

Network Embedded FEC for Overlay and P2P Multicast Over Channels with Memory

Mingquan Wu and Hayder Radha
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824
e-mail: {wumingqu, radha}@egr.msu.edu

Abstract -

Forward Error Correction (FEC) has been used extensively for distributing multimedia contents over the internet. Traditionally, FEC can only be implemented in an end-to-end mode. In [3] we have introduced network embedded FEC (NEF) framework and showed that NEF can greatly improve the decodable probability and message goodput in overlay and peer-to-peer (p2p) multicast networks. In NEF, FEC encoders and decoders (codec) are placed in the intermediate nodes of a network; the NEF codecs detect and recover lost packets within FEC blocks at earlier stages before these blocks arrive at deeper intermediate nodes or at the final leaf nodes. Analysis in [3] is based on the assumption that packet losses follow a binomial distribution. In practice, packet losses are often correlated and occur in bursts. In this paper, we evaluate the performance of NEF within random multicast distribution trees that exhibit losses-with-memory over their branches. First, we derive a closed expression for the probability that a node receives i packets as the sender sends n packets when the packet losses follow a Gilbert model. Based on this analysis, we evaluate the impact of various loss burst lengths on the performance of NEF within random multicast trees. Results show that even with very high packet loss correlation, NEF outperforms end-to-end FEC and can improve decodable probability and message goodput dramatically.

I. INTRODUCTION

Forward Error Correction (FEC) has been used extensively for distributing multimedia contents over the Internet [1][2]. Under traditional end-to-end FEC, an encoder takes k message packets and produces $n-k$ parity packets; these message and parity packets are then combined to constitute an n -packet FEC block. A receiver that receives any k of the n packets in the FEC block can decode the FEC block and recover the original k message packets.

Traditionally, FEC has been employed on an end-to-end basis over multicast trees. Hence, those nodes that are far away from the source suffer high packet losses and may not be able to receive enough packets to recover the original FEC blocks. In [3], we introduced NEF for overlay and P2P multicast networks. In NEF, FEC codecs are placed optimally in the intermediate nodes of a network. Analysis and simulation have shown that NEF outperforms FEC dramatically in decodable probability and message goodput.

Analysis in [3] is based on the assumption that packet losses follow a binomial distribution. In practice though packet losses are correlated and often occur in bursts [4][5]. It is well known that the length of packet loss burst has a significant impact on the performance of FEC and hence, on the performance of NEF; yet quantifying this impact is a challenging problem.

Several models have been established to capture the stochastic characteristic of the underlying packet loss process of the internet traffic, including the Gilbert model [4][5], the general k order Markov Chain [5] and the loss run length model [6]. Compared with the later two, the Gilbert model gives an acceptable level of accuracy and yet its computational complexity is much lower. In this paper, we adopt the Gilbert model for packet losses over links of random multicast trees.

In [7], Yee and Weldon provided an approach for evaluating the Bit Error Probability (BEP) of the Gilbert-Elliott channel using a combinatorial method. This method, however, is rather complicated to evaluate and does not lend itself to simple analysis of complex networks. In [8], the author provides a powerful tool for the analysis of general Markov processes using a system analysis approach. Based on the framework described in [8], we present a new method for evaluating any desired loss/recovery probability measure for the Gilbert model. In particular, we present a rather elegant and simple approach for evaluating the probability of receiving i packets among n packets transmitted over Gilbert erasure channels. Based on this analysis, we can evaluate the impact of loss burst length on the performance of NEF over networks that exhibits losses-with-memory.

The remainder of the paper is organized as follows. Section II provides the probability analysis of the Gilbert model. Section III evaluates the packet loss correlation on the performance of NEF. A brief summary is presented in Section IV.

