

# Effect of Channel Conditions on Inventory Database Update in Supply Chains

Murat Kocaoglu\*, Cuneyt Oksuz†, Ozgur B. Akan\*

\* Next-generation and Wireless Communications Laboratory  
Department of Electrical and Electronics Engineering, Koc University, Istanbul, Turkey  
Email: {mkocaoglu, akan}@ku.edu.tr

† Idea Teknoloji Cozumleri Bil. San. ve Tic. Ltd. Sti.,  
Sun Plaza BBDO Blok, Dereboyu Cd., Bilim Sk, No:5, 34398, Maslak, Istanbul, Turkey  
Email: {cuneyt.oksuz}@ideateknoloji.com.tr

**Abstract**—The supply chain optimization is becoming more and more important, and more challenging due to its increased complexity. Companies should properly exploit ICT systems to better keep track of their products. Communication techniques employed in the supply chain should be adaptive to any channel condition to assure continuous reliable operation. In this paper, we investigate the effects of channel conditions and the selected error control technique on the supply chain inventory database update in terms of database error rate and end-to-end delay. We derive the analytical expressions and verify them with extensive simulations and show that channel codes adapting to channel conditions would improve reliability of the inventory database.

## I. INTRODUCTION

The movement of a product from production to the customer is modeled inside the *supply chain*. Supply chain covers a wide range of processes in the lifecycle of a product from the supply of the raw material, to the movement of the product to the distributors and end users. For corporations, supply chain management is crucial. Optimization of the global supply chain is required to maximize the profit. An important process in the supply chain is the accurate update of the inventory database. Consider an inventory record, which is updated as goods are transferred from manufacturers to the distributors. Inaccurate update of the inventory database could lead to stockout and should be prevented to maximize the profit [1]. In general controlling whether the database updates contain error or not is costly for large databases. Even if an error is detected, it is hard to spot and correct it [2]. Thus, it is important that the transactions for database update are kept error-free.

Today, wireless communications is an integral part of the supply chain. Information regarding the distribution of a product is transmitted from the distributors to the manufacturers via wireless and wired links. The accuracy of the stock database is crucial as even a small change's effect on the profit is tremendous in the long run. Since checking the database for the accuracy of the entries is exhausting, especially for very large data sets, errors should be prevented before occurring,

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) TEYDEB Program under Contract TEYDEB-7120977- JMobilVeri.

during the data transmission through the wireless communication channel, which constitutes the bottleneck of the communication link between the distributors and the manufacturers. As channel between the hand terminal and wireless access point that is used for the transmission of database update information is not always reliable, it is important to analyze the effects of the channel state and employed techniques to keep database error-free. To use the channel reliably, communication techniques that adapt themselves to the channel is necessary.

In this paper, we analyze the effect of channel variations on the accuracy of the inventory database update. We show that employing adaptive error correction techniques could improve average end-to-end delay and database error rate. We consider a fading wireless channel with burst errors. Results show that depending on the reliability of the channel, we can adapt the employed error control coding scheme to reduce the delay, or erroneous update rate of the database.

The paper is organized as follows: In Sec. II, we present the system model including the wireless communication link. Later, in Sec. III, we introduce the underlying end-to-end error control mechanism and evaluate its performance for varying parameters. In Sec. IV, we present the analytical evaluations of delay and packet error rate (PER). Lastly, in Sec. V, we present our simulations and compare them with analytical results.

## II. SYSTEM MODEL

Consider the scenario where a certain product of is sold to the distributor from the manufacturer. A database at the manufacturer contains the remaining quantity of a number of products in the stock. We want to update the manufacturer database each time a batch of products are sold to the distributor. A hand terminal, which scans the sold product, generates the information packets containing the type of the product and quantity of the sold units. The end-to-end communication link between the hand terminal and the inventory database is shown in Fig. 1. Hand terminals mostly use wireless communication to connect to a wireless access point. Since wired link is assumed to be reliable we only consider the errors and delays due to the wireless communication link.

