Fano Decoding with Timeout: Queuing Analysis

W. David Pan and Seong-Moo Yoo

In mobile communications, a class of variable-complexity algorithms for convolutional decoding known as sequential decoding algorithms is of interest since they have a computational time that could vary with changing channel conditions. The Fano algorithm is one well-known version of a sequential decoding algorithm. Since the decoding time of a Fano decoder follows the Pareto distribution, which is a heavy-tailed distribution parameterized by the channel signal-to-noise ratio (SNR), buffers are required to absorb the variable decoding delays of Fano decoders. Furthermore, since the decoding time drawn by a certain Pareto distribution can become unbounded, a maximum limit is often employed by a practical decoder to limit the worst-case decoding time. In this paper, we investigate the relations between buffer occupancy, decoding time, and channel conditions in a system where the Fano decoder is not allowed to run with unbounded decoding time. A timeout limit is thus imposed so that the decoding will be terminated if the decoding time reaches the limit. We use discrete-time semi-Markov models to describe such a Fano decoding system with timeout limits. Our queuing analysis provides expressions characterizing the average buffer occupancy as a function of channel conditions and timeout limits. Both numerical and simulation results are provided to validate the analytical results.

Keywords: Fano decoder, Pareto distribution, queuing analysis, semi-Markov model.

Manuscript received May 17, 2005; revised Jan. 23, 2006.

W. David Pan (phone: + 1 256 824 6642, email: dwpan@ece.uah.edu) and Seong-Moo Yoo (email: yoos@ece.uah.edu) are with the Department of Electrical and Computer Engineering, University of Alabama in Huntsville, Huntsville, Alabama, USA.

I. Introduction

The popularity of cellular telephones is driving the trend of mobile communications systems. A very significant challenge imposed by mobility is the need for designs that can efficiently use the battery power of mobile communication terminals. For example, a user that is located close to a base station should be able to operate at lower power than users roaming further away. Similarly, low-power decoding is desired when the available power is low or the user foresees the need for an extended use before battery re-charging. In these scenarios, variable-complexity (VC) algorithms are beneficial as their computational complexity can be varied (usually as a tradeoff with certain performance metrics) according to the changing needs, thereby enabling variable power consumptions of the system running the VC algorithm. In this work, we investigate VC channel decoders.

To decode a convolutional code over a memoryless channel, we compute the distances between the received code word **r** and all possible transmitted code words **v**. The log-likelihood function log $P(\mathbf{r}|\mathbf{v})$, denoted by $M(\mathbf{r}|\mathbf{v})$, is called the *metric* associated with the path (codeword) **v**, which is a measure of the closeness between the received sequence and the coded sequence. Note that the metrics are typically converted to integers in practical implementations [1]. The criterion for deciding between two paths through the trellis is to select the one having the larger metric. Among many decoding algorithms, the Viterbi algorithm (VA) is the most popular approach. It is a maximum-likelihood (ML) decoding algorithm that guarantees optimum decoding of the convolutionally encoded information sequence. That is, the decoder output selected is always the codeword that maximizes the conditional probability of the received sequences [1]. The VA is also well suited for hardware implementation due to its regular computation structure. On the other hand, a Viterbi

decoder is of fixed complexity in that it has to examine all branches in the code tree regardless of the channel conditions. Therefore, under high channel signal-to-noise-ratio (SNR) conditions, faster decoding is not available with the Viterbi algorithm when the received sequence is "easier" to decode. By contrast, sequential algorithms are of interest since they allow decoding complexity to vary with changing channel conditions. Sequential decoding of convolutional codes was introduced in 1957 by Wozencraft as a sub-optimal method of maximum likelihood sequence estimation with typically lower computational complexity than the Viterbi algorithm at high SNR's [1]. One version of sequential decoding algorithms, the Fano algorithm [2]-[4], is considered in this work. It was reported in [5] and [6] that very large-scale integrated (VLSI) chips based on the Fano algorithm achieved significantly lower energy consumption for an AWGN channel with a high SNR (≥ 6 dB), compared to the Viterbi decoder.

A Fano decoder explores one potential path at a time by examining its metric. If the metric value stays above a threshold S, the decoder moves back to examine other paths. If no path can be found whose metric value dips below the threshold, the threshold is then loosened by adding an increment Δ (i.e., $S \leftarrow S + \Delta$) and the decoder moves forward again with a lower threshold. To ensure no endless loop occurs, the threshold is tightened ($S \leftarrow S - \Delta$) as long as the decoder moves forward to a node as a first visit. The decoder will eventually reach the end of the tree. Interested readers may refer to page 620 in [1] for a detailed description of the Fano algorithm. The Fano algorithm is a variable-complexity algorithm with complexity varying with channel conditions—at a high channel SNR, the decoder tends to move very quickly to the end of the code tree, thereby finishing the decoding quickly. However, if the channel is very noisy, the decoder has to move along different paths of the code tree, resulting in a much higher number of computations. Thus, the Fano decoder incurs a non-deterministic processing delay since the number of computations performed in decoding a block of data is a random variable, which has been found by random coding analysis to follow closely the *Pareto* distribution [1], [7], [8], [9] and [10]. In real-time Fano decoding systems, a buffer is required to smooth out the variable processing delays of the Fano decoder. Furthermore, since the decoding complexity drawn by a certain Pareto distribution can become unbounded, a maximum (timeout) limit is often employed in a practical decoder to limit the worst-case decoding complexity [6], [11]. It is conceivable that the complexity of decoding some excessively corrupted blocks (coming from an extremely noisy channel) could become infinite if the decoder gets trapped in some erratic back and forth moves on the code tree. In practical implementations of Fano decoders, therefore, the decoding complexity is usually upper-bounded by a certain imposed limit on the number of either forward moves [11] or trace back moves [6]. Whenever the decoding time of a certain block reaches the imposed timeout limit, the decoding of the block is terminated. Those blocks that cannot be completely decoded will be discarded (lost). Several approaches for recovering lost blocks are feasible. For example, a partially decoded block may either be subjected to an outer code (for example, a Reed-Solomon code) for further error correction [1], [12], and [13], or an automatic repeat request (ARQ) will be triggered for the retransmission of the lost block [14], [15].

