New results on serial concatenated and accumulated-convolutional turbo code performance

Audrey M. VITERBI*
Andrew J. VITERBI*

Abstract

Previous methods for analyzing serial concatenated turbo codes employing union error bounds are extended to determine the complete output weight enumeration function of the code; this provides the opportunity to employ a more refined bound due to Polytrev, with considerably improved results limited, however, to block lengths of about 256 bits by computational constraints. The method is then applied to a new class of "accumulated-convolutional" codes, which is a simple special subclass of serial concatenated codes inspired by the "repeat-accumulate" codes of Divsalar et al. Performance appears to be superior to that of conventional codes and results are obtained for much longer block lengths, with impressive results in regions approaching channel capacity.

Key words: Error correcting code, Turbo code, Error probability, Convolutional code.

NOUVEAUX RÉSULTATS SUR LES PERFORMANCES DES TURBO CODES SÉRIES CONSTRUITS À PARTIR DE CODES CONVOLUTIFS AVEC ACCUMULATION

Résumé

Des méthodes usuelles employant la borne de l'union pour analyser les turbo codes concaténés en série sont étendues pour déterminer la fonction complète d'énumération de poids en sortie d'un code ; on peut utiliser une borne plus fine proposée par Polytrev, borne qui améliore considérablement les résultats. Cette borne est cependant limitée, pour des raisons de complexité de calcul à des blocs de longueur de l'ordre de 256. La méthode est ici appliquée à une nouvelle classe de codes « convolutifs accumulés », qui est une simple classe particulière de codes concaténés en série, inspirés des codes à « répétition-accumulation » de Divsalar et al. Les performances obtenues sont supérieures à celles des codes classiques et les résultats, qui sont obtenus pour des blocs bien plus longs, sont impressionnants dans la région approchant la capacité du canal.

Mots clés : Code correcteur erreur, Turbo code, Probabilité erreur, Code convolutif.

Contents

I. *Introduction*
II. *Weight enumeration functions and rate-weight distributions*
III. *The tangential-sphere improved union bound*
IV. *Accumulated convolutional codes*
V. *Conclusion*
Appendix
References (10 ref.)

I. INTRODUCTION

Ever since Shannon established the discipline of information theory, it has been known that codes exist for which error probability decreases monotonically to zero as the length of the code grows to infinity. Even stronger results were established in the fifties and sixties indicating that, for most codes, the error probability decreases exponentially with the length of the code for all rates less than capacity. This optimistic scenario was, however, clouded by the facts that no well defined code construction seemed to exhibit this performance, and more importantly, to achieve it the decoding algorithm operations generally grow exponentially with code length. Early emphasis was on algebraic code construction and corresponding decoding algorithms, which fell short of the ideal performance. Later convolutional codes with sequential decoding algorithms afforded the possibility of long codes with exponentially decreasing error probabilities, but with two serious drawbacks : this performance was limited to rates below the computational cutoff rate, considerably below chan-

* QUALCOMM Incorporated, San Diego, CA 92121, USA

nel capacity, and the probability of failure to decode exhibited a Pareto distribution with respect to available buffer memory. Nevertheless, convolutional codes gradually became the standard means for reducing the required channel signal-to-noise ratio (E_b/N_0) in wireless communication, magnetic recording and other applications wherein the interference was primarily additive Gaussian noise. With short code constraint lengths, the optimal decoder could be implemented practically, and performance, while far from the ideal, achieved a coding gain in terms of dB of the E_b/N_0 of a Gaussian channel which was approximately half the difference between uncoded performance and that corresponding to channel capacity.

Further improvements were achieved by concatenating such a relatively short convolutional code as an inner code, with an outer block code, generally the Reed-Solomon algebraic code which employed a higher-order alphabet whose size corresponded approximately to 2^K where K is the constraint length of the inner convolutional code. This approach became the standard for NASA interplanetary missions as well as for satellite direct broadcasting and achieved very low error probabilities down to E_b/N_0 approximately corresponding to cutoff rate. To approach closer to capacity, much longer convolutional constraint lengths and outer block lengths were implemented, but their complexity exceeded practical limits for all but well funded space missions. A major limitation of these concatenated decoder implementations was the generation of hard decisions out of the inner decoder to be used as input to the outer decoder. Improvements were shown to result from replacing these hard decisions by soft decisions generated according to an appropriate metric.