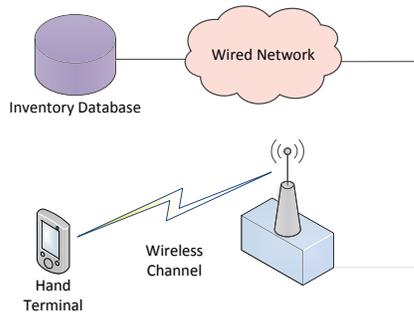


Fig. 1. System model.

Each transaction is represented with a binary  $k$ -tuple, i.e.,  $k$ -bits, which are called *the sourcewords*. A sourceword contains the product code and product quantity as shown in Fig. 2 and is generated at the hand terminal. Assume that the products are sold in base unit of 100. Then, by allocating 4 bits to the product amount, we can encode the sell of 100, 200, ...1600 number of products. A sample transaction is provided in the figure together with the corresponding database update. When the wireless communication channel is not reliable, there is the possibility that database will be updated erroneously. We show that errors depend on the channel state and employing adaptive communication techniques helps reduce the inventory error rate.

The main challenge of using wireless medium is to deal with the time varying characteristics. In the literature, Gilbert-Elliot model is widely used to characterize the fading channel with burst errors [3]. In this model, channel state changes according to a two-state Markov chain as shown in Fig. 3. Here  $p, q, h_g$  and  $h_b$  represent the channel transition probability from *bad* to *good* state, channel transition probability from *good* to *bad* state, bit error probability in the good, and bit error probability in the bad states, respectively. In general in fading channels,  $p > q$ . For example, it is calculated that at pedestrian speeds of 20 km/h,  $p \approx 10^{-3}$ , whereas  $q \approx 10^{-5}$  [3]. This assures that the channel is mostly in the good state where it occasionally drops to the deep fading state in which error probability is higher. In [4], authors derive an analytical expression using the simplified GE model to ease the analysis, in which they assume that the bit error probability is 0 in the *good* state, and bit error probability is 1 in the *bad* state. However, from an information theory perspective, bit error probability can at most be 0.5, as increasing it further actually increases the mutual information. This is trivial from the coin toss example. If we know that the channel state is in the deep fade, we can discard the result and toss a fair coin and still decode correct 50% of the time. Hence, we assume that bit error probability in the bad state is 0.5 and vary bit error probability in the good state in the wide range of from  $10^{-6}$  to  $10^{-1}$ .

### III. ERROR CONTROL AND CORRECTION

Error control and correction is a major research topic in the communications theory. Engineers have developed numerous

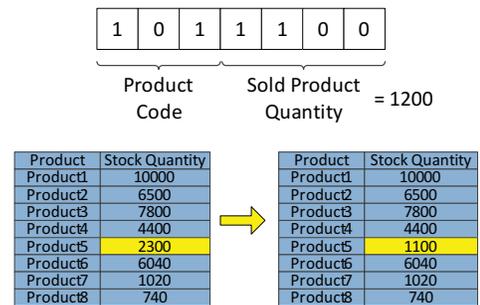


Fig. 2. Sourceword structure and product database update.

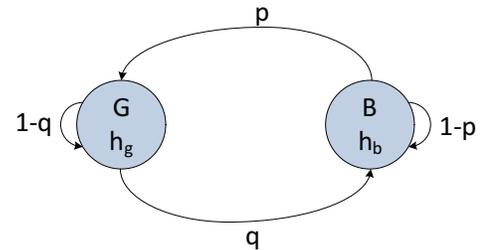


Fig. 3. Two-state Gilbert Elliot channel model.

techniques to detect and correct errors since Shannon. Error correction is mainly divided into two categories as *forward error correction*, i.e., FEC, and *automatic repeat request*, i.e., ARQ. FEC techniques exploit the idea of adding redundancy to the information to combat the effects of noise. In ARQ technique, packets are retransmitted depending on the packet acknowledgements received using the feedback channel to improve reliability. In modern communication systems, both of them are used to further decrease the packet error probability.