In this paper, we model the Fano decoding system with timeout as a discrete-time buffer (queue) system, with the goal of determining in theory the relations between the average buffer occupancy, decoding time, and channel conditions. To the best of our knowledge, there has been virtually no prior work similar to ours, which provides analytical expressions characterizing the average buffer occupancy of Fano decoding systems with timeout limits. Note that while we demonstrate that the capability of our analysis framework can accommodate the integration of the block retransmission into the discrete-time queue model, a complete analytical treatment of lost block recovery is beyond the scope of this paper.

The remainder of the paper is organized as follows. Section II introduces the discrete-time model that is suitable for characterizing the queuing behavior of Fano decoding systems. Section III presents a detailed queuing analysis of the average buffer occupancy based on a semi-Markov model. Section IV discusses the integration of the ARQ strategy into the discrete-time model and its impact on the block arrival rate. Numerical and simulation results are then discussed in section V. The paper is summarized in section VI.

II. Discrete-Time Model

In a practical Fano decoding system, the decoding complexity is usually measured in terms of elementary computational steps [13]. Therefore, we study the statistics of the Fano decoder's input buffer (queue) by using a discrete-time model, which is also employed in [15].

In this model, the time axis is partitioned into slots of equal length. We assume that the decoder can receive at most one new data block during a slot. The new blocks arrive at the decoder from the channel according to the Bernoulli process, that is, a slot carries an arriving block with probability λ and it is idle (no transmission) with probability $1 - \lambda$.

Another assumption is that the decoding time of a block is in chunks of length equal to the slot size. That is, the decoder can start and stop decoding only at the end of a slot. This approximation will yield an upper bound on the buffer occupancy. A block is allowed up to T slots for decoding, that is,

T is the timeout limit. If a block requires j slots for decoding ($j \le T$), it leaves the system at the end of the j-th slot after the beginning of its decoding, and the decoding of a new block starts (if there is a new block in the decoder's buffer) at the beginning of the following slot. If a block's decoding has to take longer than T slots, the decoder stops that block's decoding after T slots. This block cannot be completely decoded and typically will be discarded (lost).

To analyze the queue described above, we will make use of the following notation: $c_j \equiv Pr$ {decoding is completed in exactly j slots} and μ_j denotes the conditional probability that decoding is completed in j slots given that the decoding is longer than $j-1$ slots. This conditional probability is given by

$$\mu_j = \frac{c_j}{1-F_{j-1}}, \tag{1}$$

where F_j is the distribution of the decoding time:

$$F_j = \sum_{i=1}^{j} c_i, \quad j \ge 1. \tag{2}$$

As a special case, $F_0 = 0$.

It can be shown that

$$\prod_{j=1}^{k}(1-\mu_j) = 1-F_k. \tag{3}$$

The decoding time of Fano decoders follows the Pareto distribution:

$$P_F(\tau) = \Pr\{t > \tau\} = \left(\frac{\tau}{\tau_0}\right)^{-\beta}, \tag{4}$$

where τ_0 is the time such that $P_F(\tau_0) = 1$, and β (known as the Pareto exponent) can be approximated by [15]:

$$\beta = 17 - \frac{16r}{1-\log_2[1+2\sqrt{p(1-p)}]}, \tag{5}$$

where r is the code rate of the convolutional code used, and p is the crossover probability of a binary symmetric channel (BSC), which can be related to the channel SNR. Typically, a smaller β value means a lower SNR, and vice versa.

Given the slot duration T_r, we have

$$F_j = 1 - P_F(jT_r), \quad 1 \le j \le T, \tag{6}$$

and

$$c_j = F_j - F_{j-1}, \quad 1 \le j \le T. \tag{7}$$

III. Queuing Analysis

In order to determine the average buffer occupancy, we employ a semi-Markov model [15], [16], with the state of the queue being represented by the pair (n, t), where n is the number of blocks in the buffer including the block being decoded. Here, we assume the buffer is of infinite length, hence $0 \le n \le \infty$. Also, t is the number of slots of decoding already spent on the block that is being decoded currently. Whenever $t = T$ (the timeout limit), the block is removed from the buffer and considered lost, hence $0 \le t \le T-1$. The state transitions are illustrated in Fig. 1.

Let $p_{n,t}$ be the probability that the decoder's buffer contains n blocks, including the one being decoded by the decoder, which is in the t-th slot of decoding. The steady-state transition equations are given below.

For $n = 0$, we have

$$p_{0,0} = (1-\lambda)p_{0,0} + \sum_{j=1}^{T-1}\mu_j(1-\lambda)p_{1,j-1} + (1-\lambda)p_{1,T-1}. \tag{8}$$

Equivalently,

$$\lambda p_{0,0} = \sum_{j=1}^{T-1}\mu_j(1-\lambda)p_{1,j-1} + (1-\lambda)p_{1,T-1}. \tag{9}$$

And

$$p_{0,j} = 0, \quad 1 \le j \le T-1. \tag{10}$$

For $n = 1$, we have

$$p_{1,0} = \sum_{j=1}^{T-1}\mu_j\left[\lambda p_{1,j-1} + (1-\lambda)p_{2,j-1}\right] + \lambda p_{1,T-1} + (1-\lambda)p_{2,T-1} + \lambda p_{0,0}, \tag{11}$$

and

$$p_{1,j} = (1-\lambda)(1-\mu_j)p_{1,j-1}, \quad 1 \le j \le T-1. \tag{12}$$

For $n \ge 2$, we have

$$p_{n,0} = \sum_{j=1}^{T-1}\mu_j\left[\lambda p_{n,j-1} + (1-\lambda)p_{n+1,j-1}\right] + \lambda p_{n,T-1} + (1-\lambda)p_{n+1,T-1}, \tag{13}$$

and

$$p_{n,j} = (1-\mu_j)\left[(1-\lambda)p_{n,j-1} + \lambda p_{n-1,j-1}\right], \quad 1 \le j \le T-1. \tag{14}$$