Much better performance was demonstrated by empirical means in 1993 using parallel concatenated *turbo* codes with iterative soft decision decoding between components of the concatenated code. Specifically, Berrou et al [1] generated a posteriori probability (APP) soft decision metrics in decoding the first component and used these in decoding the second, in turn generating APP soft decisions for the second component which were fed back to soft decode the first component once again, continuing to iterate between components for a number of times. The unexpectedly excellent results, using relatively simple codes but large interleavers between component codes, gave rise to a new era in error-correcting coding with a totally different perspective than what had been the traditional view. The first theoretical results on turbo codes, due to Benedetto et al [2] showed that bit error probability for optimally decoded parallel concatenated codes (at high E_b/N_0) decreased only inversely with block length, rather than exponentially. Benedetto et al [3] later showed that the conventional serially concatenated code, with an appropriate interleaver, when optimally decoded, exhibited an error probability which for large E_b/N_0 decreases as a power of the interleaver length approximately equal to half the free distance of the outer convolutional code.

While much progress has been made in the selection, implementation and evaluation of both parallel and serial turbo codes, there remain two important open issues :

a) determination of whether iterative soft decision APP decoding approaches, with increasing numbers of iterations, the optimal maximum likelihood decoding of the composite code ;

b) analytical evaluation of optimal decodeur error probabilities, particularly for very long interleavers at E_b/N_0 approaching channel capacity.

The first issue is still open. This paper deals with the second issue. While union bounds have been evaluated numerically [4], they diverge at the cutoff rate. Here we obtain a complete determination of the output weight enumeration function and, using this and a recent refinement on the union bound, we obtain tighter bounds on error probability which remain finite for all E_b/N_0, and small for values well below that corresponding to cutoff rate. We obtain extensive results for a new class of codes, which we call "accumulated convolutional codes," inspired by the recent surprising results of Divsalar et al [5]

II. WEIGHT ENUMERATION FUNCTIONS AND RATE-WEIGHT DISTRIBUTIONS

For any binary block code of K inputs and N outputs, we may define the input-output weight enumeration function (IOWEF)

$$T(W, Z) = \sum_{k=1}^{K} \sum_{n=1}^{N} T_{kn} W^k Z^n \quad (1)$$

where T_{kn} is the number of codewords of (output) weight n generated by input sequences of weight k.

We also define

$$A_n(W) = \sum_k T_{kn} W^k \quad (2)$$

as the *input* weight enumeration function, being the enumerating polynomial of all input weights which generate codewords of weight n, and define

$$B_k(Z) = \sum_n T_{kn} Z^n \quad (3)$$

as the *output* weight enumeration function, being the enumerating polynomial of all (output) codewords generated by an input of weight k.

Note also that

$$T(W, Z) = \sum_n A_n(W) Z^n = \sum_k B_k(Z) W^k \quad (4)$$

For an additive white Gaussian noise (AWGN) channel with antipodal binary inputs $\pm\sqrt{E_s}$, where $E_s = E_b R$, with R being the code rate, it is well known [6] that the union-Chernoff bounds on block and bit error probabilities are, respectively,

$$P_E < T(W = 1, Z = \mathbf{e}^{-E_s/N_0})$$

$$(5) \qquad P_b < \frac{dT}{dW}(W = 1, Z = \mathbf{e}^{-E_s/N_0})$$

The weight enumeration functions of turbo codes, whether parallel or serially concatenated, can be obtained from their component codes by employing the artifice of Benedetto et al which involved taking the ensemble average over all possible interleaver permutations. Figures 1a and 1b show the typical configurations

