Optimal Broadcast Scheduling for Random-Loss Channels

Cheng Huang Lihao Xu
Dept. of Computer Science and Engineering,
Washington University in St. Louis, MO, 63130
{cheng, lihao}@cse.wustl.edu

***Abstract*— Broadcast scheduling is an important problem that has been an active research topic for nearly two decades. Virtually all known results on broadcast scheduling, however, have assumed that the broadcast channels are reliable without data corruption or loss, which is certainly far from reality. In fact, data loss imposes severe impacts on broadcast performance, namely, greatly increasing the overall time that receivers need to receive their desired data items, as briefly shown in [9].**

In this paper, we study how to systematically derive optimal schedules for random-loss broadcast channels with any given data loss probability. The key idea is to employ proper MDS (Maximum Distance Separable) codes, a class of error correcting codes, in the schedules. We provide a unified framework for analyzing the performance of a broadcast schedule, based on which the optimal schedule can be derived to minimize a schedule's expected delivery time averaged over all the receivers and the data items. We also show the robustness of the optimal schedule, whose performance degrades only slightly even when the channel changes significantly from its target random loss probability.

I. Introduction

The increasing deployment of wireless networks, in both local and wide areas, together with various personal communication devices, from mobile phone to PDAs, even to watch (such as the one being developed by Microsoft), has made ubiquitous information access possible. The inherent broadcast nature of wireless channels calls for more efficient and effective information delivery schemes other than traditional request-and-transmit type unicast approaches. In many applications, broadcast is much more efficient to disseminate popular information, such as news, weather, traffic, market and entertainment information, to a large group of users. Broadcast is also an effective means for certain wired networks, such as residential cable networks and institutional local area networks.

For a set of popular data items (simply *items* hereafter) and a group of receivers, a broadcast schedule (*schedule* hereafter) decides which item to broadcast and when. The performance of individual schedule is usually measured by the average time to deliver the items to their desired receivers. Finding the *optimal schedules*, which minimize this average delivery time, has been drawing active research efforts for nearly two decades [4], [5], [18], [14], [10], [7], [8], [9], [1], [2], [6], [11], [15], [16], [12], [3], [17].

Among volumes of previous works, Ammar and Wong [4], [5] proved that, for optimal broadcast scheduling, each item should appear cyclically and be equally spaced. Vaidya and Hameed [14], [10] showed that the optimal broadcast frequency of an item is a function of the item length and the demand probability. As significant these results are, they were all based on a common assumption, namely, items are *not* separable. The non-separability has an implication that a receiver simply discards a partial item if it does *not* start receiving from the beginning of it. From the perspective of real system practice, however, items are always broadcast in the form of network packets (or frames). And thus it is natural to see that the delivery time can be reduced with the usage of receiver buffer, which has certainly become accessible due to the hardware advances in wireless devices. Foltz and Bruck [7], [8] explored this direction and derived the optimal schedules, when items are splitted into two halves. In this paper, we pursue this direction further and consider packet as the basic unit in scheduling, which allows more flexibility and also provides more generalized results.

Even more importantly, errors in any communication channels are unavoidable, and broadcast channels are no exception. In fact, wireless channels tend to have more errors than wired channels. Errors usually occur in the form of data loss, since corrupted data can be masked as data loss by various layers in a communication system. Data loss imposes severe impacts on broadcast performance, as briefly shown in [9]. However, most existing schedules, except only a few [14], [10], [9], assume the broadcast channels are reliable and no data loss occurs during transmission. In [14], [10], *Error Correction Codes* (ECC) are applied to combat data loss and a receiver discards an entire corrupted data block if it can't be fully recovered. As we will show in this paper, surviving blocks of corrupted data can still contribute to the recovery of original items and significantly reduce the delivery time when proper coding schemes are employed. To the best of our knowledge, [9] first introduced MDS codes in scheduling and demonstrated its effectiveness in reducing the delivery time when data loss happens. However, its channel model and thus the analysis, which assumes *single* loss in any reception period, is over simplified. In this paper, we extend this idea and present a unified analysis framework for random-loss channels with any given loss probability. Based on our analysis, we then show how to systematically derive the optimal schedules.

Our contributions in this paper are four folds. First, we prove that optimal broadcast scheduling can be achieved using MDS codes. Second, we provide a unified framework to analyze the scheduling performance. A very useful property is shown to greatly simplify and facilitate numerically finding the optimal schedule, given a channel loss probability. Third, we show the robustness of the optimal schedule, whose performance degrades only slightly even when the channel changes significantly from its target loss probability. Finally, we show that using packet as the basic unit improves scheduling performance than previous schemes that do *not* separate items or merely split them into halves. The improvement is especially prominent when data loss presents.

The paper is structured as follows: Sec II presents an MDS code based scheduling scheme for loss resilient broadcast. Then a unified analysis framework is derived for a simplified scenario. The robustness of the optimal schedule is then demonstrated. Sec III extends the analysis to a more general scenario. And the improvement of scheduling using packet as the basic unit is also presented. Sec IV concludes the paper.