We define the following generating functions:

$$P_j(z) = \sum_{n=1}^{\infty} p_{n,j} z^n, \quad 0 \le j \le T-1, \tag{15}$$

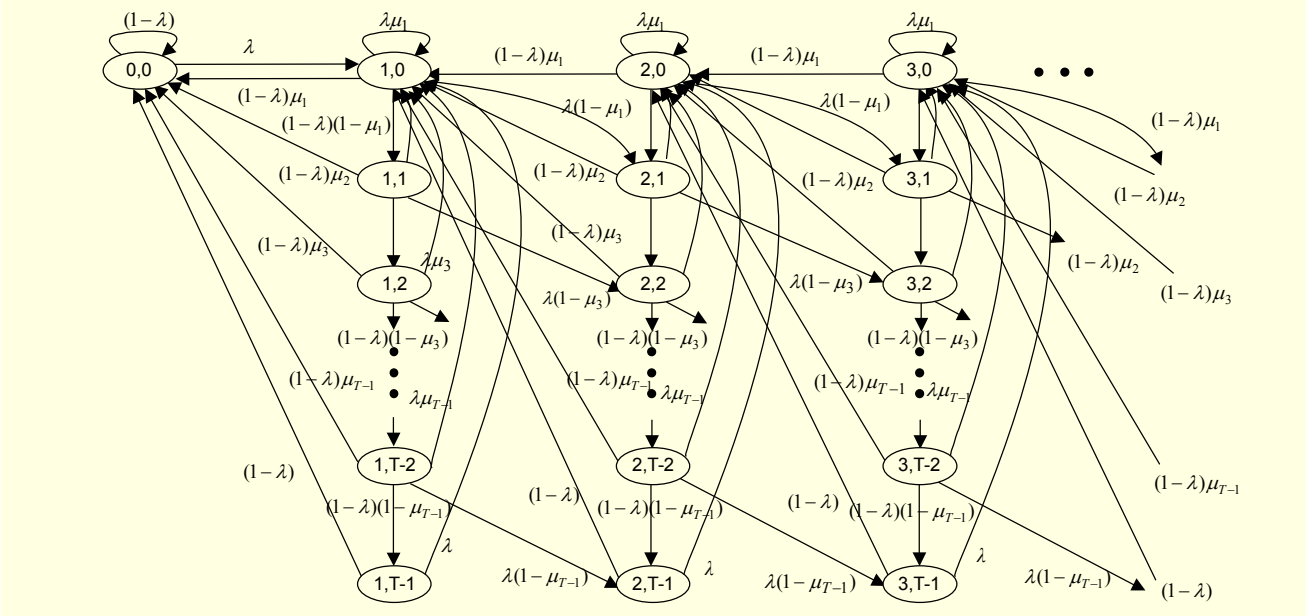

Fig. 1. Transitions of queue states. Note that all the states (n, T–1), where $n \geq 1$, have only two possible outgoing transitions with probabilities of 1– λ and λ, respectively.

and the generating function of the number of blocks in the buffer as

$$P(z) = p_{0,0} + \sum_{j=0}^{T-1} P_j(z). \tag{16}$$

We have the conservation constraint:

$$P(1) = p_{0,0} + \sum_{j=0}^{T-1} P_j(1) = p_{0,0} + \sum_{j=0}^{T-1}\sum_{n=1}^{\infty} p_{n,j} = 1. \tag{17}$$

Hence, the average buffer occupancy (average number of blocks in the buffer) is the derivative of $P(z)$ at $z = 1$ [16], [17]. We proceed to derive an expression for $P(z)$ based on the following results given by (18) and (21) (see Appendix A for the derivations):

$$P_0(z) = \frac{z(z-1)\lambda p_{0,0}}{z - g(z)}, \tag{18}$$

where $g(z)$ is defined as

$$g(z) \equiv \sum_{j=1}^{T-1} c_j f(z)^j + (1 - F_{T-1}) f(z)^T, \tag{19}$$

and $f(z)$ is defined as

$$f(z) = 1 - \lambda + \lambda z. \tag{20}$$

Moreover,

$$P_j(z) = \prod_{i=1}^{j} (1-\mu_i) f(z)^j P_0(z) = (1 - F_j) f(z)^j P_0(z). \tag{21}$$

Substituting (18) and (21) into (16), we get an expression for the generating function as a function of $p_{0,0}$:

$$P(z) = p_{0,0} + \sum_{j=0}^{T-1} (1-F_j) f(z)^j P_0(z) = \left[1 + \sum_{j=0}^{T-1} (1-F_j) f(z)^j h(z)\right] p_{0,0}, \tag{22}$$

where $h(z)$ is defined as

$$h(z) = \frac{z(z-1)\lambda}{z - g(z)}. \tag{23}$$

Applying the conservation relation (17), we get

$$P(1) = \left[1 + \sum_{j=0}^{T-1} (1-F_j) f(1)^j h(1)\right] p_{0,0} = 1. \tag{24}$$

Hence,

$$p_{0,0} = \frac{1}{1 + \sum_{j=0}^{T-1} (1-F_j) f(1)^j h(1)}. \tag{25}$$

From (19), (20) and (23), we have

$$f(1) = 1 - \lambda + \lambda \cdot 1 = 1, \tag{26}$$

and

$$g(1) = \sum_{j=1}^{T-1} c_j f(1)^j + (1 - F_{T-1}) f(1)^T = 1. \tag{27}$$

Applying L'Hospital's rule, we have

$$h(1)=\lim_{z\to 1}\frac{z(z-1)\lambda}{z-g(z)}$$
$$=\lim_{z\to 1}\frac{(2z-1)\lambda}{1-\sum_{j=1}^{T-1}jc_j f(z)^{j-1}\lambda - T(1-F_{T-1})f(z)^{T-1}\lambda} \quad (28)$$
$$=\frac{\lambda}{1-\lambda\left[T(1-F_{T-1})+\sum_{j=1}^{T-1}jc_j\right]}.$$

Notice that the average decoding (service) time can be expressed as

$$\bar{c}=T(1-F_{T-1})+\sum_{j=1}^{T-1}jc_j. \quad (29)$$

For $T\geq 1$, we have $\bar{c}\geq 1$ since

$$\bar{c}=T(1-F_{T-1})+\sum_{j=1}^{T-1}jc_j \geq (1-F_{T-1})+\sum_{j=1}^{T-1}c_j=1. \quad (30)$$

And we have $\rho\leq 1$ since $T\geq 1$. Hence,

$$\rho=\lambda\bar{c}\geq\lambda. \quad (31)$$

Thus, (28) can be rewritten as

$$h(1)=\frac{\lambda}{1-\lambda\bar{c}}=\frac{\lambda}{1-\rho}, \quad (32)$$

where $\rho=\lambda\bar{c}$ is the utilization (or load of the queue system). As long as $\rho<1$, we have $h(1)>0$. Given the distribution of the decoding time in (4), we can determine the largest allowable incoming rate $\lambda_{\max}=1/\bar{c}$, which will decrease with either an increasing timeout limit T or a decreasing β (the exponent of the distribution).