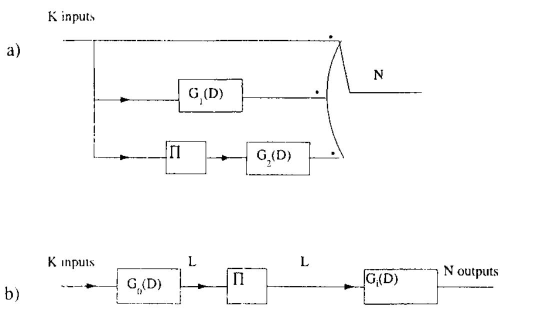


FIG. 1. — Turbo code examples a) parallel concatenated, b) serial concatenated. – Note each $G(D)$ may be a matrix of polynomials, representing a rate k/n code

Exemples de turbo codes

for parallel and serial concatenated codes. In each case the interleaver permutation is assumed to be selected randomly. For good performance, $G_1(D)$ and $G_2(D)$ forparallel, and $G_i(D)$ for serial, turbo codes must be recursive encoders. Then, as has been shown by Benedetto et al [2], for parallel concatenated codes, since that the overall code is systematic (information bits transmitted uncoded), the ensemble average IOWEF is given by

$$(6) \qquad T_p(W, Z) = \sum_{k=1}^{K} W^k Z^k \frac{B_k^{(1)}(Z)\, B_k^{(2)}(Z)}{\binom{K}{k}}$$

where the degree of the polynomial in Z is

$$N = K + \max_k \deg\left[B_k^{(1)}(Z)\right] + \max_k \deg\left[B_k^{(2)}(Z)\right]$$

and the code rate $R = K/N$. The form of (6) follows from the fact that for each output generated by $G_1(D)$, after permutation the second encoder $G_2(D)$ can generate any one of $\binom{K}{k}$ outputs, corresponding to one of as many permutations of k input *ones* among K input symbols.

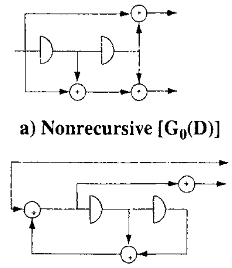
For the serial concatenated codes [3], the interleaver ensemble average IOWEF is given by

$$(7) \qquad T_s(W, Z) = \sum_{\ell = d_{min}}^{L} \frac{A_\ell^{(0)}(W)\, B_\ell^{(i)}(Z)}{\binom{L}{\ell}}$$

where L is the size of the interleaver, d_{min} is the minimum weight of the outer code, K/L is the rate of the outer code, L/N the rate of the inner code, and the overall rate is again $R = K/N$. The form of (7) follows from the fact that if ℓ is the weight of the outer code output, which after permutation is also the inner code input, for each such output there will be $\binom{L}{\ell}$ possible permutations for the inner code input sequence.

In [4] we employed (5) and (7) to obtain the union-Chernoff bounds for serially concatenated convolutional codes by computing the functions $A_\ell(W)$ *and* $B_\ell(Z)$. (We show the recursion equations, which we developed from the convolutional code state transition matrix, in Appendix I for the simple codes of Figures 2a and 2b.)

Encoder



a) Nonrecursive [G_0(D)]

b) Recursive [Gi(D)]

State Transition Matrix [$C(W,Z)$]

$$\begin{bmatrix} 1 & 0 & Z^2 & 0 \\ WZ^2 & 0 & W & 0 \\ 0 & Z & 0 & Z \\ 0 & WZ & 0 & WZ \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & WZ^2 & 0 \\ WZ^2 & 0 & 1 & 0 \\ 0 & WZ & 0 & Z \\ 0 & Z & 0 & WZ \end{bmatrix}$$

FIG. 2. — Component convolutional codes

Codes convolutifs élémentaires

We were able to obtain results in [4] for large values of L because W and Z took on numerical values as indicated by (5). The drawback was that the union bound only converges for E_b/N_0 above the value which corresponds to the cutoff rate. To go beyond this, it is necessary to use more refined bounds to be described in Section III, which require the determination of the output weight enumeration polynomial in Z, $T(W = 1, Z)$. Then for serially concatenated codes it follows from (7) that $A_\ell(1)$ can again be computed numerically,* but $B_\ell(Z)$ is a polynomial. Though it can still be computed recursively, the

* Note that we could equally handle P_b by replacing $A_l(1)$ in (8) by $[A_l(1+\epsilon) - A_l(1-\epsilon)]/2\epsilon$. However, we shall deal only with block error probability P_E throughout, because data transmission is becoming mainly oriented toward packet networks with the option to repeat incorrect packets.

recursion is among polynomials, which imposes a storage requirement proportional to L^2, hence limiting the interleaver size for which the output WEF can be practically computed to no greater than $L \leq 512$.