II. Optimal Scheduling with Coding

A. Broadcast Scheduling Model

We first describe a suitable model for the broadcast scheduling problem. The broadcast channel has simple random data loss, where each data packet has an *identical* yet *independent* loss probability p. In this paper, we focus on *periodic* schedules. As already shown in [4], [5], the optimal broadcast schedules (for lossless channels) are periodic. Thus a schedule can be represented by one of its periods. We use a simple notation to represent a schedule, where 'I' means to send the first data item, and 'II' means to send the second data item. (In this paper, we limit our discussion to two data items, although general principles and framework to be discussed apply to scheduling for more data items.) Each data item has k data packets. For example, the schedule I^1II^3 refers to alternately sending 1 packet of data item I, followed by 3 packets of data item II in one period. Again, for simplicity, this paper only discusses schedules with the form I^iII^j, where i and j are nonnegative integers. Finally, we normalize data size and channel bandwidth, and thus data delivery time.

For a given schedule, we use the *Expected Delivery Time* (EDT) to measure its performance, which is defined to be the expected time receivers need for their desired items to deliver to them, averaged over all the receivers and the data items. We assume receivers start waiting for data items at times that are *uniformly* distributed over a broadcast schedule period. (It is not hard to generalize related results in this paper to non-uniform receiver arrival times.) The delivery time of a data item includes two parts, the *waiting time* and the *transmission time*. The waiting time is the time a receiver needs to wait while its wanted item is *not* being sent over the channel. The transmission time is the one the receiver needs to finish receiving its desired item from the channel while it is being sent. At last, each data item has a demand probability associated with it, i.e., item I is demanded by a probability p_I, and item II by a probability p_{II}, where $p_I + p_{II} = 1$.

The following example explains the EDT calculation of a schedule of two items. The two items both have the same normalized size 1 with k packets. There is no data loss in the broadcast channel.

Example 1: EDT of Schedule I^kII^k
If a receiver wants item I, the delivery time will be 2 if it starts listening during the broadcast of item I. It simply receives item I starting somewhere in the middle, waits 1 time unit while item II is being transmitted, and then receives the rest of item I, for a total delivery time of 2. If the receiver starts listening during the broadcast of item II, it waits through the remainder of item II, and then receives item I, for a total delivery time between 1 and 2, depending on the initial listening time. Thus the average delivery time T_I for item I is the average of 2 and $\frac{3}{2}$ (the average of the values between 1 and 2), i.e., $T_I = \frac{7}{4}$. The analysis for a receiver wanting item II is similar, and $T_{II} = \frac{7}{4}$. Then $EDT = p_I T_I + p_{II} T_{II} = \frac{7}{4}$ □

B. Broadcast Using MDS Codes

An (n, k) error correcting code encodes k *message* packets to n *codeword* packets, where of course $n \geq k$. Such a code can usually tolerate date loss of r packets during transmission, where obviously $r \leq n - k$. Its coding rate, defined as k/n, decides its effective use of network bandwidth. On the other hand, r determines the code's loss recovery capability. It is easy to see the larger r is, the more capable the code is in loss recovery, but the lower its coding rate is, i.e., the less effective it consumes network bandwidth. When $r = n - k$, the code meets the Singleton Bound, and it is *MDS* (Maximum Distance Separable) [13]. An MDS code is optimal in terms of its data loss recovery capability with a given coding rate. Another way of interpreting the MDS property is that the entire original k message packets can be fully recovered from *any* k codeword packets.

As shown in [9], a schedule employing a proper MDS code can drastically reduce its EDT. The key idea is to encode a k-packet data item to an n-packet MDS code codeword. Then the codeword packets are sequentially used to replace the original data item in the schedule periods. In the event of data losses, any codeword packet can contribute to the recovery of original data item. When the codeword length n is sufficiently large so that a receiver can receive at least k different codeword packets in a single reception period, the desired original data item can be easily recovered after any k codeword packets are received. See the details in [9].

A natural question then is: is MDS code a best coding scheme for scheduling? We give an affirmative answer by proving the following theorem:

Theorem 1: For any fixed schedule, broadcast using MDS codes requires the minimum delivery time among all possible coding schemes.

Proof: For a fixed schedule, suppose a receiver completes reception with delivery time d after receiving l packets of its interested item. Then, $l \geq k$ must satisfy no matter what coding scheme is used. For the same loss pattern, if MDS codes are applied, then $l = k$ and the delivery time must be no greater than d. Therefore, broadcast using MDS codes requires the minimum delivery time and is thus optimal. □

There are two direct conclusions from Theorem 1. First, the optimal broadcast schedule can be found among schedules using MDS coding scheme (which is referred to as *CODING* hereafter). Second, broadcasting message packets directly (referred to as *NO CODING* hereafter) can be regarded as using a special coding scheme, a (k, k) code, and is worse than the CODING scheme. However, it is still of interest to quantitatively compare the performances of the two schedules to justify the computation overhead of the CODING scheme.

C. Optimal Broadcast Scheduling with the CODING Scheme

We first focus on a simplified scenario by assuming the demand probabilities $p_I = p_{II} = 1/2$. Then, it is conceivable that the optimal schedule with the form $\mathrm{I}^i\mathrm{II}^j$ satisfies $i = j$ because of the symmetry between items I and II. A $(wn,\ k)$ MDS code is applied to each item such that wn codeword packets are generated from the original k message packets. Then, the first n codeword packets of item I are broadcast, followed by the first n codeword packets of item II. This completes the first broadcast period. The second period broadcasts the second n codeword packets of item I and item II, respectively, and so on. The entire schedule repeats after completing all the wn codeword packets of the both items. w is chosen to be big enough such that a single reception does *not* last more than wn packets. Then, a data item can be successfully recovered if *any* k codeword packets of this item are received. This defines a schedule, based on MDS code. The schedule has the form of $\mathrm{I}^n\mathrm{II}^n$ and independent of w.