In this work, we consider several channel codes together with a retransmission strategy. We obtain the variation of end-to-end delay and packet error probability at the inventory database with varying channel conditions. Let us first discuss the channel codes that we employ in the simulations section.

For forward error correction, we only consider *linear block codes*. Linear block codes are important in coding theory due to the simplicity of encoding and decoding. A block code  $\mathcal{C}$  contains a number of  $n$ -tuples called the codewords, which are used to encode the source information. Hence a code determines the set of channel codewords and the mapping between the source and channel codewords. Hamming distance between two codewords is defined as the number of bits at which they differ. Hamming distance of a code, or simply code distance, is the minimum Hamming distance among all the codeword pairs in the codebook. Code distance determines the error correction capability of a code. Employing minimum distance decoding, a channel code with Hamming distance of  $d$  can correct up to  $\lfloor \frac{d-1}{2} \rfloor$ .

Linear nature of these codes allow simply mapping the sourcewords to the codewords with a generator matrix  $\mathbf{G}$ . Moreover, a parity check matrix  $\mathbf{H}$  can be used to detect

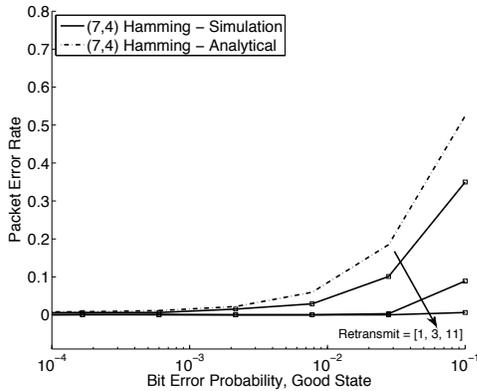


Fig. 4. Comparison of analytical and simulation results for PER vs.  $h_G$  for (7,4) Hamming.

if there is an error in the received codeword or not. In the simulations, we will exploit the error detection capability of the code rather than its error correction capability.

An  $[n, k, d]$  block code  $\mathcal{C}$  contains length- $n$  codewords each representing one of the  $M = 2^k$  source symbols, and separated from each other by at least Hamming distance of  $d$ . The codeword representing the  $i^{\text{th}}$  sourceword is  $c_i$ , and it has *weight*  $w_i$ , i.e., it contains  $w_i$  number of ones. Let  $u$  be the number of products sold at the distributor, and  $\mathbf{s}$  be the corresponding binary sourceword, i.e.,  $\mathbf{s} \in \mathbb{F}_{2^k}$ . The  $k \times n$  generator matrix  $\mathbf{G}$  of the linear block code maps the sourcewords to the codewords with the relation  $\mathbf{x} = \mathbf{s}\mathbf{G}$ . Hence, any codeword  $\mathbf{c} \in \mathcal{C}$  is in the row space of  $\mathbf{G}$ . If  $\mathbf{G}$  is structured such that  $\mathbf{G} = [\mathbf{I}_k | \mathbf{P}]$ , where  $\mathbf{I}_k$  is  $k \times k$  identity matrix,  $\mathbf{G}$  is called *systematic*. The parity check matrix  $\mathbf{H}$  is the  $n - k \times n$  matrix such that any codeword  $\mathbf{c}$  is in the null space of  $\mathbf{H}$ , i.e.,  $\mathbf{H}\mathbf{x}^T = \mathbf{0}$ . Hence the parity check matrix  $\mathbf{H}$  can be easily used to detect channel errors. Let  $\mathbf{e}$  be an error vector due to the channel errors such that the  $n$ -tuple at the receiver is  $\mathbf{y} = \mathbf{x} + \mathbf{e}$ , where addition is in modulo 2. Then parity check matrix is non-zero, since  $\mathbf{H}(\mathbf{x} + \mathbf{e}) = \mathbf{H}\mathbf{e} \neq \mathbf{0}$ . It is important to note that if error vector is the same as one of the codewords, parity check operation fails. We will see in the next section that this is the only case error detection fails.

