[17] J. K. Omura, "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-15, Jan. 1969, pp. 177–179.
[18] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM J. Res. Dev.*, vol. 13, no. 6, Nov. 1969, pp. 675–685.
[19] E. A. Bucher and J. A. Heller, "Error probability bounds for systematic convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-16, Mar. 1970, pp. 219–224.
[20] J. P. Odenwalder, "Optimal decoding of convolutional codes," Ph.D. dissertation, Dep. Syst. Sci., Sch. Eng. Appl. Sci., Univ. California, Los Angeles, 1970.
[21] G. D. Forney, Jr., "Coding and its application in space communications," *IEEE Spectrum*, vol. 7, June 1970, pp. 47–58.
[22] ——, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, Nov. 1970, pp. 720–738; "II: Maximum likelihood decoding," and "III: Sequential decoding," *IEEE Trans. Inform. Theory*, to be published.
[23] W. J. Rosenberg, "Structural properties of convolutional codes," Ph.D. dissertation, Dep. Syst. Sci., Sch. Eng. Appl. Sci., Univ. California, Los Angeles, 1971.
[24] J. A. Heller and I. M. Jacobs, "Viterbi decoding for satellite and space communication," this issue, pp. 835–848.
[25] A. R. Cohen, J. A. Heller, and A. J. Viterbi, "A new coding technique for asynchronous multiple access communication," this issue, pp. 849–855.

Andrew J. Viterbi (S'54–M'58–SM'63) was born in Bergamo, Italy, on March 9, 1935. He received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1957, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1962.

While attending M.I.T., he participated in the cooperative program at the Raytheon Company. In 1957 he joined the Jet Propulsion Laboratory where he became a Research Group Supervisor in the Communications Systems Research Section. In 1963 he joined the faculty of the University of California, Los Angeles, as an Assistant Professor. In 1965 he was promoted to Associate Professor and in 1969 to Professor of Engineering and Applied Science. He was a cofounder in 1968 of Linkabit Corporation of which he is presently Vice President.

Dr. Viterbi is a member of the Editorial Boards of the PROCEEDINGS OF THE IEEE and of the journal *Information and Control*. He is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu and has served on several governmental advisory committees and panels. He is the coauthor of a book on digital communication and author of another on coherent communication, and he has received three awards for his journal publications.

Burst-Correcting Codes for the Classic Bursty Channel

G. DAVID FORNEY, JR., MEMBER, IEEE

***Abstract*—The purpose of this paper is to organize and clarify the work of the past decade on burst-correcting codes. Our method is, first, to define an idealized model, called the classic bursty channel, toward which most burst-correcting schemes are explicitly or implicitly aimed; next, to bound the best possible performance on this channel; and, finally, to exhibit classes of schemes which are asymptotically optimum and serve as archetypes of the burst-correcting codes actually in use. In this light we survey and categorize previous work on burst-correcting codes. Finally, we discuss qualitatively the ways in which real channels fail to satisfy the assumptions of the classic bursty channel, and the effects of such failures on the various types of burst-correcting schemes. We conclude by comparing forward-error-correction to the popular alternative of automatic repeat-request (ARQ).**

INTRODUCTION

MOST WORK in coding theory has been addressed to efficient communication over memoryless channels. While this work has been directly applicable to space channels [1], it has been of little use on all other real channels, where errors tend to occur in bursts. The use of interleaving to adapt random-error-correcting codes to bursty channels is frequently proposed, but turns out to be a rather inefficient method of burst correction.

Of the work that has gone into burst-correcting codes, the bulk has been devoted to finding codes capable of correcting *all* bursts of length B separated by guard spaces of length G. We call these *zero-error* burst-correcting codes. It has been realized in the past few years that this work too has been somewhat misdirected; for on channels for which such codes are suited, called in this paper *classic bursty channels*, much more efficient communication is possible if we require only that *practically all* bursts of length B be correctible.