II. FEC DECODABLE PROBABILITY ANALYSIS FOR THE GILBERT MODEL

A Gilbert model can be represented by a two-state Markov chain. The state diagram of the Gilbert model is shown in Figure 1a, where G and B represent the Good state and Bad state, respectively. If the process is in Good state, the receiver receives all packets; if the process is in Bad state, all packets are lost.

In [8], the author has shown that a Markov process can be viewed as linear, discrete, time-invariant probabilistic matrix system. Domain transform and flow graph simplification methods can be used to analyze the Markov process. The flow graph of the Gilbert model can be obtained from the state diagram by multiplying each transition probability by z . Figure 1b shows the flow graph of the Gilbert model. Here, we have added two input and two output terminals to the flow graph, which means that the Gilbert model can be represented as a multiple input multiple output system. The system can start from Good or Bad state, and can end in Good and Bad state.

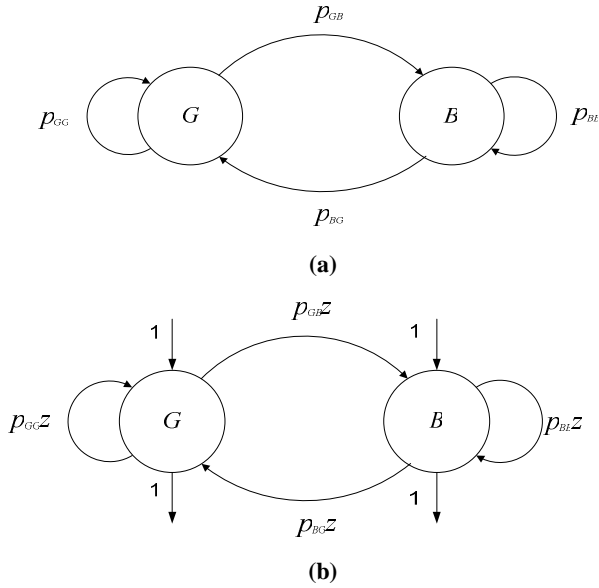


Figure 1 (a) state diagram and (b) flow graph of Gilbert model

To evaluate the desired probability measure for the Gilbert model, we construct another Markov process by extending the two-state Gilbert model using the number of correctly received packets as the indexes for the states of the extended Markov process. For example, if the receiver correctly receives i packets and the channel is in a good state, then the process is in state G_i ; on the other hand, if the receiver correctly receives i packets and the channel is in a bad state, then the process is in state B_i . The state transition of the new (extended) Markov process is shown in Figure 2a. The transition probabilities from good states to bad states and vice versa, remain the same as that of the two-state Gilbert model. From the state transition diagram, we can get the flow graph of the extended Gilbert model as shown in Figure 2b.

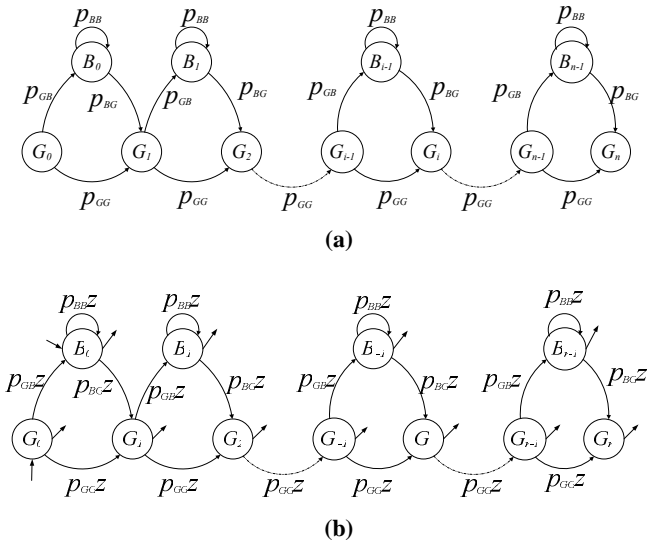


Figure 2 (a) state diagram and (b) flow graph of the extended Gilbert model

As we can see from the flow graph, the system can start at G_0 or B_0 and can end in any state. The initial probabilities that the system starts at G_0 and B_0 are the steady state probabilities of the original Gilbert model $\pi(G)$ and $\pi(B)$, respectively.