A natural question then is: for a fixed item size k, what is the *optimal value* for the block length n, given the channel loss probability p, so that the corresponding EDT is minimized?

Let the EDT of item I and II be $EDT(I)$ and $EDT(II)$, respectively. Since $EDT(I) = EDT(II)$ and $p_I = p_{II} = \frac{1}{2}$, then

$$EDT = p_I EDT(I) + p_{II} EDT(II) = EDT(I) \quad (1)$$

thus it is sufficient to focus on data item I for our analysis.

For analysis simplicity, assume a receiver starts listening at the beginning of each packet. (This changes the delivery time at most by the transmission time of one packet and is thus negligible when k is sufficiently large.) If the receiver starts at the i^{th} packet of an item I block, and finishes at the j^{th} packet of another item I block, with m item II blocks in between. The total number of packets (including both items) is

$$N = 2mn + j - (i - 1)$$

among them M are item I packets, where

$$M = N - mn = mn + j + 1 - i$$

where $1 \leq i, j \leq n$, $m \geq 0$ with the constraint $M \geq k$. Thus the normalized delivery time for item I is N/k.

Among the N packets, the loss patten of item II packets is irrelevant, since the receiver is only interested in item I. Then among the M item I packets, exactly k of them are received while all the others are lost during the broadcast. Also note the last item I packet must be received. Thus the probability of this event is $\binom{M-1}{k-1}p^{M-k}(1-p)^k$.

Let $dt_I(i)$ be the normalized average delivery time when the receiver starts listening from the i^{th} packet of item I, then

$$dt_I(i) = \sum_{(j,m):M\geq k} \frac{N}{k}\binom{M-1}{k-1}p^{M-k}(1-p)^k$$
$$= \sum_{(j,m):M\geq k} \frac{N}{M}\binom{M}{k}p^{M-k}(1-p)^k$$

Denote $dt_I(I)$ as the average delivery time when the receiver starts listening from item I packets, then

$$dt_I(I) = \frac{1}{n}\sum_{i=1}^{n} dt_I(i)$$

Similarly, denote $dt_I(II)$ as the average delivery time when the receiver starts listening from item II packets. We have

$$dt_I(II) = dt_I(1) + \frac{n}{2k}$$

Since the receiver has equal chance to start with item I or item II packets,

$$EDT(I) = \frac{1}{2}\big(dt_I(I) + dt_I(II)\big) \quad (2)$$
$$= \frac{1}{2}\big(dt_I(I) + dt(1)\big) + \frac{n}{4k} \quad (3)$$

which can be simplified by applying the following results.

Theorem 2: $dt_I(I)$ is independent of n and satisfies:

$$dt_I(I) = \frac{2}{1-p} - \frac{1}{k} \quad (4)$$

Proof: see the appendix. □

Since the schedule is completely determined by n, this theorem shows that $dt_I(I)$ is independent of it. Although not intuitive, this property is very useful in simplifying the calculation of EDT.

Thus far, the only complex term left in EDT is $dt_I(1)$, which can also be simplified as (see appendix for details):

$$dt_I(1)$$
$$= \frac{2}{1-p} - \sum_{m=0}^{\infty}\sum_{j=1}^{n}\frac{j}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \quad (5)$$
$$= \frac{1}{1-p} - \frac{n}{k} + \frac{(1-p)^k}{k!}\frac{d^{k-1}}{dp^{k-1}}\Big(\frac{n}{(1-p)(1-p^n)}\Big) \quad (6)$$

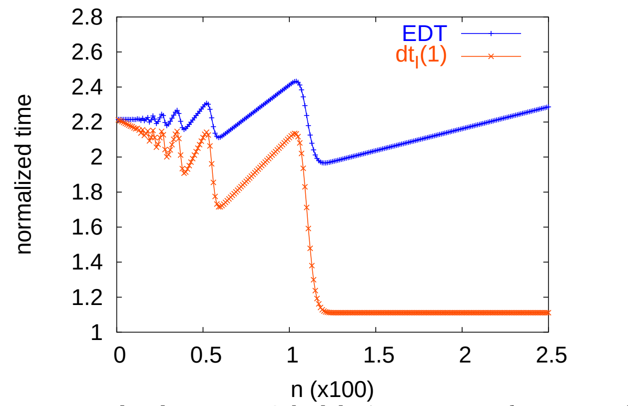

Fig. 1. Normalized Time vs. Schedule (message size $k = 100$ and channel loss probability $p = 0.1$),

Using (5), (4) and (3), it is easy to numerically compute EDT for different n and find the optimal schedule. Figure 1 shows the EDTs of different schedules with the message size $k = 100$ and the channel loss probability $p = 0.1$. We can see that $dt(1)$ converges to $\frac{1}{1-p}$ when n is sufficiently larger than k. This is true from (6), where the later two terms cancel out when n is large enough to make $(1-p^n)$ approach 1. Also intuitively, with a large block length n, a receiver starting at the first packet of item I block should always complete receiving k packets in current block. Then, the expected number of total packets is $\frac{k}{1-p}$ ($\frac{1}{1-p}$ after normalization) and independent of n. Thus, when n is large, EDT is dominated by $\frac{n}{4k}$, the second term of $dt_I(II)$, and grows almost linearly and unbounded. Therefore, we find the optimal schedule by numerically computing only EDTs of limited n. In this case, the schedule is optimal when $n = 120$.

