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*— Virtually all known results of broadcast scheduling have assumed that channels are reliable without data corruption or loss. This assumption, however, is far from reality. In fact, data loss imposes severe impact on broadcast performance, as briefly shown in [9].**

In this paper, we study how to systematically derive optimal broadcast schedules for random-loss channels. The key idea is to employ proper MDS codes in the schedules. We show that the proposed scheme can achieve optimal performance, in terms of *expected delivery time*, and is much more robust to variations of channel loss probabilities, compared to those not using codes. In addition, we study the effect of basic schedule unit and conclude that the impact is prominent when data loss presents.

I. Introduction

Broadcast is an efficient way to disseminate popular information to a large group of users. It could be useful in applications, such as sending weather, news, stock, and lottery results to Smart Watches via wireless networks; updating car navigation systems via satellite networks; distributing critical software updates via institutional local area networks and so on. Assuming different information *items* targeting at different receivers are broadcast in the same channel, then a *schedule* needs to decide which item to broadcast and when. It is easy to see that a natural performance metric is the average time to deliver all items to their desired receivers. Finding *optimal schedules*, which minimize the average delivery time, has been drawing active research efforts for decades [1–10, 12, 13, 15–19].

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 [10, 15] 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. In real systems, however, items are almost always broadcast in the form of small unit (e.g. network packet). And it is obvious that the delivery time can be reduced with the help of receiver buffer. Indeed, Foltz and Bruck [7, 8] explored this direction and derived the optimal schedules, when items can be splitted into two halves. In this paper, we pursue this direction further and consider packet (much smaller compared to the item length) as the basic schedule unit in broadcast, which allows more flexibility and in turn yields more general results.

Even more importantly, most existing schedules, except only a few [9, 10, 15], assume broadcast channels are reliable and no data loss occurs during transmission. This assumption, however, is far from reality. In fact, data loss imposes severe impacts on broadcast performance, as briefly shown in [9]. Therefore, data loss needs to be considered seriously when designing broadcast schedules. In [10, 15], *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 data of corrupted blocks can still contribute to the recovery of original items and significantly reduce the delivery time when proper coding schemes are employed. To our best knowledge, [9] first introduced MDS codes in scheduling and demonstrated their 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 oversimplified. In this paper, we generalize this idea and carry out a systematic analysis for random-loss channels with any given loss probability, which is then used to derive optimal schedules.

Our contributions in this paper are four folds. First of all, we prove that optimal broadcast scheduling can be achieved using MDS codes. Secondly, we provide a systematic analysis of the scheduling performance using MDS codes. Thirdly, we show the robustness of the optimal schedule, whose performance degrades only slightly even when the channel deviates significantly from the targeted loss probability. Finally, using packet as the basic unit yields better schedules than those in the previous work, where items cannot be divided at all or merely into halves. We show that 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 the optimal schedule is derived for a simplified scenario and its robustness is demonstrated. Sec. III studies the effect of the basic schedule unit and the gain of splitting items is 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. As shown in [4, 5], optimal broadcast schedules are periodic for lossless channels. For simplicity, we also focus on *periodic* schedules for random-loss channels. A schedule can then be represented by a single period and denoted as $\mathrm{I}^i\mathrm{II}^j$, which means sending i packets of item 'I', followed by j packets of item 'II' in one period. Note that though we perform analysis on only two items in this paper, the approach discussed is applicable to scheduling of any number of items. Also, we limit our discussion on the simplest periodic form, although more complicated ones certainly exist.

For a given schedule, we use *Expected Delivery Time* (EDT) as the performance metric, which is defined to be the expected time for a receiver to get its desired item, averaged over all the receivers and all the items. The receivers are assumed to start receiving the items at times with *uniform* distribution. (Note that it is not difficult to generalize related results to non-uniform receiver arrival times.) The delivery time of an item then includes two parts, the *waiting time* and the *transmission time*. During the waiting time, a receiver waits in idle while items not interesting to it are being transmitted over the channel. And during the transmission time, the receiver effectively spends time receiving its desired item. Assume each item consists of k packets and has a demand probability associated with it, i.e., item I is demanded with probability p_I and item II with probability p_{II}, where $p_I + p_{II} = 1$. For all results, we normalize the item length, the channel bandwidth and thus the delivery time.