Thus, (25) can be rewritten as

$$p_{0,0}=\frac{1}{1+\frac{\lambda}{1-\rho}\sum_{j=0}^{T-1}(1-F_j)}. \quad (33)$$

We then find the derivative of $P(z)$ from (22) as

$$P'(z)=p_{0,0}\sum_{j=0}^{T-1}(1-F_j)\left[jf(z)^{j-1}h(z)f'(z)+f(z)^j h'(z)\right]. \quad (34)$$

As shown in Appendix B,

$$h'(1)=\lambda\frac{1-\frac{2+\lambda}{2}\rho+\frac{\lambda^2\bar{d}}{2}}{(1-\rho)^2}, \quad (35)$$

where $\bar{d}$ is the mean square of the decoding time as given by

$$\bar{d}=T^2(1-F_{T-1})+\sum_{j=1}^{T-1}j^2c_j. \quad (36)$$

Substituting (33) into (34), we thus obtain the following expression for the average number of blocks in the buffer.

$$\frac{dP(z)}{dz}\Big|_{z=1}=p_{0,0}\sum_{j=0}^{T-1}(1-F_j)\left[jh(1)+h'(1)\right]$$
$$=\lambda\frac{uv+\lambda\sum_{j=1}^{T-1}j(1-F_j)}{1-\rho+\lambda v}, \quad (37)$$

where

$$u=\frac{1-\frac{2+\lambda}{2}\rho+\frac{\lambda^2\bar{d}}{2}}{1-\rho}, \quad (38)$$

and

$$v=\sum_{j=0}^{T-1}(1-F_j). \quad (39)$$

IV. Retransmission of Timeout Blocks

In the model described in section II, if the decoding of a block cannot be completed within the timeout limit T, then the block is considered as lost. In practice, such a lost block may be recovered by retransmission [14], [15]. The support for an ARQ can be incorporated into the discrete-time model in section II: If a block's decoding has to take longer than T slots, the decoder terminates that block's decoding after T slots and signals to the sender that the decoding fails. Consequently, the block is retransmitted at the succeeding slot. As such, any block arriving at the buffer will fall into one of the two possible categories: either a new block or a retransmitted block. When block retransmission is not allowed, new blocks arrive at the decoder according to the Bernoulli process, that is, a slot carries an arriving block with probability λ, and it is idle (no transmission) with probability $1-\lambda$. When an ARQ is used, we should also take into account those retransmitted blocks arriving at the buffer during the slots following decoding timeouts. The overall probability of the block arrival (incoming) rate will be changed to λ'. Because of the combining effect, we expect that $\lambda'\geq\lambda$, which will lead to a larger average buffer occupancy as opposed to the previous queue model without block retransmission.

In the following, we will reveal the relationship between the overall block arrival rate and the new block arrival rate. A queuing analysis similar to those in section III can then be conducted for

this modified queuing model, allowing retransmission of timeout blocks, by replacing the new block incoming rate (λ in section III) with the overall block incoming rate λ'.

From the perspective of the decoder, at any given time slot, either there will be no block being decoded, with probability $1-\lambda'$, or a block whose decoding will be finished within T slots, with probability of $\lambda' F_T$, where F_T is the distribution of the decoding time as defined in (2). Therefore, we have

$$\Pr\{\text{no retransmitted block}\} = 1-\lambda'+\lambda' F_T. \quad (40)$$

From the perspective of the buffer, at any given time slot, there will be a block arriving at the buffer unless the following two conditions are satisfied: i) there will be no new block arrival, with probability $1-\lambda$, and ii) there will be no retransmitted block arrival, with the probability given in (40). Hence, the overall block arrival probability λ' can also be expressed as

$$\begin{aligned}\lambda' &= 1 - Pr\{\text{no incoming block}\}\\ &= 1 - Pr\{\text{no new block}\} \times Pr\{\text{no retransmitted block}\}\\ &= 1-(1-\lambda)(1-\lambda'+\lambda' F_T). \end{aligned} \quad (41)$$

From (41), we can obtain the overall block arrival probability as

$$\lambda' = \frac{\lambda}{1-(1-\lambda)(1-F_T)}. \quad (42)$$

It can be easily shown that $\lambda' \geq \lambda$. From Fig. 2, we can see that as the probability of decoding timeout increases, that is, as F_T decreases, the overall rate λ' increases for a given new block arrival rate λ. This is expected since there will be more and more retransmitted blocks as F_T decreases. In the extreme case of $F_T = 0$, every new coming block (if any) will result in a retransmitted block, which will in turn result in another retransmitted block, and so on. Therefore, the overall incoming rate will reach 1. The other extreme is when $F_T = 1$, which means that each block will be decoded within T slots with probability 1. In this case, we have $\lambda' = \lambda$.

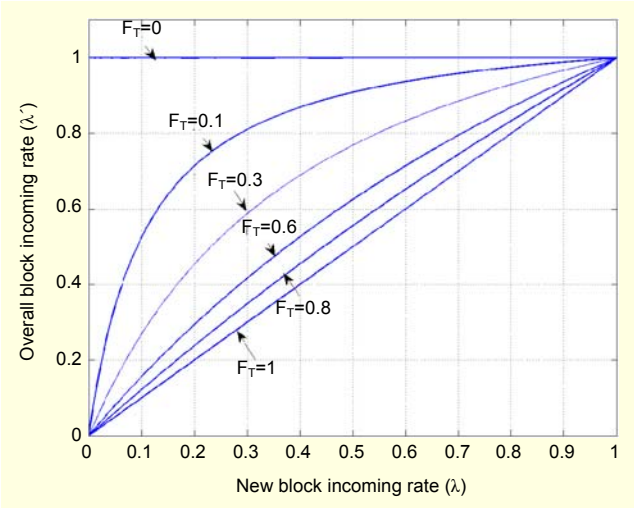

Fig. 2. The relation between the overall block incoming rate and new block incoming rate.