D. Broadcast Scheduling with the NO CODING scheme

The analysis of EDT with the NO CODING scheme is briefly presented here for the purpose of comparison with the CODING scheme. The same notations for i, m, j, M and N are used. Also let $M = qk + r$ ($q \geq 0$, $0 < r \leq n$), where q and r represent *special* quotient and remainder. Note that the range of r suggests this representation is slightly different from normal division. It is clear that there are $q+1$ opportunities to deliver the last packet of item I, which must have been delivered ONLY ONCE. This happens with a probability of $p^q(1-p)$. There are $r-1$ packets which have been delivered as least once among $q+1$ opportunities, each with a probability of $1-p^{q+1}$. Similarly, the rest $k-r$ packets have been delivered at least once among q opportunities, each with a probability of $1-p^q$. Thus the overall probability of this event is

$$Pr = p^q(1-p)(1-p^{q+1})^{r-1}(1-p^q)^{k-r}$$

And $dt_I(i)$ can be calculated as

$$dt_I(i) = \sum_{(j,m):M \geq k} \frac{N}{k} Pr$$
$$= \sum_{(j,m):M \geq k} \frac{N}{k} p^q(1-p)(1-p^{q+1})^{r-1}(1-p^q)^{k-r}$$

With the *overall* EDT computed from (1) and (3), the optimal schedule can then be obtained numerically.

E. Optimality of the CODING Scheme

Now we compare the performance of the CODING and the NO CODING schemes. Figure 2 shows EDTs for various message sizes k over a channel with a random loss probability $p = 0.01$. It is clear that the gain of using MDS codes is significant, compared to schedules without any coding scheme. Two other performance curves are also presented for $n = k$. With this simple schedule choice, however, EDT deviates greatly from the optimal values even when the CODING scheme is applied. This demonstrates the importance of choosing a proper n for a schedule.

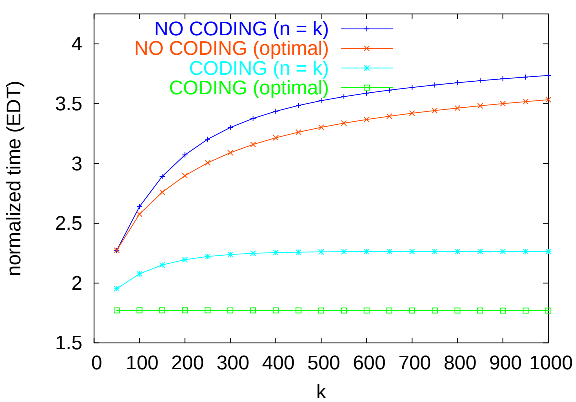

Fig. 2. EDT vs. Message Packet Number (Loss Probability $p = 0.01$)

F. Robustness of the Optimal Schedules

Broadcast channels usually have significant variations. Even when the broadcast environment is relatively stable, the reception quality of individual receivers is still likely to vary. To tolerate these dynamics, a common practice of system design is to target at the worst conditions. Similarly, we design the optimal schedule with the worst channel loss probability and evaluate its performance when actual channel characteristics are different.

Let the worst channel loss probability $p = 0.1$, which is truly an adversary value and rarely happens in a normal wired or even wireless network. Given the message size $k = 100$, the optimal schedule is $n = 120$ with the CODING scheme and $n = 300$ with the NO CODING scheme. Figure 3 shows the performance of both schedules under various loss probabilities. It is clear that performance of the schedules with the CODING scheme degrades only slightly for a wide range of channel conditions. In another word, the schedule with $n = 120$ is nearly optimal for virtually all loss probabilities less than 0.1. Scheduling with the NO CODING scheme, however, is significantly affected when the channel loss probability changes. The robustness property again shows the benefit of using MDS codes in scheduling.

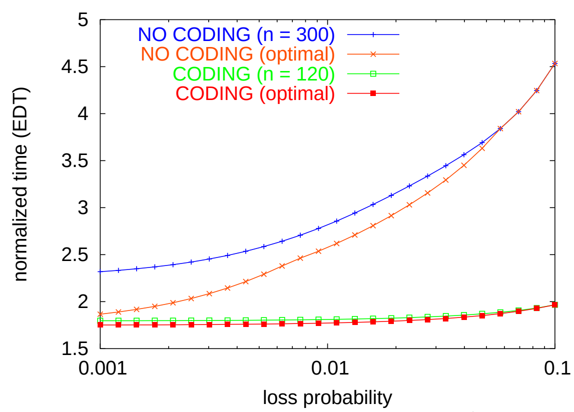

Fig. 3. EDT vs. Loss Probability (message size $k = 100$)

III. An Extension: Scheduling for Unequal Demand Items

In the previous section, we simplify the analysis by assuming equal demand probabilities for the both items I and II. In this section, the analysis is extended to a more general situation when $p_I \neq p_{II}$. Let n_I and n_{II} denote the codeword block length for items I and II, respectively. Note here the original items still have the same size k. But now two different MDS codes (a (wn_I, k) code and a (wn_{II}, k) code) are applied to the items. In general, $n_I \neq n_{II}$. We then consider the schedule is $\mathrm{I}^{n_I}\mathrm{II}^{n_{II}}$. Define the block length ratio $\rho = n_{II}/n_I$, then a result similar to Theorem 2 can be derived:

