes [12]. There exist fast methods for calculating error magnitudes for Reed-Solomon codes, such as the Forney algorithm [11, p. 183], or the frequency domain techniques [11, p. 256]. A slight reduction in field size can be obtained by using extended Reed-Solomon codes [12, p. 323], making the field size equal to

$$m = \lceil \log_2(N + M - 1) \rceil. \tag{26}$$

3 Applications and Extensions

In this section, we extend the technique in Section 2 to applications in multiterminal topologies.For applications in trunk failures, fiber optic networks with wavelength division multiplexing, packet delay and loss in packet-switched networks, distributed storage, hitless protection switching, fault-tolerant parallel transmission of continuous-amplitude discrete-time signals, and implementation via multiplexing and demultiplexing, we refer the reader to [13], [14], and [15].

3.1 Multiterminal Topologies

The discussion in Section 2 assumes a point-to-point topology as shown in Figure 3(a). Observe that the operation of the encoder of such a system does not change whether or not there are link failures. Furthermore, the encoding is a multiply-and-add operation for the data in each link, and at each step, the encoder only needs to know the running sum up to that point, and the data for the ith link. This means that the sources of different links do not have to be colocated. This enables a multipoint-to-point implementation as shown in Figure 3(b). In this scheme, the source of d_1 sends $d_1\mathbf{p}_1$ to the source of d_2. The encoder at the source of each data link d_i receives a running sum $\sum_{j=1}^{i-1} d_j\mathbf{p}_j$ from the source of d_{i-1} where $\mathbf{p}_j$ is the jth row vector of $\mathbf{P}$, $2 \le i \le N-1$. It forms $d_i\mathbf{p}_i$, and adds it to $\sum_{j=1}^{i-1} d_j\mathbf{p}_j$ to form $\sum_{j=1}^{i} d_j\mathbf{p}_j$ which it sends to the source of d_{i+1}. After the Nth encoder, the parity data $\mathbf{c} = \mathbf{dP} = \sum_{j=1}^{N} d_j\mathbf{p}_j$ is formed, which is, after delay equalization, transmitted to the destination.

Another implementation of the multipoint-to-point scheme is shown in Figure 4. In this case, every source also transmits to a central processor, perhaps via satellite, which forms $\mathbf{c} = \mathbf{dP}$, and sends it to the destination. The importance of this scheme is that there is very little processing required in the central station, and in the case of link failures, the decoding is done at the destination without any involvement of the central processor.

A natural extension of the scheme in Figure 3(b) is the multipoint-to-multipoint topology shown in Figure 5. Assume there are N links in Figure 5, and let $f : \{1, 2, \ldots, S\} \times \{S+1, S+2, \ldots, S+D\} \to \{1, 2, \ldots, N\}$ map every source-destination pair to an integer in the range from 1 to N. Let $\mathbf{P}$ be a parity generator matrix of size $N \times M$ chosen using the methods of Section 2, and let $\mathbf{p}_j$ be its jth row vector, $1 \le j \le N$. In this case, every source node i forms an outbound checksum

$$\mathbf{c}_i^{\text{out}} = \sum_{j=S+1}^{S+D} d_{i,j}\mathbf{p}_{f(i,j)} \qquad 1 \le i \le S, \tag{27}$$

and each destination node j forms an inbound checksum

$$\mathbf{c}_j^{\text{in}} = \sum_{i=1}^{S} d_{i,j}\mathbf{p}_{f(i,j)} \qquad S+1 \le j \le S+D. \tag{28}$$

At a location close to all the sources i, the sum of all outbound checksums is formed to obtain

$$\mathbf{c}^{\text{out}} = \sum_{i=1}^{S} \mathbf{c}_i^{\text{out}} = \sum_{j=S+1}^{S+D}\sum_{i=1}^{S} d_{i,j}\mathbf{p}_{f(i,j)} = \sum_{k=1}^{N} d_{f^{-1}(k)}\mathbf{p}_k = \mathbf{dP} \tag{29}$$

and transmitted to the central decoder, where $\mathbf{d} = (d_{f^{-1}(1)}, d_{f^{-1}(2)}, \ldots, d_{f^{-1}(N)})$. The central decoder is located physically close to the destination nodes, and it forms the sum of all the inbound checksums

$$\mathbf{c}^{\text{in}} = \sum_{j=S+1}^{S+D} \mathbf{c}_j^{\text{in}}. \tag{30}$$

Normally, we have the "conservation of data" equation:

$$\mathbf{c}^{\text{in}} + \mathbf{c}^{\text{out}} = \mathbf{0}. \tag{31}$$

Equation (31) expresses the fact that the sum of incoming data into a cutset that divides the network vertically in the middle is equal to the sum of outgoing data from that cutset; no data are generated in the cutset, and normally, no data are lost. And in fact, if some are lost, we generally will be able to recover by using the "total data" information available to us via $\mathbf{c}^{\text{in}}$ and $\mathbf{c}^{\text{out}}$ as will be shown below.