In this work, we evaluate the performance of different Hamming and cyclic codes to evaluate the packet error rate and delay performance of the database update. Hamming codes are determined by a single parameter  $m$ , which yields  $k = 2^m - m - 1$ ,  $n = 2^m - 1$  and  $d = 3$ . Details on these and the cyclic codes are available in [5] and is skipped here. Authors in [5] provide available  $(n, k)$  pairs for cyclic codes, which we will use in the performance evaluation. After error detection, retransmission is requested from the transmitter. The maximum number of transmissions is shown by  $\nu$ .

#### IV. PERFORMANCE ANALYSIS

We model the time varying wireless link with Gilbert-Elliot (GE) model, which is a simplified two-state Markov chain representation of the channel containing the *good* and *bad*

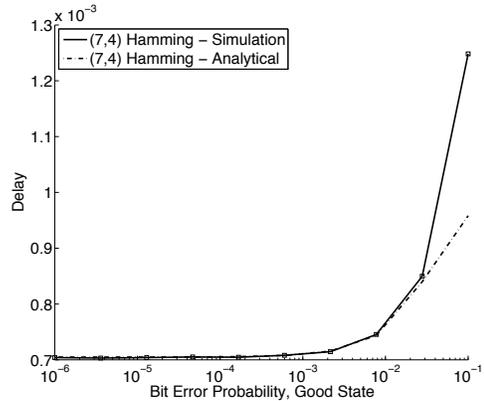


Fig. 5. Comparison of analytical and simulation results for Delay vs.  $h_G$  for (7,4) Hamming, where  $\nu=3$ .

states, as stated previously. Considering Fig. 3,  $p$  and  $q$  are the state transition probabilities from *bad* to *good* and *good* to *bad* states, respectively. We know that the steady state probabilities for the *good* and *bad* states are given as [6]

$$\pi_G = \frac{p}{p+q}, \text{ and } \pi_B = \frac{q}{p+q}. \quad (1)$$

Then the average bit error probability is obtained by

$$h = h_G \pi_G + h_B \pi_B. \quad (2)$$

Since we use channel coding purely for error detection, we do not employ minimum distance decoding at the receiver. We define 3 different probabilities to use in our calculations:

$p_c$  : Prob. that packet received free of error.

$p_{e|e}$  : Prob. that packet received with a detectable error.

$p_{c|e}$  : Prob. that packet received with an undetectable error.

We can obtain these values as

$$\begin{aligned} p_c &= \text{Prob}\{\mathbf{e} = \mathbf{c}_0 = \mathbf{0}\} = (1-h)^n, \\ p_{c|e} &= \text{Prob}\{\mathbf{e} = \mathbf{c}_i \in \mathcal{C} : \exists i \in \{1, \dots, 2^k - 1\}\} \\ p_{e|e} &= 1 - p_{c|e} - p_c, \end{aligned} \quad (3)$$

by assuming that the all-zero codeword is shown by  $\mathbf{c}_0$ .  $p_{c|e}$  in (3) is obtained using the fact that the code is linear and for  $\mathbf{y} = \mathbf{x} + \mathbf{e}$  to be a codeword,  $\mathbf{e}$  should also be a codeword.

Note that retransmission occurs only if an error is detected. Hence, we obtain the retransmission probability of a packet as