The following example explains the EDT calculation of a schedule of two items, when there is no data loss in the broadcast channel.

Example 1 *EDT of Schedule* I^kII^k

If a receiver wants item I, the normalized 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

Now we consider using MDS codes for broadcast channels with simple random data loss, where each data packet has an *identical* yet *independent* loss probability p.

An MDS (Maximum Distance Separable) [14] (n, k) error correcting code encodes k *message* packets to n *codeword* packets ($n \geq k$) and tolerates any $r = n - k$ packet losses during transmission. 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 item to a n-packet MDS code codeword. Then the codeword packets are sequentially used to replace the original item packets in schedule periods. In the event of data losses, any codeword packet can contribute to the recovery of the original item. When the codeword length n is sufficiently large so that any receiver can receive at least k different codeword packets in a single reception period, the desired original item can be easily recovered once k codeword packets are received.

Then, a natural question is whether using MDS codes is the best scheme for broadcast scheduling. We give an affirmative answer by the following theorem.

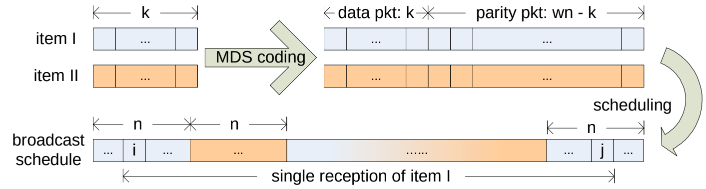


Fig. 1. Broadcast Scheduling Using MDS Codes

Theorem 1 *For any fixed schedule, broadcast using proper MDS codes requires the minimum delivery time.*

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 schedule is used. For the same loss pattern, if an proper MDS code is applied (as described in the following section), then $l = k$ and the delivery time must be no greater than d. Therefore, broadcast using proper 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 those using proper MDS codes (referred to as *CODING* hereafter). Second, broadcasting original item packets directly without using codes (*NO CODING* hereafter) can be regarded as a special case of using a (k, k) code, and is in general worse than schedules using MDS codes. However, it is still of interest to quantitatively compare the scheduling performance between using CODING and NO CODING to justify the computation overhead of MDS coding.

C. Optimal Broadcast Schedules Using CODING

We first focus on a simplified scenario by assuming the demand probabilities $p_I = p_{II} = 1/2$, where it is conceivable that the optimal schedule with the form I^iII^j satisfies $i = j$. A $(wn,\ k)$ MDS code is applied to each item such that wn codeword packets are generated from the original item of k 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 any single reception does *not* last more than wn packets. Then, an item can be successfully recovered if *any* k codeword packets of this item are received. This defines a schedule, based on the MDS code. The schedule has the form of I^nII^n and is determined completely by the broadcast *block* length n regardless of w. Figure 1 illustrates such a schedule.

A natural question then is: for a fixed item length k and given channel loss probability p, what is the *optimal value* of n 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 item I for our analysis.

As shown in Figure 1, a receiver completes single reception of item I from the i^{th} packet in one block to the j^{th} packet in another block. Let m be the number of item II blocks in between. Then, the total number of packets N (including both items) is

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

among them M are item I packets and

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

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

Among the total N packets, the loss patten of item II packets is irrelevant. And out of the M item I packets, exactly k are received while all the others are lost during the broadcast. Note that 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

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

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

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

Similarly, denote $dt_I(II)$ as the average delivery time of item I when the receiver starts within item II blocks. It is easy to see that the first partially received item II block is wasted (on average $\frac{n}{2}$ packets) and effective delivery time starts from the first packet in the following item I block, thus we have

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

Since the receiver has equal probability to start within item I or item II blocks,

$$EDT(I) = \frac{1}{2}\left(dt_I(I) + dt_I(II)\right) = \frac{1}{2}dt_I(I) + \frac{1}{2}dt_I(1) + \frac{n}{4k} \quad (2)$$

To compute $EDT(I)$ for a given loss probability p, we then need to calculate $dt_I(I)$ and $dt_I(1)$. Surprisingly, $dt_I(I)$ is independent of actual broadcast schedules, as shown in the following theorem.

Theorem 2 *$dt_I(I)$ is independent of n and satisfies (see the extended version [11] for proof):*

$$dt_I(I) = \frac{2}{1-p} - \frac{1}{k} \quad (3)$$

Although not intuitive, this property greatly simplifies the calculation of EDT.