Theorem 3: $dt_I(I)$ is dependent only on ρ and satisfies:

$$dt_I(I) = \frac{1+\rho}{1-p} - \frac{\rho}{k}$$

Proof: see the appendix. □

Also, we can get $dt_I(1)$ (see appendix for details) as

$$\begin{aligned} & dt_I(1) \\ =& \sum_{m=0}^{\infty}\sum_{j=1}^{n_I}\Big\{\frac{mn_I+mn_{II}+j}{k}\binom{mn_I+j-1}{k-1} \\ & \times p^{mn_I+j-k}(1-p)^k\Big\} \\ =& \frac{1}{1-p}-\frac{\rho n_I}{k}+\frac{\rho(1-p)^k}{k!}\frac{d^{k-1}}{p^{k-1}}\Big(\frac{n_I}{(1-p)(1-p^{n_I})}\Big) \end{aligned}$$

It is easy to verify that when the block lengths are equal ($\rho = 1$), $dt_I(I)$ and $dt_I(1)$ reduce to simpler forms in the previous section. The $EDT(I)$ can then be computed:

$$EDT(I) = \frac{1}{1+\rho}dt_I(I) + \frac{\rho}{1+\rho}\Big(dt_I(1) + \frac{n_{II}}{2k}\Big) \quad (7)$$

We can get $dt_{II}(I)$, $dt_{II}(1)$ and $EDT(II)$ in a similar way. And

$$EDT = p_I EDT(I) + p_{II} EDT(II) \quad (8)$$

Therefore, given the message block length k, the broadcast channel loss probability p and the demand probabilities p_I, p_{II}, the optimal schedule (n_I and n_{II}) can be derived numerically.

In the remaining part of this section, we apply this extended analysis framework to two special cases. One considers lossless broadcast channels. In addition to verifying the previous results in [7], [9], we show that our results cover more general situations. The other examines random-loss channels and shows it is much more important to employ optimal schedules for such channels.

A. Case I: Does splitting help for lossless broadcast channels?

When the broadcast channels are lossless, the scheduling problem reduces to a simpler version, which has been addressed by previous works. [9] presents an optimal schedule when data items are non-separable (*no splitting*). And [7] derives the optimal schedule when each data item is splitted into two halves (*half splitting*). Our optimal schedule, however, splits items into multiple packets (*arbitrary splitting*) and uses packet as the basic scheduling unit. It is easy to see that the no splitting and half splitting are just special cases of the arbitrary splitting for scheduling.

We can further simplify EDT expression for lossless channels. Since there is no data loss at all, it is intuitive that the CODING scheme and the NO CODING scheme require the same delivery time. Thus, it is unnecessary to apply MDS codes. It is sufficient to broadcast the original message packets directly. However, it is useful to notice that the EDT analysis with the CODING scheme gives *exactly* the same result as the NO CODING scheme for this special case. Therefore, the results of $dt_I(I)$ and $dt_I(1)$ as derived can be directly applied here.

When the loss probability $p = 0$, from Theorem 3:

$$dt_I(I) = (1+\rho) - \frac{\rho}{k}$$

Moreover, $dt_I(1)$ can be further simplified, because the only term left in its expression requires the power of p to be 0. Thus, $mn_I + j = k$. When $n_I \geq k$, the only possible values are $m = 0$ and $j = k$. When $n_I < k$, then $m = q$ and $j = r$, if we define $k = qn_I + r$ with $q \leq 0$ and $0 < r \leq n_I$. Then

$$dt_I(1) = \begin{cases} 1 & n_I \geq k \\ 1+\rho(k-r) & n_I < k, k = qn_I + r \end{cases} \quad (9)$$

Again, $EDT(I)$ is computed from (7) with the above two terms. Similarly, $EDT(II)$ and then EDT can be obtained from (8).

Schedules with no splitting only allow the block lengths (n_I and n_{II}) to be multiples of k. For instance, n_I and n_{II} can be 100, 200, 300, etc., when the original message size $k = 100$. Similarly, schedules using half splitting allow the block lengths to be multiples of $k/2$. For example, n_I and n_{II} can be 50, 100, 150, etc. Schedules corresponding to arbitrary splitting cases can take any integer values for n_I and n_{II}, e.g., 113, 182, 231, etc.

Table I shows the optimal schedules and the corresponding EDT values for the above three cases, when the demand probability for item I varies from 0.05 to 0.5 (or symmetrically from 0.5 to 0.95). We can make several observations: 1) The results for no splitting and half splitting schedules nicely match the results in [9], [7]; 2) In general, half splitting is better than no splitting; and 3) arbitrary splitting yields the best EDT. This is intuitive since finer granularity allows more freedom in scheduling. However, notice that differences in EDT are subtle, which in fact suggests that even no splitting is close enough to the optimal schedules. This is no longer true for lossy channels, as will be shown in the following section.