The principal purpose of this paper is tutorial. In order to clarify the issues involved in the design of burst-correcting codes, we examine an idealized model, the classic bursty channel, on which bursts are never longer than B nor guard spaces shorter than G. We see that the inefficiency of zero-error codes is due to their operating at the zero-error capacity of the channel, approximately $(G - B)/(G + B)$, rather than at the true capacity, which is more like $G/(G + B)$. Operation at the true capacity is possible, however, if bursts can be treated as erasures; that is, if their locations can be identified. By the construction of some archetypal schemes in which short Reed–Solomon (RS) codes are used with interleavers, we arrive at asymptotically optimal codes of

Paper approved by the Communication Theory Committee of the IEEE Communication Technology Group for publication without oral presentation. Manuscript received May 10, 1971.
The author is with Codex Corporation, Newton, Mass., 02195.

either the burst-locating or zero-error type. (The usefulness of RS codes in this situation is seen to be due to their being optimal in a sense similar to that in which optimal burst-correcting codes are optimal.) Finally, we note that the sensitivity to errors in the guard space which characterizes most known burst-locating schemes is avoidable at a minor cost in guard-space-to-burst ratio.

When we turn to typical real channels, however, the superiority of one error-correcting scheme over another is much harder to assert. We discuss qualitatively what may be expected with various schemes when the channel does not fit the idealized model. Finally, we compare forward error correction to the more widely used method of automatic repeat-request (ARQ).

Classic Bursty Channel

The classic bursty channel is an idealization of the experimental fact that on most channels transmission is poorer at some times than at others. In this paper a classic bursty channel is defined as one having the following properties.

1) The channel (like the girl with the curl) has two modes of behavior: "When she was good, she was very, very good, but when she was bad, she was horrid." We call these two states *burst* and *guard space*. In the burst state, channel outputs carry no information about the inputs. In the guard space, we shall initially assume that channel outputs are error-free; later we shall allow some small background error probability p. We further distinguish between the cases where the channel state is unknown at the receiver (the usual case), and where the channel state is known, when bursts may be regarded as erasure bursts. In the latter case we speak of a *classic erasure-burst channel.*

2) The channel never stays in burst mode for more than B symbols, nor does it ever stay in the guard space mode for fewer than G symbols.

The two-state assumption [2], while artificial, is not a crippling idealization of most actual channels, especially when the possibility of guard space errors is encompassed. It is the second assumption that is the Achilles heel of this model; yet, as we shall see later, the making of this assumption is in a sense unavoidable.

Capacity Theorems

The capacity C of a channel is defined as the maximum continuous rate of transmission for which arbitrarily low error probability is achievable. Its zero-error capacity C_0 is defined as the maximum rate for which zero-error probability is achievable. We recall that on all memoryless channels except erasure-type channels, C_0 is strictly less than C; and in fact that usually (on any completely connected channel) C_0 is zero.

It is clear intuitively that, since the classic bursty channel may wipe out B out of every $G + B$ transmitted symbols, its capacity in symbols per transmitted symbol must be bounded by $G/(G + B)$. Furthermore, it is evident that this capacity bound retains its validity even when feedback is permitted. An information-theoretic proof of these facts is easily constructed.

In this section we shall derive bounds on zero-error capacity under the sole restriction that infinite buffering not be allowed at the encoder. We encourage the reader not to be intimidated by the theorem-proof format, which we have adopted for brevity and for conceptual clarity; the theorems are simple (and old), and the proofs are elementary. We have organized the argument to show that there is a fundamental relationship between optimum codes for the classic bursty channel and the maximum distance separable codes [3], [4], such as the RS codes [5]. We have also centered our attention on burst-erasure correction; not only does this approach lead easily and naturally to the usual burst-error-correction results, but it also clarifies what is going on in burst-locating codes.

We consider two different types of codes. In order to show the relation between these capacity theorems and well-known block code results, we first consider (n, k) block codes, in which k information symbols determine n encoded symbols. (All symbols will be taken as q-ary for some integer q; one notable aspect of the major results is their independence of q.) The rated R of a block code is k/n. Second, in order to show how general these theorems are, we consider any encoder of finite memory, say ν q-ary memory elements, and we allow the number of inputs $k(t)$ accepted and outputs $n(t)$ put out on the channel in t time units to be any monotonic functions of t, providing only that the limit

$$\lim_{t\to\infty} \frac{k(t)}{n(t)} = R$$

exists. We call this limit the code rate R, and we call the code an (R, ν) finite-memory code. We then appeal to the following simple lemmas.

Lemma 1: In an (n, k) block code, for any set of $k - \tau$ code positions, there are at least q^τ code words all of which have identical symbols in those positions.