The probability that the sender transmits n packets and the receiver correctly receives i packets is the probability that the process starts at G_0 or B_0 and is in state G_i or B_i after n stages. We use $\phi_{G_0G_i}(n)$ to represent the multistep transition probability from G_0 to G_i :

$$\phi_{G_0G_i}(n) = P\{s(n) = G_i \mid s(0) = G_0\} \quad (1)$$

where $s(j)$ is the state of the extended Markov process at time index j . Using the method described in [8], the geometric transform of $\phi_{G_0G_i}(n)$ can be obtained as:

$$\Phi_{G_0G_i}(z) = (p_{GG}z + \frac{p_{GB}p_{BG}z^2}{1-p_{BB}z})^i \quad 0 < i \leq n \quad (2)$$

Taking the inverse geometric transform, we have:

$$\phi_{G_0G_i}(n) = \begin{cases} \sum_{m=1}^i \binom{i}{m} \binom{n-i-1}{m-1} p_{GB}^m p_{BG}^m p_{GG}^{i-m} p_{BB}^{n-i-m} & 0 < i < n \\ 0 & i = 0 \\ p_{GG}^n & i = n \end{cases} \quad (3)$$

The detail of the derivation of this equation is outlined in Appendix A.

Similarly, it can be shown that:

$$\phi_{B_0B_i}(n) = \begin{cases} \sum_{m=0}^{i-1} \binom{i-1}{m} \binom{n-i-1}{m} p_{GB}^{m+1} p_{BG}^m p_{GG}^{i-m-1} p_{BB}^{n-i-m-1} & 0 \leq i < n \\ 0 & i = n \end{cases} \quad (4)$$

$$\phi_{B_iG_i}(n) = \begin{cases} \sum_{m=0}^{i-1} \binom{i-1}{m} \binom{n-i}{m} p_{GB}^m p_{BG}^{m+1} p_{GG}^{i-m-1} p_{BB}^{n-i-m-1} & 0 < i \leq n \\ 0 & i = 0 \end{cases} \quad (5)$$

$$\phi_{B_iB_i}(n) = \begin{cases} \sum_{m=0}^{i-1} \binom{i-1}{m} \binom{n-i}{m+1} p_{GB}^{m+1} p_{BG}^{m+1} p_{GG}^{i-m-1} p_{BB}^{n-i-m-1} & 0 < i < n \\ p_{BB}^n & i = 0 \\ 0 & i = n \end{cases} \quad (6)$$

If we use $\phi(n, i)$ to represent the probability that the sender transmits n packets and the receiver correctly receives i packets, then:

$$\begin{aligned} \phi(n, i) = & \pi(G)(\phi_{G_0G_i}(n) + \phi_{G_0B_i}(n)) \\ & + \pi(B)(\phi_{B_0G_i}(n) + \phi_{B_0B_i}(n)) \end{aligned} \quad (7)$$