Thus far, the only unknown term in EDT is $dt_I(1)$, which can be calculated as [11]:

$$dt_I(1) = \frac{2}{1-p} - \sum_{m=0}^{\infty}\sum_{j=1}^{n}\frac{j}{k}\binom{mn+j-1}{k-1}p^{mn+j-k}(1-p)^k \quad (4)$$

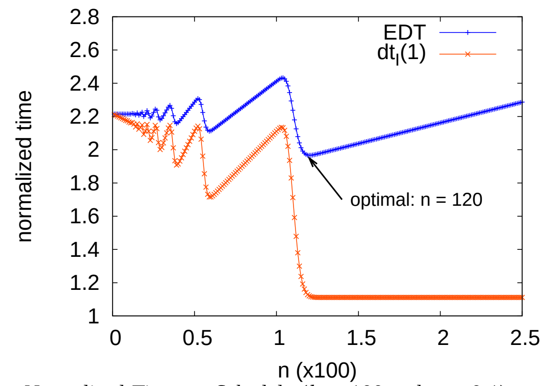


Fig. 2. Normalized Time vs. Schedule ($k = 100$ and $p = 0.1$)

Note that $dt_I(1)$ is lower bounded by $\frac{1}{1-p}$, which represents the EDT if only item I packets and no item II packets are broadcast in the channel.

Using (4), (3) and (2), it is easy to compute the EDTs for different n and find the optimal schedule. Figure 2 shows a numerical example with the item length $k = 100$ and the channel loss probability $p = 0.1$. The optimal schedule is achieved when $n = 120$. Note that as $\frac{n}{4k}$ grows linearly with n while $dt_I(1)$ is lower bounded by $\frac{1}{1-p}$, the numerical calculation of the EDT can terminate shortly after $n > \frac{k}{1-p}$.

D. Broadcast Schedules Using NO CODING

The analysis of the EDT using NO CODING is briefly presented here for the purpose of comparison with using CODING. The same notations for i, j, m, n, M and N apply. Also let $M = qk + r$ ($q \ge 0$, $0 < r \le n$), 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 the last received packet of item I is delivered ONLY at the final one of all $q + 1$ opportunities, which happens with the probability of $p^q(1-p)$. Then there are $r-1$ packets which have been delivered at least once among the $q+1$ opportunities, each with the probability of $1 - p^{q+1}$. Similarly, the rest $k - r$ packets have been delivered at least once among q opportunities, each with the probability of $1-p^q$. Thus the complete probability of this event is

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

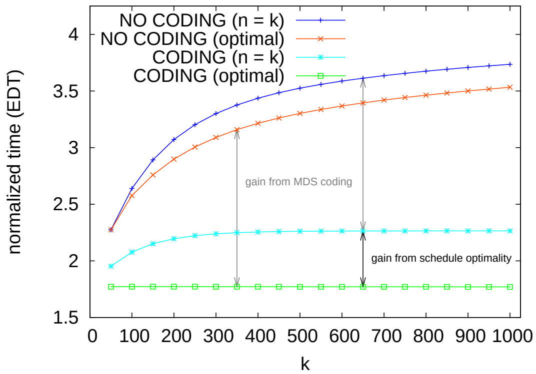
And $dt_I(i)$ can be calculated as

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

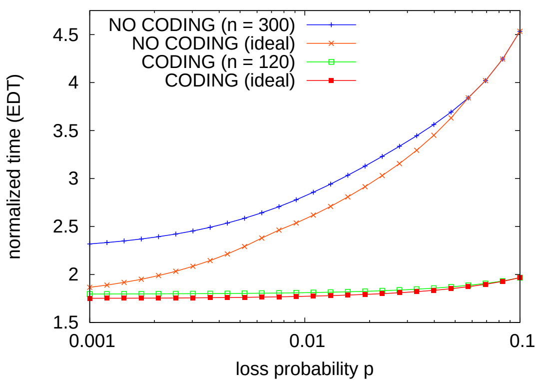
By computing the *overall* EDT with (1) and (2), the optimal schedule can then be obtained.