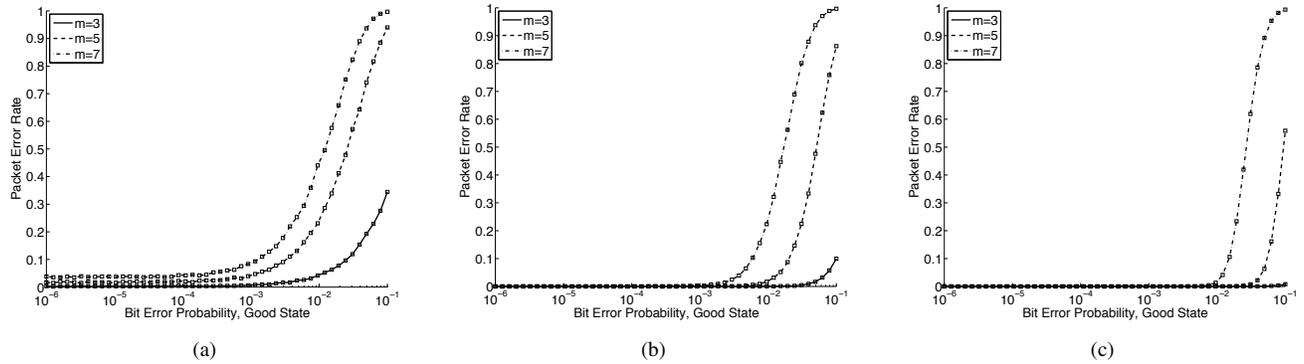
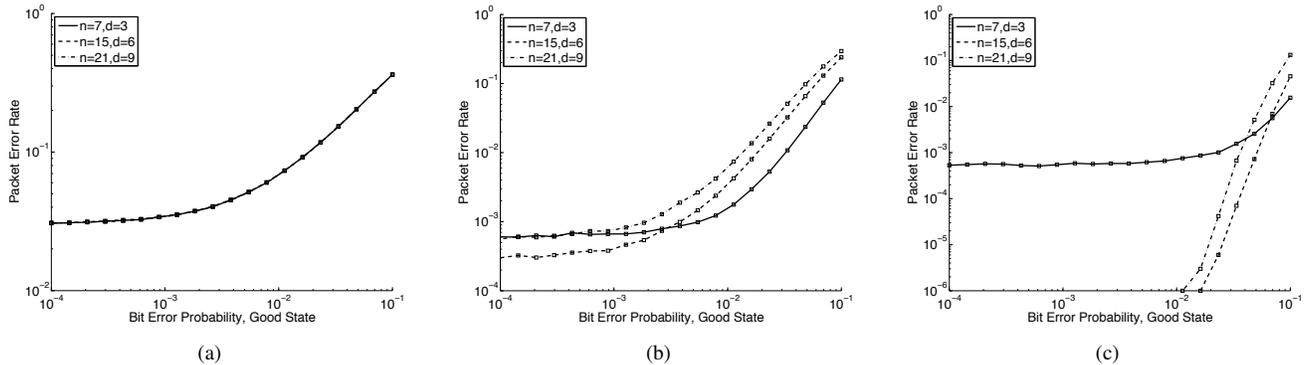
$$p_{Re} = p_{e|e} = 1 - p_{c|e} - p_c. \quad (4)$$

To evaluate (3), we need the number of ones and zeros in each codeword, i.e., its weight. Hence,

$$p_{Re} = 1 - \sum_{i=0}^{M-1} h^{w_i} (1-h)^{n-w_i}, \quad (5)$$

since  $\mathbf{e} = \mathbf{0}$  does not yield any error. Finally, similar to the analysis in [4], the average number of transmissions is

$$E[N] = \sum_{i=0}^{\nu-1} (i+1)(1-p_{Re})(p_{Re})^i. \quad (6)$$


 Fig. 6. PER vs.  $h_G$  for different Hamming codes, for MaxRetransmit=1,3 and 11 respectively.

 Fig. 7. PER vs.  $h_G$  for different cyclic codes, for MaxRetransmit=1,3 and 11 respectively. Increased code distance improves performance.

We know that each packet retransmission takes  $\frac{L}{B}$  seconds, where  $L$  is the packet length and  $B$  is the bandwidth in Hz. Then the average delay due to retransmission is

$$E[D] = \frac{L}{B} \sum_{i=0}^{\nu-1} (i+1)(1 - P_{Re})(P_{Re})^i. \quad (7)$$

For PER, we know that a packet is received correctly with probability  $p_c$ . If there were no retransmissions, it is easy to say that packet error rate  $PER = p_c$ . We employ this approximation in the next section, and compare the analytical results with the simulations to quantify the effect of retransmissions.