B. Case II: Does splitting help for random-loss broadcast channels?

The same three cases: no splitting, half splitting and arbitrary splitting are examined here, except that the broadcast channels now have random losses with a probability $p = 0.01$. Figure 4 shows the results for both lossless and lossy channels. (The two lower curves

p_I	no splitting		half splitting		arbitrary splitting	
	$n_I : n_{II}$	EDT	$n_I : n_{II}$	EDT	$n_I : n_{II}$	EDT
0.05	100 : 600	1.37321	100 : 650	1.37247	100 : 646	1.37246
0.1	100 : 400	1.5074	100 : 400	1.5074	100 : 408	1.50733
0.15	100 : 300	1.59675	100 : 300	1.59675	100 : 299	1.59675
0.2	100 : 200	1.66267	100 : 250	1.66057	100 : 231	1.65951
0.25	100 : 200	1.70417	100 : 200	1.70417	100 : 182	1.70284
0.3	100 : 100	1.745	100 : 100	1.745	100 : 144	1.73021
0.35	100 : 100	1.745	100 : 100	1.745	100 : 113	1.7343
0.4	100 : 100	1.745	100 : 100	1.745	100 : 100	1.745
0.45	100 : 100	1.745	100 : 100	1.745	100 : 100	1.745
0.5	100 : 100	1.745	100 : 100	1.745	100 : 100	1.745

TABLE I

OPTIMAL SPLITTING FOR LOSSLESS CHANNELS ($k = 100$)

corresponding to the half splitting and the arbitrary splitting almost overlap each other, as shown in Table I.) Again, it is clear that splitting does *not* help much for lossless channels. However, the gain of splitting is obvious for lossy channels. This again justifies the need for better scheduling policy than brute-force half splitting or no splitting at all. Also notice that when the CODING scheme is applied, the performance in lossy channels is very close to that of lossless channels, which is consistent with the robustness discussion in section II-F.

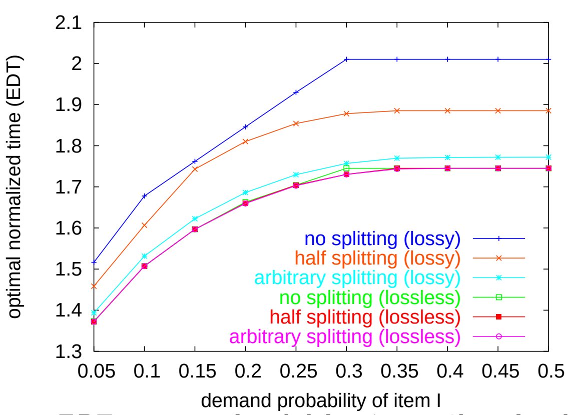

Fig. 4. EDT vs. Demand Probability (Lossy Channel with $p = 0.01$) (The three curves corresponding to lossless channels are tightly grouped together. See actual values in Table I.)

IV. CONCLUSIONS

In this paper, we propose to use proper MDS codes in broadcast scheduling to combat data losses. We provide a unified framework to analyze broadcast schedules for random-loss channels. Optimal schedules are derived with given data size and channel loss probability. We show that our optimal schedule is robust, i.e., almost immune to channel loss probability variations. This robustness helps broadcast system design and also helps to target wireless receivers with a wide variety of reception conditions. Extended analysis for unequal demand items generalizes and unifies the results from previous works.

REFERENCES

[1] S. Acharya, R. Alonson, M. Franklin, and S. Zdonik, “Broadcast Disks: Data Management for Asymmetric Communications Environments,” *Proc. 1995 ACM SIGMOD*, 199-210, May 1995.

[2] S. Acharya, M. Franklin, and S. Zdonik, “Dissemination-Based Data Delivery Using Broadcast Disks”, *IEEE Personal Communications*, 50-60, Dec. 1995.

[3] D. Aksoy, M. Altinel, R. Bose, U. Centemel, M. Franklin, J. Wang, and S. Zdonik, “Research in Data Broadcast and Dissemination,” *Proc. of 1st International Conference on Advanced Multimedia Content Processing*, Osaka, Japan, Nov. 1998.

[4] M. H. Ammar, and J. W. Wong, “The Design of Teletext Broadcast Cycles,” *Performance Evaluation*, vol. 5, pp. 235-242, Nov. 1985.

[5] M. H. Ammar, and J. W. Wong, “On the Optimality of Cyclic Transmission in Teletext Systems,” *IEEE Trans. on Communications*, vol. 35, 68-73, Jan. 1987.

[6] A. Bar-Noy, B. Patt-Shamir, and I. Ziper, “Broadcast Disks with Polynomial Cost Functions”, *Proc. INFOCOM 2000*.

[7] K. Foltz, and J. Bruck, “Splitting Schedules for Internet Broadcast Communication,” *IEEE Trans. on Information Theory*, 48(2), 345-358, Feb. 2002.

[8] K. Foltz, and J. Bruck, “Robustness of Time-Division Schedules for Internet Broadcast,” *Proc. of ISIT*, Lausanne, Switzerland, Jun. 2002.

[9] K. Foltz, L. Xu, and J. Bruck, “Coding and Scheduling for Efficient Loss-Resilient Data Broadcasting,” *Proc. of ISIT*, Yokohama, Japan, Jun. 2003.

[10] S. Hameed, and N. H. Vaidya, “Efficient Algorithms for Scheduling Data Broadcast,” *Wireless Networks*, 5(3), 183-193, 1999.

[11] S. Jiang, and N. H. Vaidya, “Scheduling Algorithms for a Data Broadcast System: Minimizing Variance of the Response Time,” *Technical Report 98-005*, Computer Science, Texas A&M University, Feb. 4, 1998.