Then

$$(8) \qquad T_s(1, Z) = \sum_{\ell = d_{min}}^{L} \frac{A_\ell(1)\, B_\ell(Z)}{\binom{L}{\ell}} \triangleq \sum_{n}^{N} C_n Z^n$$

Note that the coefficients C_n, which denote the number of codewords of output weight n, grow exponentially with N. We define, therefore, a logarithmic measure which we call the *rate-weight distribution*

$$(9) \qquad r_n \triangleq ln\,(C_n)/N$$

which is the *rate* of the subcode of codewords of weight n. Figure 3 shows the rate-weight distribution for several

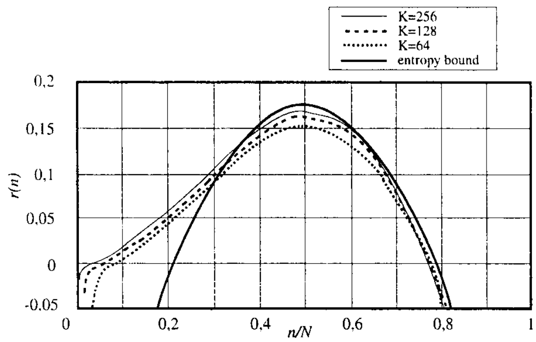


FIG. 3. — Rate-weight distribution for 4-state inner and outer codes of Figure 2

Rendement-distribution de poids pour les codes intérieurs et extérieurs à 4 états

values of K for a serially concatenated code of rate 1/4 with component codes which are rate 1/2 convolutional codes, respectively the outer code of Figure 2a and the inner code of Figure 2b. Also shown for comparison is the rate-weight distribution for the ensemble average of single randomly selected (N, K) codes ; this is simply

$$\bar{r}_n = \frac{1}{N} ln \left[2^K \binom{N}{n} 2^{-N} \right] = \frac{1}{N} ln \binom{N}{n} - (1 - R)\, ln\, 2$$

$$\sim H\,(n/N) - (1 - R)\, \ell n 2$$

where $H(.)$ is the entropy function to the base e.

Note the close convergence, for large N, of the serially concatenated code to the random code rate-weight distribution for mid level values of n, but the considerable divergence at low and high n.

III. THE TANGENTIAL-SPHERE IMPROVED UNION BOUND

As noted in Section I, traditional error probability bounds have generally concentrated on determining the exponent of the block or convolutional code error probabilities. Several variants of Gallager's exponential bounds [7, 8] have been applied to turbo codes, with mediocre improvements over the union bound. Given that, as was shown by Benedetto et al [2] [3], the error probability of turbo codes decreases only as a power of the interleaver (and hence block) size, these exponential bounds are not particularly suited to turbo code performance. Instead, as was first noted by Sason and Shamai [9], a more recent bound due to Poltyrev [10] yields far superior results. This so called *tangential-sphere* bound (TSB) is a refinement on the union bound which isolates those received signal-plus-noise vectors which fall outside a cone of angle θ from the correct transmitted signal vector. Denoting this N-dimensional cone $C(\theta)$, we have the bound on block error probability

$$(10) \qquad P_E < \Pr\,(E; z \in C(\theta)) + \Pr\,(z \notin C(\theta))$$

where E denotes error and z is the received N-dimensional noise vector. Using simple geometric constructions it is shown that if the code's output WEF is $T(Z) = \sum_{n=1}^{N} C_n Z^n$, then