When a packet is received, the inventory database is updated. Let  $M_t$ ,  $u_t$ , and  $\hat{u}_t$  be the quantity of a certain product in the database at time  $t$ , actual and estimated quantities of the sold product. Accordingly, we have,

$$M_t = M_{t-1} - \hat{u}_{t-1} = M_0 - \sum_{i=0}^{t-1} \hat{u}_i. \quad (8)$$

Assume that when there is an error in the packet, the probability that the packet is switched to any other packet is the same for all the packets, i.e., packet error is uniformly distributed. This is actually not the case since all the error vectors are not equally likely, however it greatly simplifies the analysis. Correspondingly, the standard deviation from the

actual inventory count is

$$\sqrt{E[(M_t - M_t^{act})^2]} = \sqrt{E[(u - \hat{u})^2]}, \quad (9)$$

and with the uniform error and uniform input assumption, we obtain,

$$\begin{aligned} E[(u - \hat{u})^2] &= \frac{1}{M} \sum_{i=0}^{M-1} \left( \sum_{j=0}^{M-1-i} j^2 + \sum_{k=0}^i k^2 \right), \\ &= \frac{1}{M} \sum_{i=0}^{M-1} \left( \frac{(M-1-i)(M-i)(2M-i-1)}{6} \right. \\ &\quad \left. + \frac{(i)(i+1)(2i+1)}{6} \right) \end{aligned} \quad (10)$$

since the source symbols are in the set  $\{0, 1, \dots, M-1\}$ . The derivation of this expression is easy and skipped here. In the next section, we compare our analytical expressions with the simulation results.

## V. SIMULATION RESULTS

In this section, we evaluate the delay and database error rate, i.e., packet error rate for varying channel and coding parameters. Different Hamming and cyclic codes are evaluated with varying  $n$ ,  $k$  and  $d$ . We evaluate the system performance in MATLAB via both simulations and the evaluation of analytical expressions.

In our simulations, we generate channel state transition vector for each packet according to the GE model given in Fig. 3. After the generation of the channel state vector, a random source input is selected uniformly from the set  $\{0, 1, \dots, M - 1\}$ , and converted into binary sourcewords. Each sourceword is mapped to the channel codewords via a generator matrix  $\mathbf{G}$ , which is generated via MATLAB for Hamming and cyclic codes. During transmission, each bit is switched according to the bit error probability of the current channel state. The received  $n$ -tuple is then multiplied with the parity check matrix to see if there is an error, in which case the same symbol is retransmitted. If the source is still not correctly decoded after  $\nu$  trials, transmitter skips the transmission and corresponding symbol is received in error. After a large number of source transmissions (0.1-1 M), we compare the source database with the destination database, and obtain the average PER via simulations as  $PER = \frac{\text{Packets in Error}}{\text{Total Packets in Database}}$ .

Fig. 4 shows both the simulation and analytical results for the packet error rate for (7,4) Hamming, which is used as its weight distribution is available and easy to implement. Analytical result is obtained in Sec. IV. Each database contains  $N = 1000$  entries, each from a set with cardinality of  $2^M$ , where  $p = 0.5, q = 0.001$  and  $h_B = 0.5$ . As we have explained in the previous section, the difference between the analytical and the theoretical error probabilities are due to retransmissions. Hence, when channel conditions become unfavorable, maximum number of retransmissions should be increased to decrease the error rate in the manufacturer database. The theoretical and analytical delay performance is provided in Fig. 5. The results overlap most of the time.

We observe the effect of increasing the code length with Hamming, and increasing source cardinality with cyclic code.