When $n \leq M$ link failures occur, the destination nodes with failed links inform the central decoder of the indices of the failed links, $k_1, k_2, \ldots, k_n$. The encoder recovers failed links using $\mathbf{c} = \mathbf{c}^{\text{out}} = \mathbf{dP}$, $\tilde{\mathbf{c}} = \sum_{k=1, k \neq k_1, k_2, \ldots, k_n}^{N} d_{f^{-1}(k)} \mathbf{p}_k = \mathbf{c}^{\text{in}}$, $k_1, k_2, \ldots, k_n$, and by inverting an appropriate matrix with the methods of Section 2.

Some simplifications exist for $M = 1$. For the configuration in Figure 5 and for $M = 1$, there are, in fact, three different strategies that the receivers and the central decoder can use in the case of link failures as described below.

1. Nothing special is done. The decoder always broadcasts $c^{\text{in}} \oplus c^{\text{out}}$ to all receivers. In the case of a failure, say $d_{k,l}$ as above, the lth receiver keeps transmitting its incomplete inbound checksum, and since $c^{\text{in}} \oplus c^{\text{out}} = d_{k,l}$, the decoder instantly broadcasts $d_{k,l}$ to all receivers. Since the lth receiver, and no other, is expecting $d_{k,l}$, the recovery is complete. This method has the advantage of being instantaneous. The disadvantages of this approach are that security may be a problem due to broadcasting, and that a method needs to be devised for the detection of multiple line failures to different receivers since, otherwise, several receivers will expect to receive their failed link data, but actually receive the sum of the data in all the failed links which is useless to their users.

2. The receiver whose one or several input links has failed, say the lth receiver, stops transmitting its inbound checksum. The central decoder forms $c^{\text{in}} \oplus c^{\text{out}}$ which now equals c_l^{in}, and sends it to the lth receiver. If only one link to the lth receiver has failed, the lth receiver recovers the

data in that link by the same principle, using its healthy inbound links, its inbound checksum c_l^{in}, and the technique in Section 1.2. If more than one link going into the lth receiver has failed, this condition is detected by the lth receiver, and the necessary action is taken to inform the end-users. If two or more receivers have simultaneous failures in their inbound links, they all stop transmitting their inbound checksums; the multiple failure is detected by the central decoder, and the necessary action is taken to inform the destination nodes, which in turn inform their end-users. As the first method above, this method is also nearly instantaneous. Further, it requires only one link between a receiver and the central decoder (the direction of transmission on this link needs to be switched in the case of a failure). Moreover, the security problem existent above does not exist here. The only disadvantage of this approach is the requirement of decoders at the receivers. But, since these decoders are extremely simple, this is not a significant disadvantage.

3. The two methods above do not generalize to protection against multiple line failures, since in that case the central decoder needs to know the indices of the failed links for matrix inversion in decoding. In the third method for $M = 1$, which is a special case of the method for $M > 1$, the receivers continue to send their incomplete inbound checksums to the central decoder in the case of failure, also transmitting the indices of the failed channels via a side channel (the low-rate side channel could be derived from the high-rate data link). The central decoder takes the necessary action if there are more than one simultaneous link failures. In the case of a single failure, it performs the decoding, and sends the recovered data only to the pertinent receiver. This solution does not have the security problem the first method has, and it automatically detects multiple failures. It does not require extra complexity at the receivers as in the second method above, and it generalizes to the case for $M > 1$. However, it requires a side channel, and may be slower than the first two methods above, but it is still faster than a system that requires transmitter and receiver switchovers to extra capacity.