Proof: There are q^k words in the code, but only $q^{k-\tau}$ possibilities for the symbols in any $k - \tau$ positions, so at least one possibility must be repeated q^τ or more times.

Lemma 1(a): In an (R, ν) finite-memory code, for any set of $k(t) - \tau - \nu$ positions among the $n(t)$ outputs before time t, there are at least q^τ code words all of which leave the encoder memory in the same state at time t, and all of which have identical symbols in those positions.

Proof: There are $q^{k(t)}$ code words of length $n(t)$ and $\leq q^\nu$ encoder memory states, so there must be at least $q^{k(t)-\nu}$ code sequences all of which leave the encoder in some identical state. There are only $q^{k(t)-\nu-\tau}$ possibilities for the symbols in any $k(t) - \nu - \tau$ positions, so at least one possibility must be repeated at least q^τ times among any set of at least $q^{k(t)-\nu}$ code words.

The *minimum distance* d of a block code is defined as the minimum number of positions in which two words differ; we also define the *minimum span* S of a code as the minimum number of consecutive positions outside of which two words are the same. Trivially $d \leq S$. Lemma 1 immediately yields the following corollary.

Corollary 1: In an (n, k) block code, $d \leq S \leq n - k + 1$.

Proof: By Lemma 1 there are at least two words which are the same in the first $k - 1$ positions.

Block codes for which $d = n - k + 1$ are called *maximum distance separable codes*. The only known general class of such codes is the RS codes. These are q-ary codes with blocklengths q or less, where q is a prime power [6, pp. 21–29]; hence only the nonbinary RS codes are nontrivial. Singleton [3] refers to constructions which give codes of length $q + 1$, for q a prime power, and proves that $k \leq q - 1$, $n - k \leq q - 1$ for any maximum distance separable code.

The class of block codes for which $S = n - k + 1$ is much greater. The RS codes of course satisfy this equality; but, more generally, any cyclic block code satisfies $S = n - k + 1$ (because any $n - k$ consecutive erasures can always be cyclically permuted into the check positions).

Now consider erasure correction. A pattern of erasures is called *correctible* if no two code words have the same symbols in all positions outside the erasure pattern.

Corollary 2: In an (n, k) block code no pattern of more than $n - k$ erasures is correctible.

That is, every erasure to be corrected requires one check symbol. For example, a rate-1/2 block code can correct no more than $n/2$ erasures.

Theorem 1: Any (R, ν) finite-memory code capable of correcting erasure bursts of length B separated by guard spaces of length G has rate $R \leq G/(G + B)$. That is, the zero-error capacity of the classic erasure-burst channel is bounded by $C_0 \leq G/(G + B)$.

Proof: By Lemma 1(a), if there are more than $n(t) - k(t) + \nu$ burst symbols in the first $n(t)$ received symbols, then there are at least two code words which are identical in the $k(t) - \nu - 1$ or fewer guard space symbols and which leave the encoder in the same state. These two words therefore cannot be distinguished on the basis of the first $n(t)$ received symbols, and since they both leave the encoder in the same state no information from future received symbols can help to distinguish them. Thus a decoding error will occur for one or the other of these code words if the number $N_x(t)$ of burst symbols in the first $n(t)$ symbols exceeds $n(t) - k(t) + \nu$, or if the burst density $N_x(t)/n(t)$ satisfies

$$\frac{N_x(t)}{n(t)} > 1 - \frac{k(t)}{n(t)} + \frac{\nu}{n(t)}.$$

Let the channel alternate forever between B burst bits and G guard space bits; then as $t \to \infty$ the burst density approaches $B/(G + B)$, while the right side approaches $1 - R$ for any finite ν, so that if $B/(G + B) > 1 - R$ there is a t large enough that the inequality is satisfied. Hence we must have $B/(G + B) \leq 1 - R$ to guarantee zero errors.

Q.E.D.

For example, no rate-1/2 finite-memory code can have a guard-space-to-burst ratio of better than one-to-one.

We now take up error correction. Two patterns of errors are called *simultaneously correctible* if no two code words differ only in the positions covered by the union of the two error patterns. For if this condition holds, then there is no common received word into which two different code words can be transformed by changing the first code word in some of the positions of the first error pattern and the second in some of the positions of the second; while if it does not hold, there is such a common received word. (Note that not every position in an error pattern is required actually to be in error.) The close relation of error correction to erasure correction is shown by the following lemma.