V. Results

In this section, we will evaluate the analytical results obtained in section III numerically. We will then present the simulation results to further validate the numerical results.

1. Numerical Results

Without loss of generality, we choose the duration of a single time slot as $T_r = 1$ in (6) and $\tau_0 = 1$ for the Pareto distribution given in (4). That is, the minimum time it takes the decoder to decode a block is one time slot. Then, for a given β, the average decoding time $\bar{c}$ and the mean square of the decoding time $\bar{d}$ can be calculated according to (29) and (36), respectively.

As shown earlier, λ (the probability of having an incoming block in a time slot, which can be viewed as the data block incoming rate) cannot be chosen arbitrarily. There exists a largest allowable value $\lambda_{\max} = 1/\bar{c}$ for a given T and β as shown in Fig. 3. If $\lambda > \lambda_{max}$, then the buffer occupancy will go to infinity. As can be seen in Fig. 3, if $\beta = 1.5$ and $T = 100$, then $\lambda_{max} < 0.3$. However, if $\beta = 2.5$ and $T = 100$, then $\lambda_{max} > 0.4$. This is because a larger β generally corresponds to a higher channel SNR, and thus a shorter decoding time on average. Consequently, the Fano decoder that decodes faster can handle a higher data income rate. On the other hand, for the same β,

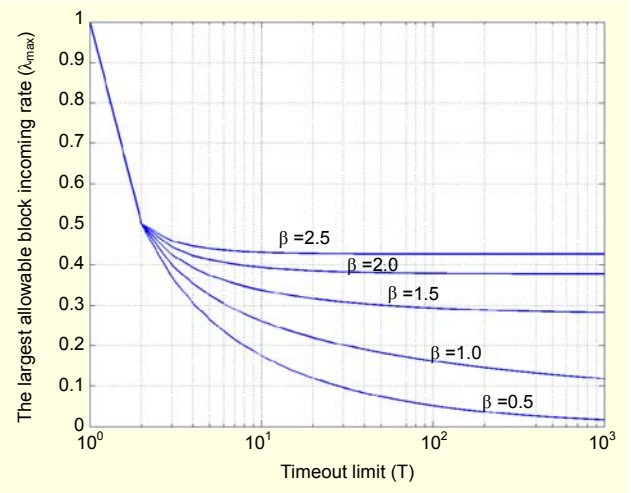

Fig. 3. The largest allowable incoming rate λ_{max} versus timeout limit T.

λ_{max} generally decreases with increasing T, which is to be expected. However, if the channel condition is sufficiently good (for example, $\lambda_{max} = 2.5$), then λ_{max} tends to be less insensitive to the change of T after T goes beyond a certain value ($T > 10$ in the case of $\beta = 2.5$). This is also expected since if the timeout limit is large enough, then most of the data blocks from the good channel can be completely decoded prior to timeout. Note that in Fig. 3, curves corresponding to different β values converge at $T = 2$. The reason for this is that the average decoding time $\bar{c}$ turns out to be the same regardless of the β value. This is also true for the special case of $T = 1$.

Numerical results for the average buffer occupancy are summarized in Fig. 4 for two β values. We can see that the average buffer occupancy (number of blocks) increases monotonically with increasing probability of arriving blocks. On the other hand, the average buffer occupancy decreases monotonically with decreasing limits of decoding time T. This is expected since a smaller T means that the decoder is given a tighter time budget to decode a block. Consequently, more blocks that require a long decoding time will be simply discarded to make the buffer less occupied. On the other hand, since a larger β corresponds to a higher channel SNR, and in turn to a smaller average decoding time, for fixed λ and T, the average buffer occupancy will decrease with an increasing β (with the case of $T = 1$ being the only exception, which is explained below). Note in Fig. 4(a), the curves for $T = 100$ and $T = 1000$ overlap almost completely with each other. This means that if the channel is sufficiently good ($\beta = 2.5$), then the buffer occupancy becomes insensitive to the change of timeout limits. The reason why the curves corresponding to $T = 1$ do not resemble the other curves for $T > 1$ is that when $T = 1$, the Markov model depicted in Fig. 1 degenerates into a simple two-state model, as shown in Fig. 5. A steady-state analysis of this model can readily yield the state probabilities as $p_{0,0} = 1 - \lambda$, and $p_{1,0} = \lambda$. Hence the average buffer occupancy is $0 \times p_{0,0} + 1 \times p_{1,0} = \lambda$. That is, for the case of $T = 1$, the average buffer occupancy grows linearly with an increasing λ (note the Log scale used for the buffer occupancy in Fig. 4) and is independent of the Pareto exponent β. This is also expected since $T = 1$ means that each block can only spend at most 1 time slot for decoding before the decoding reaches its timeout limit.