This method can be generalized to a general topology for any M. Refer to Figure 6, which shows the node i of a network with an arbitrary topology. It is assumed that there is a central processor that handles recoveries. The node i talks to the central processor via $3M$ connections. The first M connections are used to send the inbound checksum for node i

$$\mathbf{c}_i^{\mathrm{in}} = \sum_{j:\ i \text{ receives from}} d_{j,i}\mathbf{p}_{f(j,i)}, \tag{32}$$

the second M connections are used to send the outbound checksum for node i

$$\mathbf{c}_i^{\text{out}} = \sum_{j\text{: } i \textbf{ transmits to}} d_{i,j}\mathbf{p}_{f(i,j)}, \tag{33}$$

and the remaining M lines are used by the central processor to send the recovered data to node i in the case of failures in its inbound links. The central processor forms the summation of all the inbound checksums to obtain $\mathbf{c}^{\text{in}} = \sum_i \mathbf{c}_i^{\text{in}}$, and the summation of all the outbound checksums to obtain $\mathbf{c}^{\text{out}} = \sum_i \mathbf{c}_i^{\text{out}}$. Then, the operation of the central processor is exactly the same as the operation of the system in Figure 5. Note that going from the topology of Figure 5 to the arbitrary topology of Figure 6 is accomplished by paying a penalty of going from M protection lines to $3M$ protection lines.

In the cases with multiple destinations, feedback channels are needed from the destination nodes to the central decoder *only.* These feedback channels are not protected by the multiterminal diversity coding system, and should be protected with either a conventional technique such as dual feeding, or with point-to-point diversity coding.

The diversity coding system introduced here should be compared with a system that switches over transmitters and receivers to spare capacity under the management of a protocol in the case of line failures. The advantage of the diversity coding system is its speed. Since transmitter and receiver switchovers are not required and protocol delays are avoided, the recovery can take place very fast, in an almost instantaneous manner. If speed is not the only criterion, then in order for the diversity system to be preferable, its cost should be less than the system with switchovers. Since most of the cost in a wide area network is in the physical links, the diversity coding system should not introduce a large number of extra links over long distances. In that regard, the point-to-point system described above is very efficient. It does not require any more links than the system with switchovers. The multipoint-to-multipoint and the multipoint-to-point systems are efficient as long as the source nodes, as well as the destination nodes, are physically located in close proximity of the other source or destination nodes, respectively, and the source cluster and the destination cluster are distant. In other words, these topologies are efficient as long as their physical layouts approximate point-to-point topologies. On the other hand, the general network topology solution requires three times more links than the system with switchovers. It can be preferable only if the speed advantages overweigh the extra cost due to the extra links.

4 Implementation

In this section, we describe the synchronization requirements caused by length differences among links between sources and destinations in the configuration in Figure 5. The technique can be extended to other configurations.

Since the scheme is based on recovering failed links using data sent over physically diverse, and hence of different length links, care must be taken to synchronize the coded data. This can be accomplished by delay equalization. It is important to note that in ordinary operation, the delay on the data links is not changed. In the event of a link failure, the restoration is accomplished in a time less than the maximum differential delay between any source destination pair over the coded and uncoded links as quantified below.

For example, consider Figure 5 which has been redrawn in Figure 7 in detail to show synchronization for transmission between the source node 1 and the destination node $S+1$. As shown in Figure 5, let $t_{i,j}$ be the propagation delay between the destination node j and the source node i, let $t_{i,c}$ be the propagation delay between the source i and the summing junction, let $t_{c,d}$ be the propagation delay between the summing junction and the decoder, and let $t_{j,d}$ be the propagation delay between the destination node j and the central decoder. Then, the source node i launches the data $d_{i,j}$ and the output parity $\mathbf{c}_i^{\rm out}$ at the same time t. Normally, the data arrive at the destination node j at time $t+t_{i,j}$, and are delivered to the end-user. Simultaneously, at the summing junction, the output parity data from all sources are synchronized and their sum is transmitted at time $t+\delta_c$ where

$$\delta_c = \max_{1\leq i\leq S} t_{i,c}. \tag{34}$$

For that purpose, output parity data $\mathbf{c}_i^{\rm out}$ from source i is delayed by $\Delta_{i,c} = \delta_c - t_{i,c}$ seconds. At the destination node j, the input parity $\mathbf{c}_j^{\rm in}$ is formed at time $t+\delta_j$, and transmitted to the central decoder, where

$$\delta_j = \max_{1\leq i\leq S} t_{i,j} \qquad S+1\leq j\leq S+D. \tag{35}$$

To form $\mathbf{c}_j^{\rm in}$ at the destination node j, data from source i is delayed $\Delta_{i,j} = \delta_j - t_{i,j}$ seconds. At the central decoder, the decoding is performed at time $t+\delta_d$ where