[12] S. Khanna, and S. Zhou, “On Indexed Data Broadcast,” *Proc. of the 30th Annual ACM Symposium on Theory of Computing (STOC-98)*, pp. 463-472, New York, 1998.

[13] F. J. MacWilliams, and N. J. A. Sloane, *The Theory of Error Correcting Codes*, Amsterdam: North-Holland, 1977.

[14] N. H. Vaidya, and S. Hameed, “Scheduling Data Broadcast in Asymmetric Communication Environments,” *Wireless Networks*, 5(3), 171-182, 1999.

[15] C. J. Su, and L. Tassiulas, "Broadcast Scheduling for Information Distribution," *Proc. INFOCOM'97*, vol. 1, 109-117, 1997.
[16] L. Tassiulas, and C. J. Su, "Optimal Memory Management Strategies for a Mobile User in a Broadcast Data Delivery System," *IEEE JSAC Special Issue on Networking and Performance Issues of Personal Mobile Communications*, 15(7), 1226-1238, Sep. 1997.
[17] N. H. Vaidya, and S. Hameed, "Improved Algorithms for Scheduling Data Broadcast," *CS Tech. Report 96-029*, Texas A & M, Dec. 1996.
[18] J. W. Wong, and M. H. Ammar, "Analysis of Broadcast Delivery in a Videotex System," *IEEE Trans. on Computers*, Vol. 34, 863-866, Sep. 1985.

APPENDIX

A. Proof of Theorem 2

Let l denote the number of lost packets of item I. Then the total number of broadcast packets of item I is $k+l$. Also define $t_l(i)$ to be the delivery time of receiving *exact* k packets with l packet losses, when starting from the i^{th} packet in item I block. dt_l represents the average delivery time of $t_l(i)$ over all i's.

Let $k+l = qn+r$, where q and r are quotient and reminder, respectively ($q \geq 0$, $n > r \geq 0$). The proof is divided into two cases, when $r=0$ and $r \neq 0$.

CASE I: $r = 0$

$$t_l(i) = \begin{cases} 2qn - n & i = 1 \\ 2qn & i \neq 1 \end{cases}$$

$$\begin{aligned} dt_l &= \frac{1}{n}\sum_{i=0}^{n} t_l(i) \\ &= \frac{1}{n}\big((2qn-n) + 2qn(n-1)\big) \\ &= 2qn - 1 \\ &= 2(k+l) - 1 \end{aligned}$$

CASE II: $r \neq 0$

$$t_l(i) = \begin{cases} 2qn + r & 1 \leq i \leq n-r+1 \\ 2qn+n+r & n-r+1 < i \leq n \end{cases}$$

$$\begin{aligned} dt_l &= \frac{1}{n}\sum_{i=0}^{n} t_l(i) \\ &= \frac{1}{n}\big((2qn+r)(n-r+1) + (2qn+n+r)(r-1)\big) \\ &= 2qn + 2r - 1 \\ &= 2(k+l) - 1 \end{aligned}$$

In both cases, $dt_l = 2(k+l)-1$ is independent of n. $dt_I(I)$ is equal to the expected value of normalized dt_l over all l's.

$$dt_I(I) = \sum_{l=0}^{\infty}\binom{k+l-1}{k-1}p^l(1-p)^k\frac{dt_l}{k} \tag{10}$$

$$= \sum_{l=0}^{\infty}\binom{k+l-1}{k-1}p^l(1-p)^k\frac{2(k+l)-1}{k} \tag{11}$$

Thus, we have shown that $dt_I(I)$ is independent of n. And we simplify $dt_I(I)$ as follows:

Let $m=0$ and $n=\infty$ in the following equality,

$$\sum_{i=m}^{n}\binom{i}{k}x^i = \frac{x^k}{k!}\frac{d^k}{dx^k}\Big(\frac{x^m - x^{n+1}}{1-x}\Big)$$

and with $0 \leq x < 1$, we can get

$$\begin{aligned} \sum_{i=0}^{\infty}\binom{i}{k}x^i &= \frac{x^k}{k!}\frac{d^k}{dx^k}\Big(\frac{1}{1-x}\Big) \\ &= \frac{x^k}{(1-x)^{k+1}} \end{aligned}$$

Since $\binom{i}{k} = 0$ when $i < k$, we can define $l = i - k$ and get

$$\sum_{l=0}^{\infty}\binom{k+l}{k}x^{k+l} = \frac{x^k}{(1-x)^{k+1}} \tag{12}$$

$$\Longrightarrow \sum_{l=0}^{\infty}\binom{k+l}{k}x^{l} = \frac{1}{(1-x)^{k+1}} \tag{13}$$

Take the derivative of both sides of (13), we can get

$$\sum_{l=0}^{\infty}l\binom{k+l}{k}x^{l-1} = \frac{k+1}{(1-x)^{k+2}} \tag{14}$$

$$\Longrightarrow \sum_{l=0}^{\infty}l\binom{k+l}{k}x^{l} = \frac{(k+1)x}{(1-x)^{k+2}} \tag{15}$$

By substituting (13) and (15), we can simplify (11) as

$$dt_I(I) \tag{16}$$

$$=(1-p)^k\Big[\frac{2k-1}{k}\times\frac{1}{(1-p)^k} + \frac{2}{k}\times\frac{kp}{(1-p)^{k+1}}\Big] \tag{17}$$

$$=\frac{2k-1}{k} + \frac{2p}{1-p} \tag{18}$$

$$=\frac{2}{1-p} - \frac{1}{k} \tag{19}$$

In summary, we have shown that $dt_I(I)$ is independent of n with the value as in (19). □