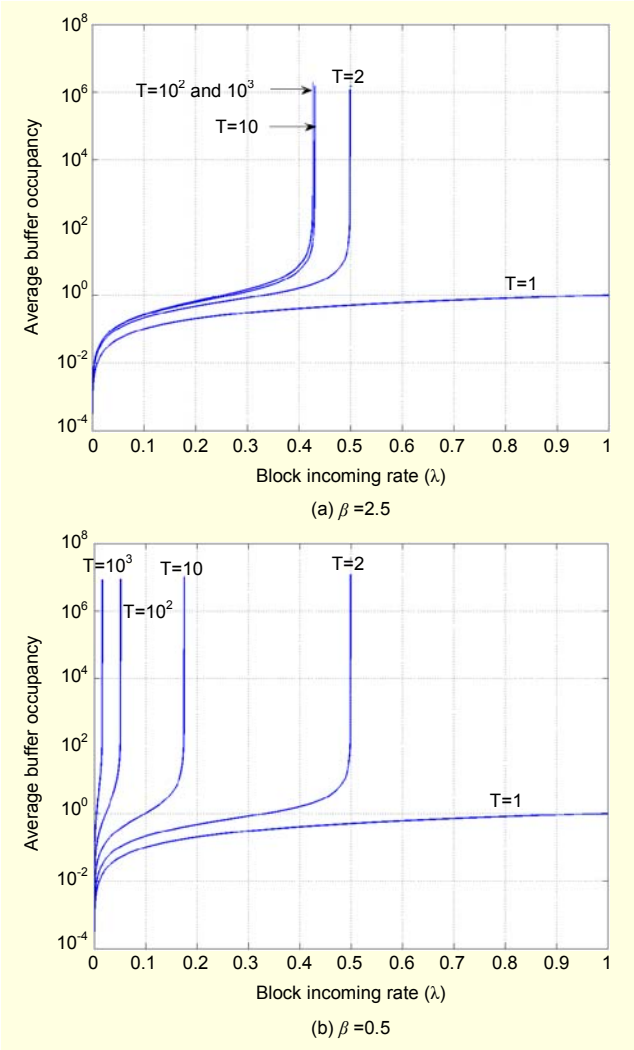

Fig. 4. Average buffer occupancy as a function of timeout limit T and β.

In fact, we can also obtain the same result by using (37) for $T = 1$. From (29) and (36), we have $\bar{c} = \bar{d} = 1$. Then from (31), we have $\rho = \lambda$. We thus have $u = 1$ and $v = 1$ from (38) and (39). Substituting into (37), we can therefore find the average buffer occupancy to be λ.

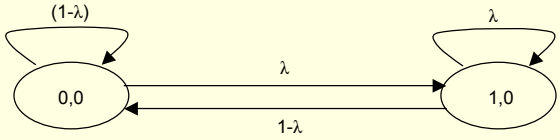

Fig. 5. The two-state Markov model.

2. Simulation Results

We simulated the Fano decoding system with timeout based on the discrete-time model described in section II. Random number generators were used to simulate the Bernoulli block arrival process as well as the Fano decoding time, which follows the Pareto distribution given in (4). During the simulation, the number of data blocks in the queue was recorded for each time slot. Then the average buffer occupancy over the duration of the simulation could be calculated. The average buffer occupancies found through simulations of 10^7 time slots for $\beta = 0.5$ and $\beta = 2.5$ are shown in Fig. 6. It can be seen that the simulation results agree fairly well with the numerical results in Fig. 4. However, the effect of finite duration of the simulations can also be

observed in Fig. 6. That is, when the block incoming rate approaches λ_{max} in Fig. 3, as prescribed by the theory, the theoretical average buffer occupancy will jump quickly to ∞ as shown in Fig. 4. By contrast, the average buffer occupancies obtained via simulation tend to move gradually towards "infinity", which is actually the largest possible buffer occupancy that can be achieved for the duration of the simulation. In the worst case, the decoder can take forever to decode a block, and there will be a block arrival for each time slot. Therefore, the largest possible buffer occupancy is the duration (total number of time slots) of the simulation (10^7 blocks in Fig. 6).

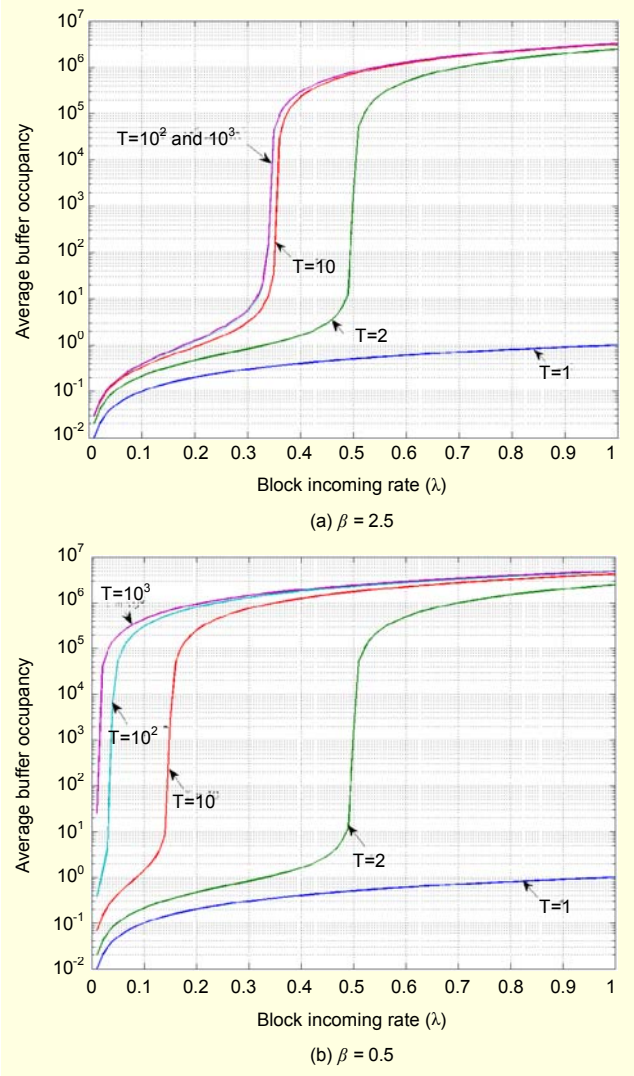

Fig. 6. Average buffer occupancy obtained by simulations.

VI. Concluding Remarks

In a practical Fano decoding system, not only are buffers required to absorb the variable processing delays of the sequential decoders, but also a limit will often be imposed in the decoder so that the decoding operation will be terminated if the decoding time reaches the limit. We have investigated the queuing behavior of a Fano decoding system with buffer by modeling the system with timeout limits using discrete-time semi-Markov models. The queuing analysis reveals the relations between the average buffer occupancy, the channel conditions, and decoding time limits. The largest possible incoming rate is also determined as a function of the Pareto exponent and decoding timeout limits.

In a practical decoding system, there is the issue of buffer capacity planning, which typically involves trade-offs between the assignment of buffer size and the probability of buffer overflow due to finite buffer sizes. A conservative approach can be taken by assuming that the timeout limit is sufficiently large (for example, T = 1000). From Fig. 4, the buffer sizes of 100 and 10,000 can be chosen under varying β values (that is, varying channel conditions). It can be inferred from Fig. 3 that even larger buffers will offer little help in lowering the buffer overflow probability as the data arrival rate approaches the maximum allowable limits.