Lemma 2: Any partition of an uncorrectible erasure pattern results in two disjoint error patterns which are not simultaneously correctible.

Proof: From the definition of an uncorrectible erasure pattern there are two code words identical outside the union of the two error patterns.

This is the reason why it always takes twice as much redundancy to correct errors as erasures. For example, in block codes we need two check symbols for every error to be corrected.

Corollary 3: In any (n, k) block code there are two error patterns of $\lfloor (n - k)/2 \rfloor + 1$ or fewer positions which are not simultaneously correctible; that is, we can guarantee correction of no more than $\lfloor (n - k)/2 \rfloor$ errors.

Note: $\lceil x \rceil$ is the least integer not less than x, and $\lfloor x \rfloor$ is the greatest integer not greater than x.

Proof: By Corollary 2 there is an uncorrectible erasure pattern of $n - k + 1$ erasures, which may be partitioned into two subsets of $\lceil (n - k + 1)/2 \rceil$ and $\lfloor (n - k + 1)/2 \rfloor$ positions which are not simultaneously correctible, by Lemma 2. The proof is completed by noting that

$$\left\lfloor \frac{n - k + 2}{2} \right\rfloor = \left\lceil \frac{n - k + 1}{2} \right\rceil.$$

For example, a rate-1/2 block code can correct no more than $n/4$ symbol errors. For burst correction, we have the following corollary.

Corollary 4: If some pattern of erasure bursts of length B separated by guard spaces of length G is uncorrectible, then there are at least two patterns of error bursts of length $\leq \lceil B/2 \rceil$ separated by guard spaces of length $\geq G + \lfloor B/2 \rfloor$ which are not simultaneously correctible.

Proof: Partition the uncorrectible erasure bursts into error bursts of sizes $\lceil B/2 \rceil$ and $\lfloor B/2 \rfloor$ separated by guard spaces of size $G + \lfloor B/2 \rfloor$ and $G + \lceil B/2 \rceil$, as shown in Fig. 1, and apply Lemma 2.

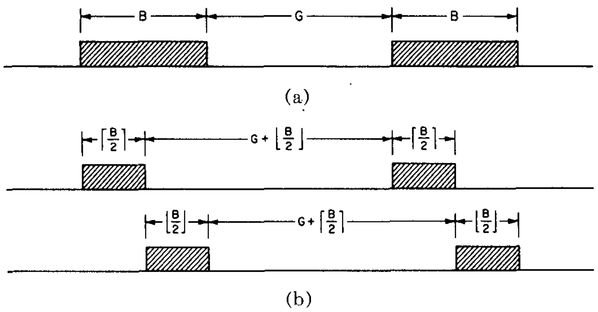

Fig. 1. (a) Uncorrectible erasure burst pattern. (b) Two error burst patterns that are not simultaneously correctible.

Consequently we have the following Theorem.

Theorem 2: Any (R, ν) finite-memory code capable of correcting all error bursts of length B separated by guard spaces of length G has $R \leq (G - B)/(G + B)$. That is, the zero-error capacity of the classic bursty channel is bounded by $C_0 \leq (G - B)/(G + B)$.

Proof: By Corollary 4 a code satisfying the assumption is capable of correcting all erasure bursts of length $2B$ separated by guard spaces of length $G - B$. But then Theorem 1 implies that $R \leq (G - B)/(G - B + 2B)$.

For example, no rate-1/2 finite-memory code can correct all bursts of length B unless they are separated by guard spaces of at least $3B$.

Theorem 2 is known as the Gallager bound. Similar theorems were proved by Reiger [7], for linear block codes of length $B + G$, and by Wyner and Ash [8], for convolutional codes with a decoding constraint length of $B + G$. Gallager [9, pp. 289–290] first proved the result in general, assuming only finite decoding delay. Our alternate assumption here of finite encoder memory is possibly more to the point, since it shows that the limitations are inherent in any realizable code, apart from the realizability of the decoder. Massey [10] sketched a still more general proof showing that an error-free decision on the entire (perhaps infinite) transmitted sequence on the basis of the entire received sequence was possible only if $R \leq (G - B)/(G + B)$.