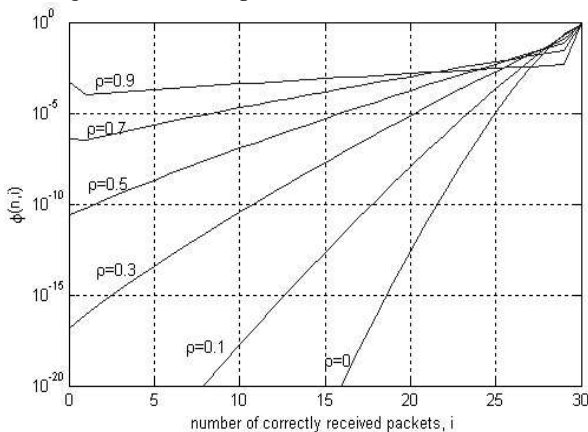
where $\pi(G) = \frac{P_{BG}}{P_{GB} + P_{BG}}$ and $\pi(B) = \frac{P_{GB}}{P_{GB} + P_{BG}}$ are the steady state probabilities of the original Gilbert model. The desired probability measure $\phi(n, i)$ can be completely evaluated using any two parameters that characterize the underlying Gilbert erasure channel. Traditionally, the transitional probabilities P_{GB} and P_{BG} (or $P_{GG} = 1 - P_{GB}$ and $P_{BB} = 1 - P_{BG}$) are used for such characterization. A more useful insight and analysis can be gained by considering other parameter pairs. In particular, in [7] the authors have used the average loss rate p and the *packet correlation* ρ to represent the state transition probabilities, where:

$$P_{GB} = p(1 - \rho); \quad P_{BG} = (1 - p)(1 - \rho)$$

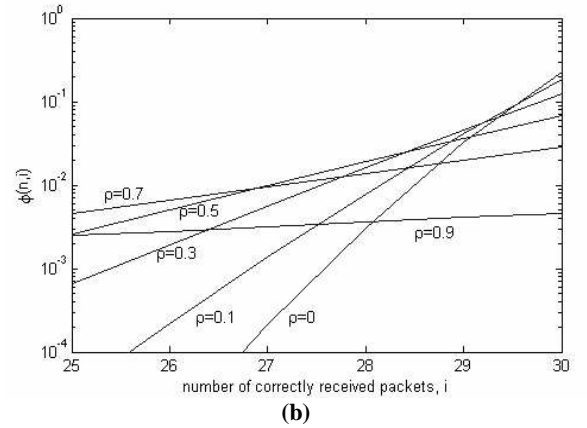
The steady state probabilities are directly related to the average loss rate p : $\pi(G) = 1 - p$ and $\pi(B) = p$. The packet erasure correlation ρ provides an average measure of how the states of two consecutive packets are correlated to each other. In particular, when $\rho = 0$, then the loss process is memoryless, and the above probability measures reduce to the special case of a memory-less Binary Erasure Channel (BEC). On the other hand, as the value of ρ increases, then the states of two consecutive packets become more and more correlated. Hence, we find the parameters p and ρ provide an intuitive, insightful, and broad characterization for the impact of channel coding on networks with losses.

Figure 3a plots the probability of a receiver correctly receives i packets when the source send n packets over the Gilbert channel. Here, n is set to 30, the average loss rate p is set to 1%, and the packet correlation ρ is changed from 0 to 0.9. As compared with the Binomial model (where $\rho = 0$), we can see as ρ increases, the probability of receiving a smaller number of packets increases.

Figure 3b is the detail view when the received packets i changes from 25 to 30. From the graph we can see that as the packet correlation increases, the probability of the receiver to receive higher number of packets decreases.



(a)

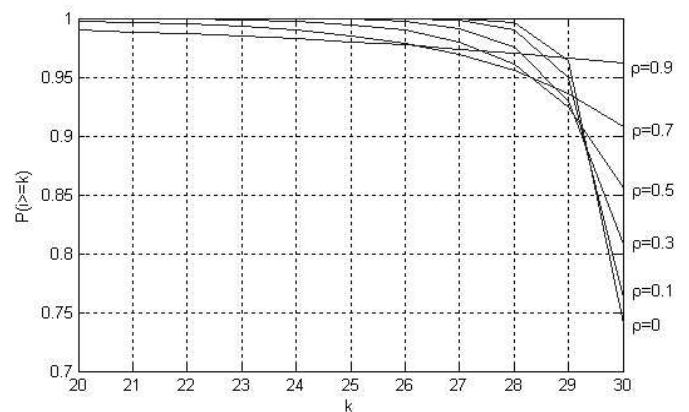

 Figure 3 (a) Probability of the receiver to receive i packets. (b) Detailed view when i changes from 25 to 30