We should point out that the expression in (37) is a fairly generic result, as it does not require the decoding time to follow necessarily the Pareto distributions. In fact, given any probabilistic distribution (either theoretical or empirical) of the decoding time, we can determine the average buffer occupancy by utilizing (37). Therefore, theoretical results obtained in this paper are applicable to other channel decoders with the characteristics of variable decoding time. For example, both the Turbo decoder [18] and the low-density parity check decoders [19] exhibit variable decoding latencies due to multiple decoding iterations, where the number of iterations actually chosen can be viewed as a controllable timeout limit.

Appendix A. Derivations of $P_0(z)$ and $P_j(z)$

From (15), we have

$$P_0(z) = \sum_{n=1}^{\infty} p_{n,0} z^n = p_{1,0} z + \sum_{n=2}^{\infty} p_{n,0} z^n .$$

From (11) and (13), we have

$$\begin{aligned} P_0(z) &= \sum_{j=1}^{T-1} \mu_j \left[(1-\lambda) \sum_{n=1}^{\infty} p_{n+1,j-1} z^n + \lambda \sum_{n=1}^{\infty} p_{n,j-1} z^n \right] \\ &\quad + \lambda \sum_{n=1}^{\infty} p_{n,T-1} z^n + (1-\lambda) \sum_{n=1}^{\infty} p_{n+1,T-1} z^n + \lambda z p_{0,0} \\ &= \frac{1}{z} \sum_{j=1}^{T-1} \mu_j f(z) P_{j-1}(z) + \frac{f(z)}{z} P_{T-1}(z) + \lambda z p_{0,0} \\ &\quad - \sum_{j=1}^{T-1} \mu_j (1-\lambda) p_{1,j-1} - (1-\lambda) p_{1,T-1}, \end{aligned}$$

where we define

$$f(z) \equiv 1-\lambda+\lambda z. \tag{43}$$

Using (9) for $\lambda p_{0,0}$, we obtain the expression for $P_0(z)$, which is related to $P_j(z)$ for $j \geq 1$ as

$$P_0(z)=\frac{f(z)}{z}\sum_{j=1}^{T-1}\mu_j f(z)P_{j-1}(z)+\frac{f(z)}{z}P_{T-1}(z)+\lambda(z-1)p_{0,0}. \tag{44}$$

Next, we derive the expression for $P_j(z)$ for $j \geq 1$. From (12) and (14),

$$\begin{aligned} P_1(z) &= \sum_{n=1}^{\infty} p_{n,1}z^n = p_{1,1}z + \sum_{n=2}^{\infty} p_{n,1}z^n \\ &= (1-\lambda)(1-\mu_1)p_{1,0}z + \sum_{n=2}^{\infty}(1-\mu_1)\left[(1-\lambda)p_{n,0}+\lambda p_{n-1,0}\right]z^n \\ &= (1-\lambda)(1-\mu_1)P_0(z) + \lambda(1-\mu_1)zP_0(z) \\ &= (1-\mu_1)f(z)P_0(z). \end{aligned} \tag{45}$$

Generally, for $j \geq 1$ we have

$$\begin{aligned} P_j(z) &= p_{1,j}z + \sum_{n=2}^{\infty} p_{n,j}z^n \\ &= (1-\lambda)(1-\mu_j)p_{1,j-1}z \\ &\quad + \sum_{n=2}^{\infty}(1-\mu_j)\left[(1-\lambda)p_{n,j-1}+\lambda p_{n-1,j-1}\right]z^n \\ &= (1-\mu_j)f(z)P_{j-1}(z). \end{aligned} \tag{46}$$

From (45) and (46), we get

$$P_j(z) = \prod_{i=1}^{j}(1-\mu_j)f(z)^j P_0(z) = (1-F_j)f(z)^j P_0(z). \tag{47}$$

Substituting (47) into (44) yields

$$\begin{aligned} P_0(z) &= \frac{f(z)}{z}\sum_{j=1}^{T-1}\mu_j(1-F_{j-1})f(z)^{j-1}P_0(z) \\ &\quad + \frac{f(z)}{z}(1-F_{T-1})f(z)^{T-1}P_0(z) + \lambda(z-1)p_{0,0} \\ &= \frac{P_0(z)f(z)}{z} + \lambda(z-1)p_{0,0}, \end{aligned} \tag{48}$$

where $g(z)$ is defined as

$$g(z) \equiv \sum_{j=1}^{T-1} c_j f(z)^j + (1-F_{T-1})f(z)^T. \tag{49}$$

From (48), we have

$$\left[1-\frac{g(z)}{z}\right]P_0(z) = \lambda(z-1)p_{0,0}. \tag{50}$$

Hence,

$$P_0(z) = \frac{z(z-1)\lambda p_{0,0}}{z-g(z)}. \tag{51}$$

Appendix B. Derivations of (35)

In the following, we derive the expression for $h'(1)$, which is used in (34) to determine the average buffer occupancy.

From (23), by applying L'Hospital's rule twice we obtain

$$\begin{aligned} h'(1) &\equiv \frac{dh(z)}{dz}\Big|_{z=1} \\ &= \lim_{z\to 1}\frac{-\left[1-g'(1)\right]z(z-1)+\left[z-g(z)\right](2z-1)}{\left[z-g(z)\right]^2} \\ &= \lambda\frac{g''(1)+2\left[1-g'(1)\right]}{2\left[1-g'(1)\right]^2}. \end{aligned} \tag{52}$$

From (19), we can find $g'(z)|_{z=1}$ and $g''(z)|_{z=1}$ by using the fact that $f'(z)|_{z=1}=\lambda$, and $f''(z)=0$, $\forall z$,

$$g'(z) = (1-F_{T-1})Tf(z)^{T-1}f'(z) + \sum_{j=1}^{T-1} jc_j f(z)^{j-1}f'(z). \tag{53}$$