$$\delta_d = \max(\delta_c + t_{c,d}, \max_{S+1\leq j\leq S+D} \delta_j + t_{j,d}), \tag{36}$$

and each input parity line $\mathbf{c}_j^{\rm in}$ is delayed $\Delta_{j,d} = \delta_d - t_{j,d}$ seconds. The decoded data are delivered to the end-user at time $t+\delta_d+t_{S+1,d}$. In the case of the diversity coding system, the maximum delay introduced due to synchronization, as well as the maximum required storage per destination node is of the order of the differential delay between the source and the destination taken by the actual data and the parity data.

In particular, for a point-to-point system, assume there are N information-bearing links between the source and the destination as in Figure 8, and a single extra link is provided for protection, to be used for diversity coding as in Figure 8(a), or for transmitter and receiver switchover as in Figure 8(b). Assume for simplicity that the propagation delay between the source and the destination over each of the N links is τ seconds (in the general scheme, we can consider τ as the maximum propagation delay in the N links). Let, for the same source and destination pair, the propagation delay and processing delays over the spare capacity sum to $\rho\tau$ seconds, where $\rho \geq 1$. With diversity coding, when the direct link fails, the data launched at time t arrives at the destination at time $t + \rho\tau$ instead of $t + \tau$ seconds, thus incurring an added delay of $(\rho - 1)\tau$ seconds, and requiring *total* storage of $(\rho - 1)\tau$ seconds of data at the destination node, i.e., only as much as the differential delay. Whereas, in a system that switches its transmitters and receivers over to spare capacity, in addition to τ seconds for a cut to be detected, the time needed for the transmitter to be informed and the retransmitted data to reach the receiver over the spare capacity lines is $2\rho\tau$ seconds. Hence, the added delay is $2\rho\tau$ seconds, and $(\rho + 1)\tau$ seconds of data needs to be stored at the transmitter node *per each link*, i.e., a total of $N(\rho + 1)\tau$ seconds of data needs to be stored. For a differential distance of 500 Km and transmission at 1.7 Gbps, this corresponds to a delay of 2.5 msec and total storage of 530 Kbytes of information for the diversity coding system. Note that this is much more manageable than 55 msec of delay and 11.13 Mbytes of storage *per link* that would be required if we opted for transmitter and receiver switchover to spare capacity, assuming source-to-destination distance of 5,000 Km, and no loss of data as in the diversity coding system.

Synchronization problems are significantly harder in a system that switches transmitters and receivers to spare capacity, since in such a system both the transmitter and the receiver need to be synchronized after a switchover. This increases the switchover delay significantly. For stationary networks, synchronization is achieved and maintained with diversity coding when the system is first started as opposed to the resynchronization necessary in a switchover system after a link failure is detected. With nodal clock drifts and time-varying link path lengths, synchronization can be maintained in a diversity coded network comprised of SONET links by using the SONET pointers to track the changes in differential delay [16]. Other methods may be appropriate for the existing

DS-N hierarchy.

5 Summary and Conclusions

In this paper we introduced a channel coding approach, called diversity coding, to self-healing and fault-tolerance in digital communication networks by treating link failures as an erasure channel problem.

Diversity coding achieves nearly instantaneous recovery and transparency to the end-user. The recovery is accomplished at the receiver end, without informing the transmitter, thus a feedback channel to the source nodes is not needed. In configurations where there is a single destination, no feedback channel is needed; in configurations with multiple destinations, only a feedback channel to a central decoder is needed. The required excess capacity is minimum; to protect failures in up to M simultaneous channels, only M more channels are needed with protection against failure in all the channels, including excess capacity. In the case of a failure, rerouting of traffic is not needed, saving the search time for available routes, processing delay, and complexity. The synchronization problem is solved by delay equalization at the network initialization time, as opposed to every time a failure occurs, saving handshaking protocol delays.

We have shown that diversity coding is efficiently applicable to point-to-point, multipoint-to-point, and multipoint-to-multipoint topologies. It can also be used in arbitrary topologies if the speed of recovery is the most important concern. Encoding-decoding complexity and memory requirements are very small and the system can be implemented with off-the-shelf components. Since there is no insertion delay on the data links, the technique can be implemented as an add-on to an existing network. It can be implemented for transmission at arbitrarily high speeds by parallel processing at low speeds and demultiplexing and multiplexing of high speed data. In conjunction with the existing error detection schemes, it can be used for forward error correction for random and burst errors, reducing delay since nonselective and selective repeat requests of the windowed protocols can be eliminated. It is extended to trunked lines where failures occur simultaneously. Finally, its applications can be extended to routing in packet-switched networks, distributed storage, hitless protection switching, and protection of continuous-amplitude discrete-time signals.