For FEC codes, we are often concerned about the probability that a node can receive enough packets to decode an FEC block. For a (n, k) FEC code, this probability is:

$$P(i \geq k) = \sum_{i=k}^n \phi(n, i) \quad (8)$$

Figure 4 plots the average decodable probability of a receiver when the sender sends FEC blocks through a Gilbert channel. The block size n is set 30, the number of message packets in a FEC block k is changed from 20 to 30; $p = 1\%$ and ρ is changed from 0 to 0.9. The FEC code rate equals k/n .

From the plot, we can see that for a large portion of code rates, the higher the packet correlation, the lower the decodable probability; but this is the case when the code rate is very high. For example, when $k/n \geq 27/30$, the decodable probability for $\rho = 0.9$ is bigger than that when $\rho = 0.7$. In the Gilbert model, as ρ increases, the packet loss burst length increases. For FEC codes with high coding rate, any FEC block that suffers even a few packet losses may not be able to be decoded. Given the average loss rate unchanged, longer loss burst means fewer FEC blocks suffer from packet loss, this may result in an average higher decodable probability.


 Figure 4 decodable probability for a receiver when sender sends a (n, k) FEC block

III. IMPACT OF PACKET LOSS CORRELATION ON THE PERFORMANCE OF NEF

A. Analysis of NEF Based Routes

We want to evaluate the packet loss correlation on the performance of NEF on a general multicast distribution tree. The FEC codes we are going to use are Reed Solomon codes. In our analysis, we use the following notations:

$RS(n, k)$	Reed Solomon code with k data packets and $n-k$ parity packets.
$T ; T $	A multicast tree with a root node r & a total number of nodes $ T $.
$T^c ; T_l^c$	A sub-tree rooted at some node $c \in T$ but does not include the node c . The set of leaf nodes of T^c .
$P_v(i)$	Probability that node $v \in T$ receives exactly i packets.
$P_{v v-1}(i, j)$	Probability that node v receives i packets given that its parent $v-1$ sends j packets.
p	the packet loss probability between the link from $v-1$ to v

We use $P_{v|v-1}(i, j)$ to represent the probability that node v receives i packets given that its parent $v-1$ sends j packets. For links with memory-less losses, this probability is simply a binomial distribution:

$$P_{v|v-1}(i, j) = \binom{j}{i} (1-p)^i p^{j-i} \quad (9)$$

On the other hand, if the packet losses follow the Gilbert model, then

$$P_{v|v-1}(i, j) = \phi(j, i) \quad (10)$$

where from (7) we have:

$$P_{v|v-1}(i, j) = \pi(G)(\phi_{G_0 G_i}(j) + \phi_{G_0 B_i}(j)) + \pi(B)(\phi_{B_0 G_i}(j) + \phi_{B_0 B_i}(j))$$

When computing the probability $P_v(i)$ that a node v receives exactly i packets, we need to consider two cases; first, we consider the case when the parent node $v-1$ has no codec; second, we consider the case when the parent node $v-1$ has a NEF codec. If node v 's parent does not have a codec, the probability that node v receives i packets is:

$$P_v(i) = \sum_{j=i}^n P_{v-1}(j) P_{v|v-1}(i, j) \quad (11)$$

Note that $P_{v|v-1}(i, j) = 0, \forall j < i$. In other words, node v can receive i packets only when its parent $v-1$ sends at least i packets.