$$P_E < \min_r \int_{-\infty}^{S} \frac{e^{-z_1^2/2}}{\sqrt{2\Pi}}$$

$$\cdot \left\{ \sum_{n \leq \frac{Nr^2/S^2}{1 + r^2/S^2}} C_n \left[Q\left(\sqrt{\frac{n}{N-n}}\,(S - z_1)\right) - Q\left(\frac{r}{S}(S - z_1)\right) \right] \right.$$

$$\gamma \left[\frac{N-2}{2}, \left(\frac{r^2}{S^2} - \frac{n}{N-n} \right) \frac{(S - z_1)^2}{2} \right]$$

$$(11) \qquad \left. + \bar{\gamma}\left(\frac{N-1}{2}, \frac{r^2\,(S - z_1)^2}{2 S^2} \right) \right\} dz_1 + Q(S)$$

where r is the radius* of the cone measured along the tangent to the signal sphere of radius $S = \sqrt{\frac{2E_s N}{N_0}}$, E_s is the signal energy of each of the N binary symbols and N_0 is the one-sided AWGN density, $Q(x) = \int_x^{\infty} \frac{e^{-z^2/2}}{\sqrt{2\Pi}} dz$ is the complementary error function, $\gamma(k, x) = \frac{1}{\Gamma(k)} \int_0^x t^{k-1} e^{-t} dt$ is the incomplete γ-function, and $\bar{\gamma}(k, x) = 1 - \gamma(k, x)$. Because this bound in this form is not particularly well documented, we provide the derivation in Appendix II. Suffice it to note that the $\bar{\gamma}$ term outside the summation and $Q(S)$ correspond to signal-plus-noise vectors outside the cone, the second term of (10). Also, if we let $r \to \infty$ and eliminate the cone, then

$$\lim_{r \to \infty} P_E(r) < \int_{-\infty}^{\infty} \frac{e^{-z_1^2/2}}{\sqrt{2\Pi}} \sum_{n \leq N} C_n Q\left(\sqrt{\frac{n}{N-n}}\,(S - z_1)\right) dz_1$$

$$= \sum_{n = d_{min}}^{N} C_n \int_{-\infty}^{\infty} Pr\left[z_2 > \sqrt{\frac{n}{N-n}}\,(S - z_1) \right] p(z_1)\, dz_1$$

* For a given N, the minimizing value of r/S remains constant for all values of S. Note also that for virtually all cases of possible interest, S being proportional to $\sqrt{N}$ renders $Q(S)$ absolutely neglible.

where z_1 and z_2 are independent unit variance zero mean Gaussian variables. We may express this as

$$\lim_{r \to \infty} P_E(r) < \sum_{n=d_{min}}^{N} C_n \, Pr\left(Z > \sqrt{\frac{n}{N-n}}\, S\right) \mathrm{d}Z$$

where $Z = z_2 + \sqrt{\frac{n}{N-n}}\, z_1$

for which $E(Z) = 0$, $Var(Z) = 1 + \frac{n}{N-n} = \frac{N}{N-n}$

so that

$$(12) \quad \lim_{r \to \infty} P_E(r) < \sum_{n=d_{min}}^{N} C_n \int_{\sqrt{\frac{nS}{N-n}}}^{\infty} \frac{\mathbf{e}^{-Z^2(N-n)/2N}}{\sqrt{2\Pi N/(N-n)}} \mathrm{d}Z$$

$$= \sum_{n=d\,min}^{N} C_n Q\left(\sqrt{nS/N}\right) \text{ where } S = \sqrt{2NE_s/N_0}$$

which is the traditional union bound. (Note that the union-Chernoff bound of (5) replaces $Q(x)$ by $\mathbf{e}^{-x^2/2}$.)