6 Acknowledgements

We gratefully acknowledge the valuable discussions with J. E. Abate, R. Adleman, G. Ash, R. A. Calderbank, B. G. Cortez, W. Daumer, R. S. Dighe, J. Kaplan, J. Salz, and N. J. A. Sloane, and thank the anonymous reviewers for their insightful comments.

References

[1] Special Issue on Surviving Disaster, *IEEE Communications Magazine,* June 1990.

[2] Session on Restoration of Fiber Networks, *Proceedings of IEEE Global Communications Conference,* Vol. 2, pp. 23.1.1–23.6.5, Dallas, Texas, November, 1989.

[3] D. D. Falconer and R. D. Gitlin, "Frequency-diversity coding for data-under-voice (DUV) transmission," unpublished Bell Laboratories Technical Memorandum, 1975.

[4] A. M. Patel, US Patent No: 4,205,324, May 1980.

[5] D. L. Schilling, US Patents No: 4,796,260, January 1989, 4,849,974, 4,847,842, and 4,849,976, July 1989.

[6] A. F. Pavelchek, D. Manela, D. L. Schilling, "An EDAC scheme for slow frequency hopped systems," *Proceedings of IEEE International Conference on Communications,* Vol. 1, pp. 9.1.1-9.1.5. Toronto, Canada, June 1986.

[7] J. K. Wolf, A. D. Wyner, and J. Ziv, "The capacity of the postal channel," *Information and Control,* Vol. 16, pp. 167–172, 1970.

[8] N. F. Maxemchuk, "Dispersity routing," *Proceedings of the International Communications Conference*, pp. 41.10–41.13, 1975.

[9] N. Shacham and P. McKenney, "Packet recovery in high-speed networks using coding and buffer management," *Proceedings of IEEE INFOCOM '90,* Vol. 1, pp. 124–131, San Francisco, California, June 1990.

[10] T. Kitami and I. Tokizawa, "Cell loss compensation schemes employing error correction coding for asynchronous broadband ISDN," *Proceedings of IEEE INFOCOM '90,* Vol. 1, pp. 116–123, San Francisco, California, June 1990.

[11] R. E. Blahut, *Theory and Practice of Error Control Codes,* Addison-Wesley Publishing Company, Reading, Mass., 1983.

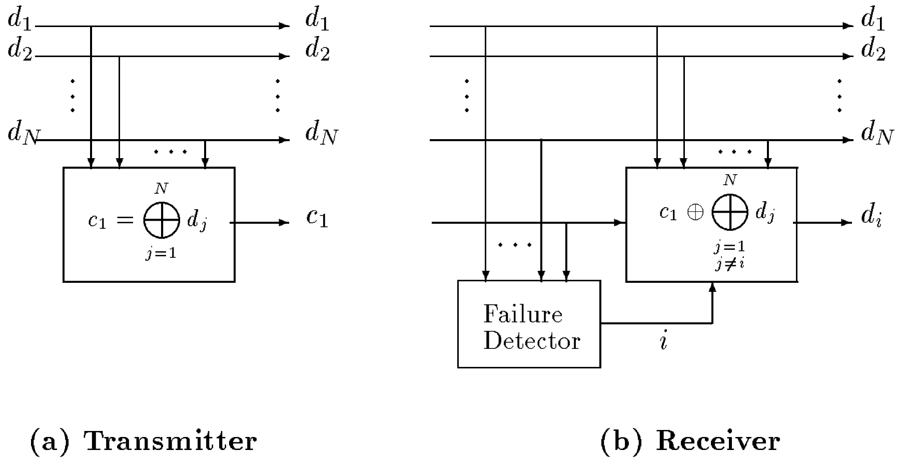
[12] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, New York, 1977.

[13] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, "Diversity coding: using error control for self-healing in communication networks," *Proceedings of IEEE INFOCOM '90,* Vol. 1, pp. 95–104, San Francisco, California, June 1990.