Hence,

$$g'(1) \equiv g'(z)|_{z=1} = \lambda\left[T(1-F_{T-1}) + \sum_{j=1}^{T-1} jc_j\right] = \lambda\bar{c} = \rho, \tag{54}$$

where $\bar{c}$ is the average decoding time as given in (29). Since we require $\rho < 1$, therefore

$$1-g'(1) = 1-\rho > 0. \tag{55}$$

Furthermore,

$$\begin{aligned} g''(z) &= (1-F_{T-1})T(T-1)f(z)^{T-2}\left[f'(z)\right]^2 \\ &\quad + \sum_{j=1}^{T-1} j(j-1)c_j f(z)^{j-2}\left[f'(z)\right]^2, \end{aligned}$$

and hence

$$\begin{aligned} g''(1) &\equiv g''(z)|_{z=1} \\ &= \lambda^2\left[(1-F_{T-1})T(T-1) + \sum_{j=1}^{T-1} j(j-1)c_j\right] = \lambda^2\left(\bar{d}-\bar{c}\right), \end{aligned} \tag{56}$$

where $\bar{d}$ is the mean square of the decoding time as given by

$$\bar{d} = T^2(1-F_{T-1}) + \sum_{j=1}^{T-1} j^2 c_j. \tag{57}$$

It can be easily shown that $\bar{d} \geq \bar{c}$ since $T \geq 1$. Hence, we have $g''(1) \geq 0$, and thus $h'(1) \geq 0$ by using (55). Therefore, from (52), (54) and (56), we have

$$h'(1) = \lambda \frac{1 - \frac{2+\lambda}{2}\rho + \frac{\lambda^2 \bar{d}}{2}}{(1-\rho)^2}. \quad (58)$$

References

[1] S. Lin and D.J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd Edition, Pearson Education, 2004.

[2] R.M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Trans. Information Theory*, vol. IT-9, Apr. 1963, pp. 64-74.

[3] R.G. Gallager, *Information Theory and Reliable Communication,* Wiley, New York, 1968.

[4] J.M. Wozencraft and I.M. Jacobs, *Principles of Communication Engineering*, Wiley, 1965.

[5] R.O. Ozdag and P.A. Beerel, "A Channel Based Asynchronous Low Power High Performance Standard-Cell Based Sequential Decoder Implemented with QDI Templates," *Proc. of 10th International Symposium on Asynchronous Circuits and Systems,* Apr. 2004, pp.187-197.

[6] S. Singh, P. Thienniviboon, R.O. Ozdag, S. Tugsinavisute, R. Chokkalingam, P.A. Beerel, and K.M. Chugg, "Algorithm and Circuit Co-Design for a Low-Power Sequential Decoder," *Proc. of Asilomar Conf. on Signal, Systems and Comp.*, Oct. 1999.

[7] T. Hashimoto, "Bounds on a Probability for the Heavy Tailed Distribution and the Probability of Deficient Decoding in Sequential Decoding," *IEEE Trans. on Information Theory*, vol. 51, no. 3, Mar. 2005, pp. 990-1002.

[8] I.M. Jacobs and E.R. Berlekamp, "A Lower Bound to the Distribution of Computation for Sequential Decoding," *IEEE Transactions on Information Theory*, vol. 13, no. 2, 1967, pp. 167-174.

[9] F. Jelinek, "An Upper Bound on Moments of Sequential Decoding Effort," *IEEE Transactions on Information Theory*, vol. 15, no. 1, 1969, pp. 140-149.

[10] R. Sundaresan and S. Verdu, "Sequential Decoding for the Exponential Server Timing Channel," *IEEE Trans. on Communications*, vol. 46, no. 2, Mar. 2000, pp. 705-709.

[11] F. Pollara, "A Software Simulation Study of a Sequential Decoder using the Fano Algorithm," *TDA Progress Report 42-81*, JPL, NASA, Jan.-Mar. 1985.

[12] O.R. Jensen and E. Paaske, "Forced Sequence Sequential Decoding: a Concatenated Coding System with Iterated Sequential Inner Decoding," *IEEE Transactions on Communications*, vol. 46, no. 10, 1998, pp. 1280-1291.

[13] W. Pan and A. Ortega, "Buffer Control for Variable Complexity Fano Decoders," *IEEE Global Telecommunications Conference*, San Antonio, Texas, Nov. 2001, pp. 176-180.

[14] S. Kallel and D. Haccoun, "Sequential Decoding with an Efficient Partial Retransmission ARQ Strategy," *IEEE Trans. on Communications*, vol. 39, no. 2, Feb. 1991, pp. 208-213.

[15] N. Shacham, "ARQ with Sequential Decoding of Packetized Data: Queuing Analysis," *IEEE Trans. on Communications*, vol. 32, no. 10, Oct. 1984, pp. 1118-1127.

[16] L. Kleinrock, *Queuing Systems Volume 1: Theory,* Wiley, 1975.

[17] D. Gross and C.M. Harris, *Fundamentals of Queuing Theory*, Wiley, 1998.

[18] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo–Codes," *Proc. IEEE International Conference on Communications*, May 1993, pp. 1064– 1070.

[19] Flarion Technologies Inc., "Vector - LDPC Codes for Mobile Broadband Communications," Nov. 2003. http://www.flarion.com/ products/ whitepapers/Vector-LDPC.pdf.

W. David Pan is an Assistant Professor in the Department of Electrical and Computer Engineering, University of Alabama in Huntsville, Huntsville, Alabama, USA. He received the PhD degree in electrical engineering from the University of Southern California in 2002, and the MS degree in computer engineering from the University of Louisiana at Lafayette in 1998. His research interests include image and video coding, communication, and multimedia information assurance.

Seong-Moo Yoo received the BS degree in economics from Seoul National University, Seoul, Korea, and the MS and PhD degrees in computer science from the University of Texas at Arlington in 1989 and 1995. Since September 2001, he has been an Associate Professor in the Electrical and Computer Engineering Department of the University of Alabama in Huntsville, Huntsville, Alabama, USA. He was the conference chair of ACM Southeast Conference 2004 in Huntsville. He was also the co-program chair of ISCA 16th International Conference on Parallel and Distributed Computing Systems (PDCS-2003), August 2003, Reno, Nevada, USA. His research interests include wireless networks, parallel computer architecture, and computer network security. He is a Senior Member of IEEE and a member of ACM.