While Poltyrev's TSB bound appears to be somewhat more complex than the union bound, it is easily computable numerically. Figure 4 shows the TSB and union

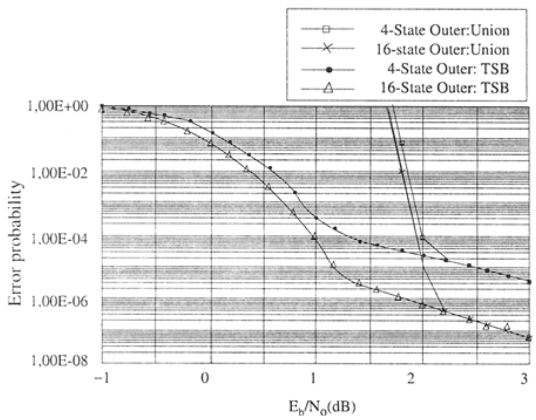


FIG. 4. — Error probability bounds for $K = 256$, $R = 1/4$ concatenated code with 4-state inner codes

Bornes de la probabilité d'erreur pour un code concaténé avec K = 256, R = 1/4 et des codes intérieurs à 4 états

bounds for $K = 256$, $N = 1024$, for the 4-state outer, 4-state recursive inner, serial concatenated code of overall rate 1/4 whose rate-weight distribution is shown in Figure 3. For comparison, the results are also shown in Figure 4 when the outer 4-state code is replaced with a 16-state nonrecursive rate 1/2 code whose tap generator polynomials are $1 \oplus D^2 \oplus D^3 \oplus D^4$ and $1 \oplus D \oplus D^4$. The computational memory burden in computing the polynomials $B_\ell(Z)$ make it difficult to go to higher K and N. In the next section, we consider an inner code for which the $B_\ell(Z)$ polynomial can be expressed in closed form so that much higher values of K can be considered.

IV. ACCUMULATED CONVOLUTIONAL CODES

We now consider a very simply described class of binary codes inspired by the interesting and surprisingly good results obtained by Divsalar et al [5] for the class of repetition-accumulator (RA) codes. These involved merely repeating each input information bit k times and, after interleaving, passing the output to an accumulator whose transfer function is $1/(1 \oplus D)$. We generalize from the trivial repetition code to an arbitrary rate R convolutional code as the outer code of a serially concatenated turbo code, but employing the accumulator as the inner code of unity rate, as shown in Figure 5.

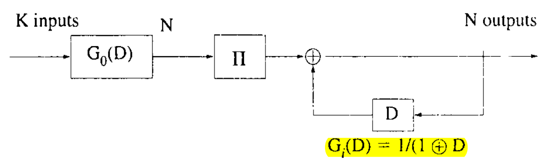


FIG. 5. — Accumulated-convolutional code

Codes convolutifs avec accumulation

First, following [5], we show that for this inner code,

$$(13) \qquad B_\ell(Z) = \sum_{n=\lceil \ell/2 \rceil}^{N} \binom{N-n}{\lfloor \ell/2 \rfloor} \binom{n-1}{\lceil \ell/2 \rceil - 1} Z^n$$

thus avoiding the necessity to compute the inner code polynomial recursively. Consider first the case when ℓ, the number of input ones to the inner code, is even. The first, third and all odd input ones will start a sequence of ones at the code output and the second, fourth and all even input ones will end it, as shown in Figure 6. Thus

Input	0	0	1	0	0	1	0	0	•	•	•	•	0	1	0	0	0	0	1	•	•	•	•
Output	0	0	1	1	1	0	0	0	•	•	•	•	0	1	1	1	1	1	0	•	•	•	•

FIG. 6. — Typical input and resulting output of accumulator $(1/(1 \oplus D)$

Entrée typique et sortie associée d'un accumulateur 1/(1 + D)

there are $\ell/2$ runs of output ones with as many starting input ones and ending input ones. Now if the total number of output ones (the codeword weight) is n, the number of ways that these can be distributed among the totality of output symbols, N, is $\binom{N-n}{\ell/2}\binom{n-1}{\ell/2-1}$, since the first binomial represents the number of ways the $\ell/2$ starting ones can be placed, while the second binomial is the number of ways the $\ell/2$ ending ones can be placed, the last one being fixed by the totality of output ones. On the other hand, if ℓ is odd, the number of starting ones which we are free to choose is $\lfloor \ell/2 \rfloor$, the integer value of $\ell/2$, because the last starting one is fixed by the number of output ones, n, which must end at the end of the block

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.