E. Optimality of Schedules Using CODING

Now we compare the scheduling performance of using CODING and NO CODING. Figure 3(a) shows the EDTs for various item length k with the channel loss probability $p = 0.01$. It is clear that the gain from using MDS codes is significant, compared to schedules without using any code. Also, the gain from schedule optimality is significant too, which



(a) Optimality of the CODING scheme ($p = 0.01$)



(b) Robustness of the Optimal Schedules ('Ideal' schedules are obtained assuming perfect knowledge of p.)

Fig. 3. Scheduling Performance

emphasizes the importance of choosing a proper n as the broadcast schedule.

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 for the worst channel loss probability and evaluate its performance with channels of different characteristics.

Figure 3(b) shows a numerical example, where the broadcast schedules are designed assuming the worst channel loss probability $p = 0.1$. Note that this loss rate is truly adversary and rarely happens in a normal wired or wireless network. Given the item length $k = 100$, the optimal schedules are $n = 120$ using CODING and $n = 300$ using NO CODING. Then both schedules are evaluated with channels of completely different loss probabilities (ranging from 0.001 to 0.1). Results, as in Figure 3(b), show that 1) the schedule performance using CODING remains stable under the wide range of channel characteristics, as opposed to the large variation using NO CODING; 2) the performance using CODING stays very close to the *ideal* optimal, where schedules can be tailored precisely with *a priori* knowledge of the channel loss probability; 3) the performance using NO CODING scheme degrades noticeably from the ideal optimal, when the perfect knowledge of the channel is not available during design phase. Therefore, using proper MDS codes makes practical to design schedules, that are targeted at the worst channel loss probability and yet achieve close to ideal optimal performance in real channels with varying loss probabilities.

III. Effect of Basic Schedule Unit

In the proposed scheme, items are divided into packets, which are then used as the basic scheduling units. This is more general than the previous work, such as [9] when items are non-separable (*no splitting*), and [7] when items can only be splitted into two halves (*half splitting*). We use (*arbitrary splitting*) to denote the proposed scheme, which obviously incorporates 'no splitting' and 'half splitting' as special cases. For example, the item consists of $k = 100$ packets. Then, 'no splitting' and 'half splitting' limit the schedule block length n to multiples of k (100, 200, 300, etc.) or $\frac{k}{2}$ (50, 100, 150, etc.), respectively. It is clear that these schedules are just special cases of 'arbitrary splitting', where n can take any value (113, 182, 231, etc.). Hence, all schemes can be analyzed in a unified framework, which then allows us to quantitatively study the effect of basic unit in scheduling.

Before proceeding with the analysis, we first generalize the results from the previous section by relaxing the equal demand assumption. Now the demand probabilities for both items are arbitrary and $p_I \neq p_{II}$ in general. Let n_I and n_{II} denote the schedule block length for items I and II, respectively. The item length remains as k, but now two different MDS codes (one (wn_I, k) code and one (wn_{II}, k) code) are applied (in general, $n_I \neq n_{II}$). The broadcast schedule is then $\mathrm{I}^{n_I}\mathrm{II}^{n_{II}}$. Defining the schedule block length ratio $\rho = n_{II}/n_I$, the following results can then be derived [11]:

$$dt_I(I) = \frac{1+\rho}{1-p} - \frac{\rho}{k} \tag{5}$$

$$dt_I(1) = \sum_{m=0}^{\infty}\sum_{j=1}^{n_I}\left\{\frac{mn_I + mn_{II} + j}{k}\binom{mn_I + j - 1}{k-1} \times p^{mn_I + j - k}(1-p)^k\right\} \tag{6}$$

$$EDT(I) = \frac{1}{1+\rho}dt_I(I) + \frac{\rho}{1+\rho}\left(dt_I(1) + \frac{n_{II}}{2k}\right) \tag{7}$$

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

These results can be reduced to simpler forms as in the previous section, when the schedule block lengths are equal ($\rho = 1$). Similarly to the equal demand case, given the item 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 also be obtained.

Now we study the effect of basic schedule unit by answering the questions whether splitting items helps in broadcast scheduling for lossless and random loss channels, respectively.

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

Lossless broadcast channels are the special case of random loss channels when the loss probability $p = 0$. And since there

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 : 150	1.7304	100 : 144	1.73021
0.35	100 : 100	1.745	100 : 100	1.745	100 : 113	1.74343
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 SCHEDULES FOR LOSSLESS CHANNELS ($k = 100$)

is no data loss at all, the delivery time is *exactly* the same for schedules using either CODING or NO CODING. Therefore, although they were derived using NO CODING for lossless channels, schedules in [7, 9] can be analyzed in the unified framework (5) - (8) derived using CODING for random loss channels.