[14] C.-L. I, E. Ayanoglu, R. D. Gitlin, and J. E. Mazo, "Transparent self-healing communication networks via diversity coding," *Proceedings of the IEEE 1990 International Conference on Communications,* pp. 308.6.1-308.6.6, Atlanta, Georgia, April 1990.

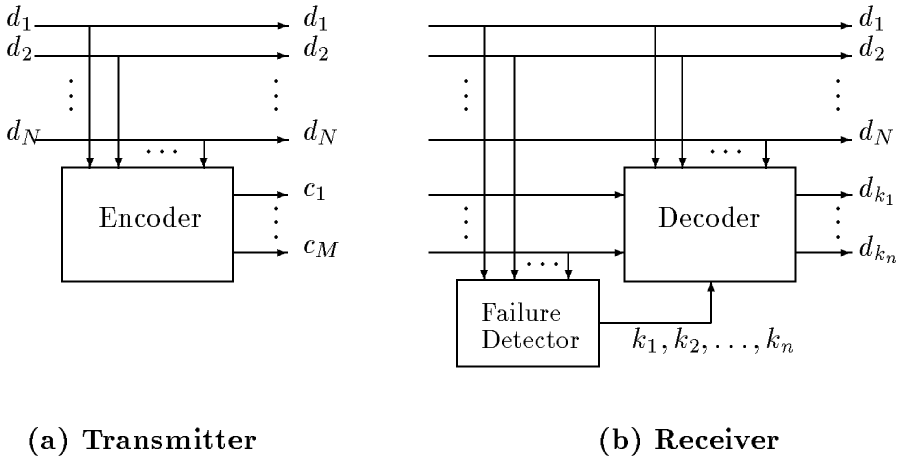
[15] E. Ayanoglu, C.-L. I, R. D. Gitlin, and I. Bar-David, "Analog diversity coding for transparent self-healing communication networks," *Proceedings of IEEE Global Communications Conference,* pp. 500.5.1–500.5.6, San Diego, California, December 1990.

[16] R. Ballart and Y.-C. Ching, "SONET: Now it's the standard optical network," *IEEE Communications Magazine,* Vol. 27, pp. 8–15, March 1989.


d_1
d_2
d_N
d_1
d_2
d_N
$c_1 = \bigoplus_{j=1}^{N} d_j$
c_1
d_1
d_2
d_N
$c_1 \oplus \bigoplus_{j=1, j\neq i}^{N} d_j$
d_i
Failure Detector
i

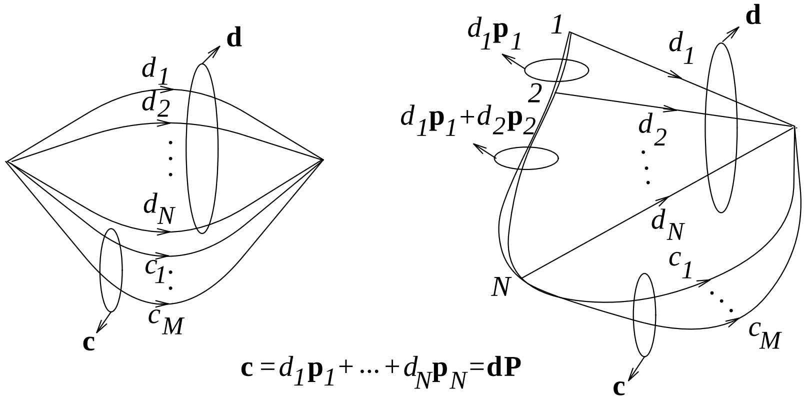
(a) Transmitter **(b) Receiver**

Figure 1: 1-for-N diversity coding system.



(a) Transmitter **(b) Receiver**

Figure 2: M-for-N diversity coding system.



(a) Point-to-point **(b) Multipoint-to-point**

Figure 3: Topologies for diversity coding.

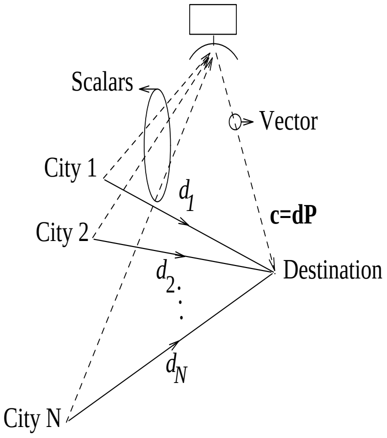


Figure 4: Using diversity coding in a multipoint-to-point application via a central node.