When a node has a codec for a $R(n, k)$ block, and if that node receives less than k packets and cannot decode the FEC block, it will just forward the received packets as usual; if it receives k or more packets, the node can decode the block and reconstruct the original data. It can also reproduce the lost parity packets. In fact, a codec can produce more or less than $n-k$ parity packets if desired; however, in this paper, we assume that the NEF codecs reconstruct the original data and reproduce the lost parity packets using the same $R(n, k)$ code. A node that has a NEF codec and which receives $k \leq j \leq n$ packets will send n packets. If the parent of node v has a codec, the probability that it receives i packets becomes

$$P_v'(i) = \begin{cases} \sum_{j=k}^n P_{v-1}(j) P_{v|v-1}(i, n) & k \leq i \leq n \\ \sum_{j=k}^n P_{v-1}(j) P_{v|v-1}(i, n) + \sum_{j=i}^{k-1} P_{v-1}(j) P_{v|v-1}(i, j) & 0 \leq i \leq k \end{cases} \quad (12)$$

For the root node (r) of the tree, we define

$$P_r(i) = \begin{cases} 0 & 0 \leq i \leq n-1 \\ 1 & i = n \end{cases} \quad (13)$$

Equation (11) and (12) are recursive functions, with the initial condition from (13), we can calculate the probability $P_v(i)$ for any node v in the multicast tree.

Here we use P_v^{dec} to represent the probability that node v can decode a $RS(n, k)$ block:

$$P_v^{dec} = P_v(i \geq k) = \sum_{i=k}^n P_v(i) \quad (14)$$

For $p2p$ network, all nodes in the multicast tree are end users; for proxy-based networks, only leaf nodes are end users. We define the average decodable probability of a tree T for $p2p$ and proxy-based overlay networks, respectively, as:

$$P_{avg}^{dec} = \frac{\sum_{v \in T-r} P_v^{dec}}{|T|-1} \quad (15); \quad P_{avg-leaf}^{dec} = \frac{\sum_{v \in T_l^r} P_v^{dec}}{|T_l^r|} \quad (16)$$

B. Results

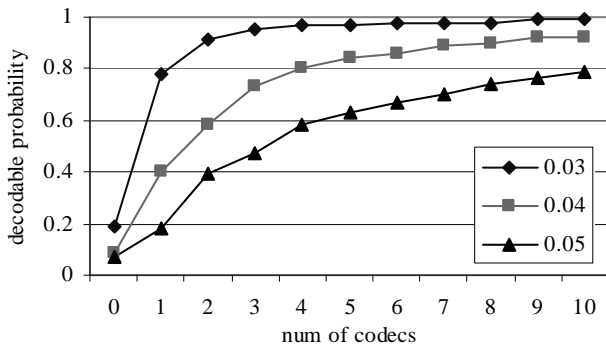
We designed a greedy algorithm to place a specified number of codecs in the intermediate nodes of a random tree. The greedy algorithm maximizes the average decodable probability. The greedy algorithm finds the best location for the first codec, then the next best location for the second one, and so on. Let $T^c \subset T$ be the sub tree rooted at node $c \in T$ not including c . If c is set as a "codec node", only those nodes $v \in T^c$ will benefit from this selection; meanwhile, the "codec node" c

itself will not be affected. For nodes $v' \in T - T^c$, everything remains unchanged. Let P_v^{dec} and $P_v'^{dec}$ denote the average decodable probability for node $v \in T^c$ before and after node c is set as a codec node, respectively. We need to find $c \in T$ that maximizes the following:

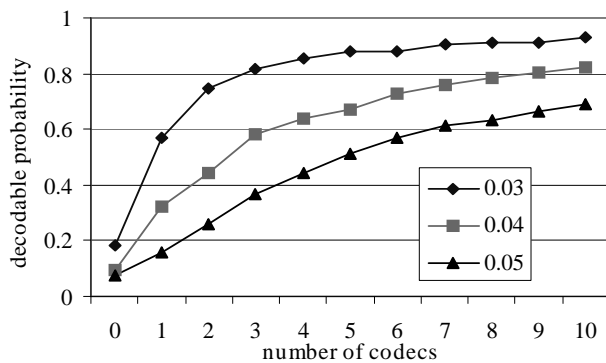
$$\max_{c \in T} \left(\sum_{v \in T^c} (P_v'^{dec} - P_v^{dec}) \right)$$

We implement the greedy algorithm in ten random trees of 100 nodes. The average loss rate per-link is set to 3%, 4% and 5% respectively. The FEC code we use here is $RS(223,255)$. We calculate the average decodable probability for different packet loss correlation ρ . The number of codecs is increased from 0 to 10. The results are the average of ten random trees. We have also calculated average message goodput, the results have the same pattern as that of the decodable probability. Because of the space limitation, the results for the messages goodput are not shown here.