Moreover, the results can be greatly simplified for lossless channels. From (5) and $p = 0$, we get

$$dt_I(I) = (1+\rho) - \frac{\rho}{k} \quad (9)$$

And $dt_I(1)$ can also be simplified, because the only term left in (6) requires the power of p to be 0. So $mn_I + j = k$. Thus when $n_I \geq k$, the only possible values are $m = 0$ and $j = k$. And for $n_I < k$, $m = q$ and $j = r$, if we define $k = qn_I + r$ with $q \leq 0$ and $0 < r \leq n_I$. To summarize, we have

$$dt_I(1) = \begin{cases} 1 & n_I \geq k \\ 1 + \rho(k-r) & n_I < k, k = qn_I + r \end{cases} \quad (10)$$

Then $EDT(I)$ and EDT can be computed with (7) and (8).

Table I shows the numerical comparisons of the optimal schedules and the corresponding EDT values using different basic units, with the demand probability for item I varying from 0.05 to 0.5 (or symmetrically from 0.5 to 0.95). It is verified that the optimal schedules for 'no splitting' and 'half splitting' match well with the results in [7, 9]. And 'arbitrary splitting' yields slightly better performance by using packet as the basic schedule unit. It is, however, worth pointing out that the gain is so little that it in fact suggests that even 'no splitting' can achieve close enough to the optimal performance. Nevertheless, this is no longer true in arbitrary random loss channels, as will be shown next.

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

The effect of basic schedule unit is examined again for the broadcast channel with the random loss probability $p = 0.01$. From the results shown in Figure 4, the gain of 'arbitrary splitting' is obvious over 'no splitting' or 'half splitting'. Hence, it is important to adopt more elaborate schedules than brute-force splitting items into halves or not splitting at all, when the broadcast channel is no longer error-free. Also notice that the scheduling performance using CODING in the lossy channel is very close to the lossless channel, which is consistent with the robustness analysis in the Section II-F.

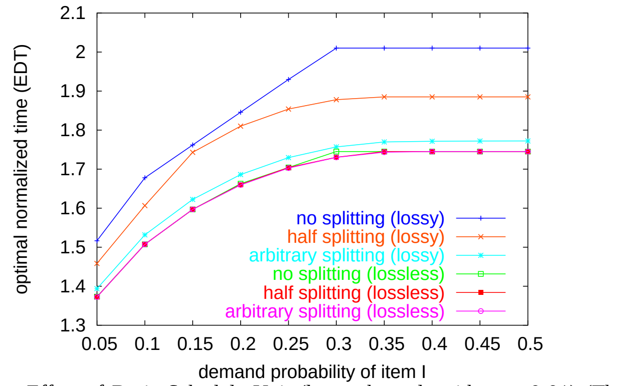


Fig. 4. Effect of Basic Schedule Unit (lossy channels with $p = 0.01$) (The three curves corresponding to lossless channels are tightly grouped together. See actual values in Table I.)

IV. CONCLUSIONS

In this paper, we propose to use proper MDS codes in broadcast scheduling to combat data losses. We systematically derive the optimal schedule for the random-loss channel. The performance gain using CODING and its robustness over wide range of channel loss probabilities are shown with numerical results. As a special case, we apply our analysis to the lossless channel, where results from the previous work are unified and generalized.

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 Effi cient Loss-Resilient Data Broadcasting," *Proc. of ISIT*, Yokohama, Japan, Jun. 2003.

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

[11] C. Huang, and L. Xu, "Optimal Broadcast Scheduling for Random-Loss Channels," *Washington University Tech. Report*, Apr. 2004. Available at http://www.nisl.wustl.edu/Papers/Tech/broadcast.pdf.

[12] 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.

[13] 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.

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

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

[16] C. J. Su, and L. Tassiulas, "Broadcast Scheduling for Information Distribution," *Proc. INFOCOM'97*, vol. 1, 109-117, 1997.

[17] 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.

[18] N. H. Vaidya, and S. Hameed, "Improved Algorithms for Scheduling Data Broadcast," *CS Tech. Report 96-029*, Texas A & M, Dec. 1996.

[19] 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.