To summarize, we have shown that the capacity of the classic bursty channel is bounded by $C \leq G/(G + B)$, that the zero-error capacity of the classic erasure-burst channel is bounded by the same quantity, but that the zero-error capacity of the classic bursty channel is bounded by $C_0 \leq (G - B)/(G + B)$. The difference between signaling at arbitrarily low probability of error and at zero probability of error on the classic bursty channel can therefore be quite large; the guard-space-to-burst ratio must exceed

$$\frac{G}{B} \geq \frac{1 + R}{1 - R}$$

for zero error, but only

$$\frac{G}{B} \geq \frac{R}{1 - R}$$

for arbitrarily low error. For $R = 1/2$, for example, $G \geq 3B$ for zero error, whereas $G \geq B$ for arbitrarily small error. We shall see in the next section that these bounds can be effectively achieved when G and B are large and there are no errors in the guard space.

Archetypal Coding Schemes

We shall now exhibit some coding schemes which approximately meet the bounds of the previous section when the guard space is error-free. These schemes are offered as theoretical archetypes of various classes of schemes of practical interest, rather than as practical techniques directly applicable to real channels. We do feel that they bring out clearly the important issues in burst correction.

It is evident from intuitive capacity arguments that any code must introduce constraints over a number of channel symbols of the order of $B + G$, since only over this time span can we be sure of channel behavior not too much worse than average.

Interleaving is the most obvious method of obtaining long code constraint lengths. Sophisticated designers have commonly avoided it, since the usual interleaving schemes proposed by unsophisticated designers are rather poor burst correctors. However, it is still more sophisticated to observe (with [8]) that there is nothing objectionable per se in schemes which combine short codes with interleaving, as long as the decoder operates sensibly.

The usual type of interleaver is a block interleaver, in which, for example, bits are laid down in the rows of a $B \times N$ matrix, and read out from the columns. In this paper we shall use a somewhat simpler and more effective type of interleaver, which we have called [11] a periodic (or convolutional) interleaver. (Similar interleavers were independently proposed by Cowell and Burton [12] and Ramsey [13].) Schematically, as illustrated in Fig. 2, symbols to be interleaved are arranged in blocks of N (by a serial/parallel conversion, if necessary). The ith symbol in each block is delayed by $(i - 1)NB'$ time units through a $(i - 1)$ B' stage shift register clocked once every N symbol times, where $B' = B/N$. (A time unit thus corresponds to the transmission of a block of N symbols.) Output bits may be serialized for channel transmission. At the receiver, groups of N symbols are reblocked, and the ith symbol in each block is delayed by $(N - i)NB'$ time units through an $(N - 1)B'$ stage shift register.

We call this a $B \times N$ interleaver. Correspondingly there exists a similar but inverse $B \times N$ deinterleaver, also illustrated in Fig. 2. The combination has the following properties.

1) All symbols receive a total delay of $(N - 1)B'$ time units, or $N(N - 1)B' = (N - 1)B$ symbol times (plus the channel delay).

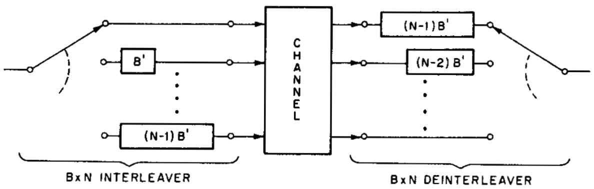

Fig. 2. Periodic interleaver and corresponding deinterleaver ($B' = B/N$).

2) The memory requirements at transmitter and receiver are $N(N-1)B'/2$, or $N(N-1)B' = (N-1)B$ total.

3) A single channel burst affecting B' or fewer blocks ($B - N + 1$ or fewer symbols) passes through the deinterleaver in such a way as to affect only one of the N deinterleaver output streams at a time. See Fig. 3. Repeated bursts separated by guard spaces of $(N-1)B'$ or more blocks ($(N-1)B + N - 1$ symbols) also affect only one of the N output streams at a time.

4) A channel burst affecting kB' or fewer blocks affects no more than k of the N deinterleaver output streams at a time. Repeated bursts separated by guard spaces of $(N-k)B'$ or more blocks affect only k of the output streams at a time. See Fig. 4.