Figure 5 plots the decodable probability when packet loss correlations ρ are set to 0 and 0.5 respectively. As we can see from the graph, higher packet loss correlation causes the overall performance of NEF decreases, but still, as the number of codecs increases, the decodable probability improves dramatically.



(a)



(b)

Figure 5 decodable probability for (a) $\rho=0$ and (b) $\rho=0.5$

In order to get a better understanding of the impact of the packet correlation on the effectiveness of NEF, we analyzed

the decodable probability when the number of codec is set to 5 and the packet correlation is increased from 0 to 0.9. The result is plotted in Figure 6. Here, it can be observed that the decodable probability decreases as ρ increases from 0 to 0.9.

As the erasure correlation increases, the FEC codec performance is reduced due to an increase in the number of “long bursts” of lost packets in an FEC block. Nevertheless, the overall performance of the 5-NEF codecs stays very close to the optimum performance (at $\rho=0$), and does not drop-off significantly only at high values of ρ when it approaches 1. For very high packet correlation, we can use interleaving or longer code words to further improve the performance of NEF.

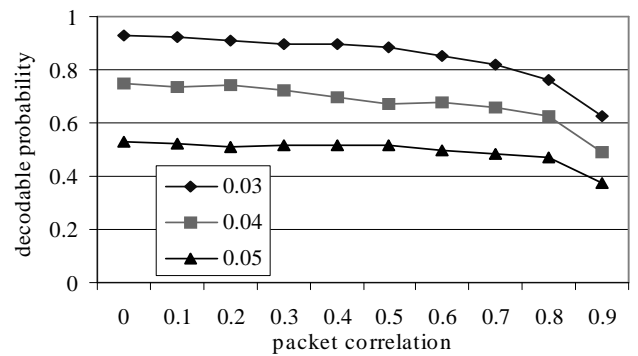


Figure 6 decodable probability when the number of codec is set to 5 and packet correlation is changed from 0 to 0.9

IV. SUMMARY

In this paper, we derived a closed function to calculate the probability of a receiver to receive i packets when a sender sends n packets through the Gilbert channel. Based on this analysis, we evaluate the impact of packet correlation on the performance of NEF over random multicast trees that consists of erasure channels with memory. Results show that NEF is very effective in environment where packet losses have a strong correlation.

REFERENCES

- [1] J.C Henion, “An Efficient Software Implementation of a FEC code”, Proc. of IDMS’97, Darmstadt, Germany, Sept.1997.
- [2] Luigi Rizzo, Lorenzo Vicisano, “RMDP: an FEC based Reliable Multicast protocol for wireless environments,” ACM Mobile Computing and Communication Review, vol. 2, no. 2, April 1998.
- [3] Hayder Radha and Mingquan Wu, “Overlay and Peer-to-Peer Multimedia Multicast with Network-Embedded FEC,” IEEE International Conference on Image Processing (ICIP), September 2004.
- [4] W. Jian “QoS measurement and Management for Internet real-time multimedia services”, Ph.D thesis, Department of Computer Science, Columbia University, 2003
- [5] M. Yajnik, J. Kurose, D. Towsley, “Packet Loss Correlation in the Mbone Multicast Network,” *IEEE Global Internet Conference*, London, November 1996.