B. Simplification of $dt_I(1)$

$$dt_I(1) = \sum_{(j,m):M\geq k} \frac{N}{k}\binom{M-1}{k-1}p^{M-k}(1-p)^k$$

where $N = 2mn + j - (i-1)$, $M = mn + j - 1 + i$ and $i = 1$. Thus,

$$\begin{aligned} &dt_I(1) \\ &= \sum_{m=0}^{\infty}\sum_{j=1}^{n} \frac{2mn+j}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \\ &= \sum_{m=0}^{\infty}\sum_{j=1}^{n} \frac{2(mn+j)}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \\ &- \sum_{m=0}^{\infty}\sum_{j=1}^{n} \frac{j}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \end{aligned}$$

Let $l = mn + j$ and use (13) again, then the first item

$$\begin{aligned} &\sum_{m=0}^{\infty}\sum_{j=1}^{n} \frac{2(mn+j)}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \\ &= \sum_{l=0}^{\infty} 2\binom{l}{k}p^{l-k}(1-p)^k = \frac{2}{1-p} \end{aligned}$$

Let $q_1 = 0$ and $q_2 = \infty$ in the following equality:

$$\sum_{q=q_1}^{q_2}\binom{qn+a}{b}x^{qn+a} = \frac{x^b}{b!}\frac{d^b}{dx^b}\Big(\frac{x^{q_1 n+a} - x^{(q_2+1)n+a}}{1-x^n}\Big)$$

and with $0 \leq x < 1$, we can get

$$\sum_{q=0}^{\infty}\binom{qn+a}{b}x^{qn+a} = \frac{x^b}{b!}\frac{d^b}{dx^b}\Big(\frac{x^a}{1-x^n}\Big)$$

Using the above equality, the second item of $dt_I(1)$ becomes

$$\begin{aligned} &\sum_{m=0}^{\infty}\sum_{j=1}^{n} \frac{j}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \\ &= \sum_{j=1}^{n}\frac{j}{k}p^{1-k}(1-p)^k\sum_{m=0}^{\infty}\binom{mn+j-1}{k-1}p^{mn+j-1} \\ &= \sum_{j=1}^{n}\frac{j}{k!}(1-p)^k\frac{d^{k-1}}{dp^{k-1}}\Big(\frac{p^{j-1}}{1-p^n}\Big) \\ &= \frac{n}{k} + \frac{1}{1-p} - \frac{(1-p)^k}{k!}\frac{d^{k-1}}{dp^{k-1}}\Big(\frac{n}{(1-p)(1-p^n)}\Big) \end{aligned}$$

In summary,

$$\begin{aligned} dt_I(1) &= \frac{2}{1-p} - \sum_{m=0}^{\infty}\sum_{j=1}^{n}\frac{j}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \\ &= \frac{1}{1-p} - \frac{n}{k} + \frac{(1-p)^k}{k!}\frac{d^{k-1}}{dp^{k-1}}\Big(\frac{n}{(1-p)(1-p^n)}\Big) \end{aligned}$$

C. Proof of Theorem 3 and Derivation of $dt_I(1)$ ($n_I \neq n_{II}$)

Use similar notations as in the proof of Theorem 2, except that let $M = qn_I + r$ ($q \geq 0$, $0 < r \leq n_I$) now. Thus,

$$t_l(i) = \begin{cases} qn_I + qn_{II} + r & 1 \leq i \leq n_I - r + 1 \\ qn_I + qn_{II} + n_{II} + r & n_I - r + 1 < i \leq n_I \end{cases}$$

Then,

$$\begin{aligned} dt_l &= \frac{1}{n_I}\sum_{i=0}^{n_I} t_l(i) \\ &= \frac{1}{n_I}\big((qn_I + qn_{II} + r)(n_I - r + 1) \\ &+ (qn_I + qn_{II} + n_{II} + r)(r-1)\big) \\ &= (qn_I + r) + (qn_I + r - 1)\frac{n_{II}}{n_I} \\ &= (k + l) + (k + l - 1)\frac{n_{II}}{n_I} \end{aligned}$$

Define the ratio between the block length of item I and II as $\rho = n_{II}/n_I$. Then

$$dt_l = (k + l) + \rho(k + l - 1)$$

Substitute this into (10), we can simplify and get

$$dt_I(I) = \frac{1+\rho}{1-p} - \frac{\rho}{k}$$

It is easy to see that if $\rho = 1$ ($n_{II} = n_I$), the above equation reduces to (19).

In a similar way, we can simplify $dt_I(1)$ as

$$\begin{aligned} &dt_I(1) \\ &= \sum_{m=0}^{\infty}\sum_{j=1}^{n_I}\Big\{\frac{mn_I + mn_{II} + j}{k}\binom{mn_I + j - 1}{k-1} \\ &\times p^{mn_I + j - k}(1-p)^k\Big\} \\ &= \frac{1}{1-p} - \frac{\rho n_I}{k} + \frac{\rho(1-p)^k}{k!}\frac{d^{k-1}}{p^{k-1}}\Big(\frac{n_I}{(1-p)(1-p^{n_I})}\Big) \end{aligned}$$