The unsophisticated approach would now be to let the input symbols in any one block be a code word from a block code of length N capable of correcting up to t symbol errors. For example, the rate-1/2 binary (24, 12) Golay code corrects up to 3 bit errors. With this code and a $B \times 24$ interleaver we can correct all bursts of length approximately $3B$ separated by guard spaces of approximately $21B$. This 7-to-1 guard-space-to-burst ratio is far inferior to even the 3-to-1 ratio required for zero-error capacity.

We should note, however, that if the location of bursts can be detected, then use of a cyclic symbol-erasure-correcting code is perfectly respectable. For example, the (24, 12) Golay code can correct any 12 cyclically consecutive erasures. We see that with this code and a $B \times 24$ interleaver we can correct all bursts of length approximately $12B$ separated by guard spaces of approximately $12B$, which is the best we can hope for. The reason this technique works well for burst-erasure correction but not for burst-error correction is that a cyclic binary code achieves the bound $S \leq n - k + 1$ but generally falls far short of the bound $d \leq n - k + 1$.

These observations lead us to look for a maximum distance separable code for burst-error correction. Let q be a prime power and let b be an integer such that $q^b \geq N$; then there exists an RS code of length N with supersymbols consisting of b q-ary symbols. Then on each of K input streams we take consecutive segments of b symbols as the information supersymbols in an (N, K) RS code, and generate N code supersymbols, which then form the input streams to the interleaver. At the decoder we perform error-correction on the N deinterleaver out-

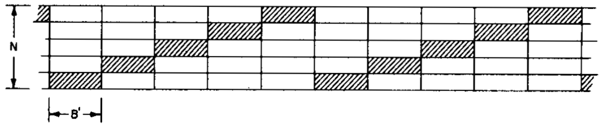

Fig. 3. Appearance of bursts of B' blocks separated by guard spaces of $(N-1)B'$ blocks in N deinterleaver output streams.

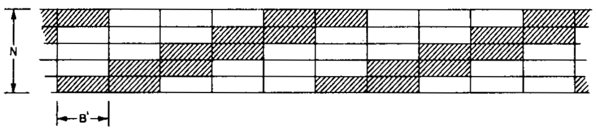

Fig. 4. Appearance of bursts of kB' blocks separated by guard spaces of $(N-k)B'$ blocks in N deinterleaver output streams.

puts after reblocking into supersymbols, as illustrated in Fig. 5.

We can correct up to $(N-K)/2$ errors with this code; therefore if we use $B \times N$ interleavers we can correct bursts of length approximately $(N-K)B/2$ separated by guard spaces of approximately $(N+K)B/2$. (The approximation comes from the blocking of input data into code blocks of length bN symbols, and clearly becomes insignificant if B is substantially larger than bN.) Hence we obtain guaranteed burst correction with a guard-space-to-burst ratio of nearly $(N+K)/(N-K) = (1+R)/(1-R)$, in agreement with the zero-error-capacity bound for the classic bursty channel. In other words, for $B \gg N \log_q N$, there is a code of rate $R = K/N$ which meets the bound, so the bound is asymptotically tight. (Burton has recently come upon a similar scheme, with $N - K = 2$, from a different direction [14]. Peterson [15, pp. 198–199] suggested using very long noninterleaved RS codes for burst correction; such codes are asymptotically optimum but more complex and less related to other burst-correction schemes than those described here.)

In exactly the same way, the use of an erasure-correcting (N, K) RS code with a $B \times N$ interleaver on the classic erasure-burst channel succeeds in correcting all erasure bursts of length approximately $(N-K)B$ separated by guard spaces of approximately KB, for a guard-space-to-burst ratio of nearly $K/(N-K) = R/(1-R)$, which is the zero-error-capacity bound for the classic erasure-burst channel.

Since the erasure zero-error-capacity bound equals the capacity bound, this suggests that we could approach the capacity of the classic bursty channel with a similar scheme if the decoder could only tell with high probability where the bursts were. But this is really not so difficult. Again we suppose an (N, K) RS code used with a $B \times N$ interleaver. We suppose initially there has been no burst for some time. When a burst begins, as soon as an error appears in the bottommost stream it is detected, since an (N, K) RS code can detect up to $(N-K)$ symbol errors. At this point the start of the burst has been located with probability $(1 - q^{-m})$ to be within the

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.