- [6] H. Sanneck and G. Carle, "A framework model for packet loss metrics based on loss runlengths", *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000*. San Jose, CA, USA, pp.177-187.
- [7] J. R. Yee and E. J. Weldon, "Evaluation of the performance of error-correcting codes on a Gilbert channel," *IEEE Trans. Commun.*, vol. 43, pp. 2316-2323, Aug. 1995.
- [8] R. A. Howard, *Dynamic Probabilistic Systems*. New York: Wiley, 1971.
- [9] M. Zorzi, R. Rao, "Lateness Probability of a Retransmission Scheme for Error Control on a Two-state Markov Channel", *IEEE Transactions on communications*, Vol. 47, NO. 10, pp.1537-1548.

APPENDIX A

From Figure 2(b), the flow graph from G_0 to G_i can be simplified as:

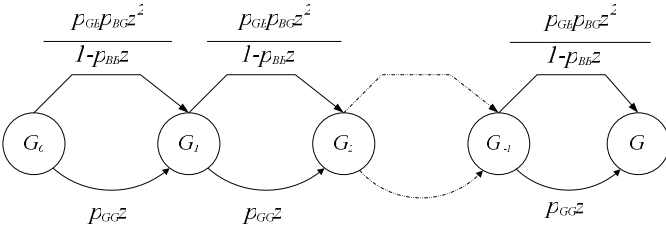


Figure 7 Simplified flow graph from G_0 to G_i

From the flow chart we can express equation (2) as:

$$\Phi_{G_0 G_i}(z) = \left(p_{GG} z + \frac{p_{GB} p_{BG} z^2}{1 - p_{BB} z} \right)^i \quad 0 < i \leq n$$

Now let's find the inverse geometric transform of $\Phi_{G_0 G_i}(z)$.

$$\begin{aligned} \Phi_{G_0 G_i}(z) &= p_{GG}^i z^i \left(1 + \frac{p_{GB} p_{BG} z}{p_{GG} (1 - p_{BB} z)} \right)^i \\ &= p_{GG}^i z^i \sum_{m=0}^i \binom{i}{m} \left(\frac{p_{GB} p_{BG} z}{p_{GG} (1 - p_{BB} z)} \right)^m \\ &= p_{GG}^i z^i \sum_{m=0}^i \binom{i}{m} \left(\frac{p_{GB} p_{BG}}{p_{GG}} \right)^m z^m \left(\frac{1}{1 - p_{BB} z} \right)^m \\ &= p_{GG}^i z^i + \sum_{m=1}^i \binom{i}{m} p_{GB}^m p_{BG}^m p_{GG}^{i-m} z^{m+i} \left(\frac{1}{1 - p_{BB} z} \right)^m \end{aligned}$$

We know that for $m \geq 1$ the inverse transform of

$$\left(\frac{1}{1 - p_{BB} z} \right)^m \text{ is } \frac{1}{(m-1)!} (n+1)(n+2)\dots(n+m-1) p_{BB}^n$$

$$\text{The inverse transform of } z^{m+i} \left(\frac{1}{1 - p_{BB} z} \right)^m$$

would then be

$$\frac{(n-m-i+1)(n-m-i+2)\dots(n-i-1) p_{BB}^{n-m-i}}{(m-1)!}$$

$$= \binom{n-i-1}{m-1} p_{BB}^{n-m-i}$$

From above we can get:

$$\begin{aligned} \phi_{G_0 G_k}(n) &= p_{GG}^i \delta(n-i) \\ &+ \sum_{m=1}^i \binom{i}{m} \binom{n-i-1}{m-1} p_{GB}^m p_{BG}^m p_{GG}^{i-m} p_{BB}^{n-i-m} \end{aligned}$$

When $k = 0$, it is trivial to get $\phi_{G_0 G_i}(n) = 0$.

From this we prove equation (3) is correct. Equation (4) to (6) can be derived in a similar manner. It is worth noting that [9] derives a similar expression, however using a very different approach and different notations.