#### A. Increased Sourceword Length, Fixed Distance

In this section, we compare the performance of (7,4), (31,26) and (127,120) Hamming codes. As observed in Fig. 6, we see that, even though increasing the maximum retransmission helps in terms of error probability, increasing the code source cardinality does not. This is expected since by increasing the sourceword length, packet length increases, which increase the number of uncorrectable error patterns, since there is no accompanying code distance increase to compensate. Hence, the best strategy is to choose the code with minimum length that satisfies the minimum packet error requirement. Then, corresponding delay will also decrease, allowing faster update of the database. Additionally, increasing the maximum number of transmissions significantly improves the PER for Hamming code, especially for small  $m$  values.

#### B. Increased Code Distance, Fixed Sourceword Length

To understand the error correction with increasing code distance, we employ cyclic channel codes with fixed source set cardinality. We have used the  $(n, d) = (7, 3), (15, 6)$ , and  $(21, 9)$  cyclic codes. Each of these codes has source set cardinality of  $2^4 = 16$ . As observed, when maximum number of transmissions is 1, all the channel codes achieve the

TABLE I  
STANDARD DEVIATION OF STOCK QUANTITY ERRORS FOR VARYING  $n, k$ .

		$n$				
		7	9	15	21	27
$k$	2	-	1.6199	1.6241	1.6331	1.6249
	3	2.8436	2.8608	2.8847	2.8918	2.8599
	4	5.1073	-	5.1677	5.1761	-
	5	-	-	9.4620	9.4360	-
	6	-	-	17.3676	17.1902	17.5428
	7	-	-	32.5440	32.4697	32.0260
	8	-	-	60.9367	62.1382	61.2458
	9	-	-	114.6683	113.3815	114.6837

same PER due to the structure of the code. When maximum number of retransmissions is increased, different error patterns arise. We observe that some codebooks are affected from retransmission more than the others. Moreover, the best code depends on the state of the wireless channel for a given  $\nu$ . For example, considering Fig. 7(b), for  $\nu = 3$ , (7,4,3) cyclic code performs better than (15,4,6) cyclic code for  $h_G < 210^{-3}$ , whereas opposite is true for  $h_G > 210^{-3}$ . Similar results showing the dependence of the optimum codebook choice to the channel parameters are observed in Fig. 7(b).

Table I shows the variation of the standard deviation of the stock quantity error which, as observed, mostly depends on the number of source symbols. This result shows that quantifying the sold product with more resolution, i.e., selling in smaller base units could help reduce the deviation from the actual stock value. If we evaluate our analytical results regarding standard deviation in (10), we obtain results much higher than the simulations, i.e.,  $\sigma = 3.1, 9.1, 26, 73.8$  for  $k = 2, 3, 4$  and 5 respectively. This tells us that, the assumption that all the errors are equally likely is fallacious. Therefore, we can exploit this fact and adjust the error distribution by properly selecting the codewords, especially codeword weights, such that the deviation of error is small. This arises another optimization problem that could help to improve the accuracy of the inventory database by properly the codeword weights.

## VI. CONCLUSIONS

In this paper, we have investigated the supply chain database update error and delay performance for varying channel conditions and channel codes. We have derived the analytical expressions for the packet error rate and end-to-end delay and shown that they are similar with simulations. Results show that adaptive error control could improve the database reliability.

## REFERENCES

- [1] Y. Kang and S. Gershwin, "Information inaccuracy in inventory systems", *IIE Trans.*, vol. 37, no. 9, 2005.
- [2] L. Orman, "Transaction repair for integrity enforcement," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 996-1009, Nov.-Dec. 2001.
- [3] P. Lettieri, C. Fragouli, and M. Srivastava, "Low power error control for wireless links," in *Proc. ACM MobiCom*, 1997.
- [4] F. Liu, C. Lin, "An Analytical Model to Study the Packet Loss Burstiness over Wireless Channels," in *Proc. IEEE GLOBECOM'10*, Dec. 2010.
- [5] W. Peterson and E. J. Weldon, *Error-Correcting Codes*. Cambridge, MA: MIT Press, 1972.
- [6] G. Haflinger and O. Hohlfeld, "The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet," in *Proc. IEEE MMB'08*, Mar. 2008.