



## A chapter preloading mechanism for e-reader in mobile environment

Chen-Yuan Chuang<sup>a,b,\*</sup>, Yi-Bing Lin<sup>b</sup>, Zhihao Julie Ren<sup>c</sup>

<sup>a</sup> Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taipei, Taiwan

<sup>b</sup> Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan

<sup>c</sup> Telecom & Management SudParis, Evry, France

### ARTICLE INFO

#### Article history:

Received 31 August 2011

Received in revised form 7 August 2012

Accepted 12 August 2012

Available online 20 August 2012

#### Keywords:

Buffer management

E-book

E-reader

Preload

Waiting time

Zipf's law

### ABSTRACT

E-book is a fast growing service for mobile Internet. A user reads e-books through an e-reader that could be a dedicated reading device or a mobile application on the smartphones or the tablets. With content protection of Digital Right Management (DRM), e-books take longer chapter loading time due to necessary data decryption. This paper propose a mechanism that preloads a small number of chapters into the buffer for DRM-protected e-books. When the user reads a chapter in the buffer, our solution reduces the user waiting time without affecting existing content protection. We also analyze the buffer replacement cost and effects of the limited buffer resource.

© 2012 Elsevier Inc. All rights reserved.

## 1. Introduction

Nowadays, *e-book* service has been deployed in many countries. A user reads e-books through an *e-reader* that could be a dedicated reading device or a mobile application on the smartphones or the tablets [2,3]. Most studies and surveys focused on e-reader features and functionalities [16,23]. In particular, *ePub* [11] has become a dominant standard of e-book format.

We observe that while the e-reader performs data loading, pagination, rendering, and page turning, the waiting time directly influences the user experience. As [20] demonstrated, the e-book chapter loading is a major procedure that consumes CPU time and memory resource, especially if the Digital Right Management (DRM) mechanism is involved. We have conducted general e-book user experience survey in the library of National Chiao-Tung University; the details are given in [19]. However, user experience of preloading to speed up encryption/decryption has not been investigated in the literature.

For open content, the e-reader can simply load the whole e-book from storage or file system [7]. On the contrary, DRM-protected e-books must be encrypted in order to avoid illegal copying [13,10,6]. To access DRM-protected e-book content, the e-reader requests a key and then spends additional CPU time and memory resource on data decryption when loading the required chapter into the buffer. Consequently, everytime the user turns to the next or jumps to another chapter, he/she must wait for a long response time until the new chapter is loaded.

Fig. 1 depicts the extracted files from a sample ePub. Rather than text-level compression [14], an ePub file basically is a zip-archived file, and the mimetype file (Line 1) indicates that the zip file follows the ePub-specific file structure. The ePub works with XML and Web technologies. XML files describe the metadata of the e-book, including book title, author, Table of Contents, and manifest file (Lines 3 and 7).

\* Corresponding author at: Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taipei, Taiwan.

E-mail addresses: [zzy@cht.com.tw](mailto:zzy@cht.com.tw) (C.-Y. Chuang), [liny@csie.nctu.edu.tw](mailto:liny@csie.nctu.edu.tw) (Y.-B. Lin), [juliae.ren@gmail.com](mailto:juliae.ren@gmail.com) (Z.J. Ren).

```

1  mimetype
2  META-INF/
3      container.xml
4      signatures.xml
5      encryption.xml
6  OEBPS/
7      content.opf
8      chap1.xhtml
9      chap1-pic1.png
10     chap2.xhtml
11     css/
12         style.css
    
```

Fig. 1. Files in a sample ePub.

In most of e-books, each chapter is encapsulated in an XHTML file (Lines 8 and 10). Like a Web page, the XHTML file might also reference some pictures and CSS files (Lines 9 and 12). File signatures and encryption (Lines 4–5) are optional and proprietary, but note that the DRM mechanism, e.g., ADEPT (Adobe Digital Experience Protection Technology) [1], only encrypts the chapter files rather than the whole ePub file.

Practically, the e-readers are built upon the web-browser components with the common APIs: `loadURL()` and `loadData()` [12,22,17]. The former loads the chapter contents from web sites or files, and the latter loads a single string as the chapter contents. Therefore, to feed the plaintext in secure paths (e.g., memory/buffer rather than files), the e-reader may host a local `ServerSocket` which decrypts the chapter files and mimics as a web site. Alternatively, the e-reader decrypts the chapter files and composes all materials into a string [15]. As a result, the DRM-protection induces extra user waiting time on chapter loading as compared with DRM-free scenario. In Chungghwa Telecom [8], we have conducted experiments which indicate that with a cache (buffer) mechanism, an e-reader with DRM-protection performs as fast as DRM-free one does.

This paper proposes a buffer management scheme that preloads a small number of chapters into the limited buffer. If the user intends to read a chapter in the buffer, this proposed mechanism reduces chapter loading time and thus improves the user experience.

## 2. Chapter preloading mechanism

The proposed chapter preloading mechanism uses a multi-element ring buffer [21], where each element holds one chapter of the e-book. Moreover, the e-reader loads and presents the current chapter in main thread and then preloads remaining chapters as background process. As long as the memory usage of the process is allowed on the mobile device, the e-reader can decrypt the ciphertext from files and preload the plaintext into buffer chapter by chapter. The concept of our preloading mechanism is similar to cache management described in [5,9], but the execution is very different.

Our mechanism and system model can apply to most e-readers that provide `Next` and `Previous` buttons, and support chapter jumping from the Table Of Contents (TOCs) of the e-book. Although some of them can further jump back to a recently read chapter, we will not cover the `Return/Back` and `Bookmark` in this paper. It is expected that a dedicated element to cache the last-read chapter may improve performance and chapters with bookmarks have higher probability to be jumped to as well, and our model can easily be extended to accommodate such a feature.

### 2.1. Notation and definitions

Let  $N$  be the number of chapters in a given e-book. We first rank all chapters according to its *Most Likely Reading Order* (MLRO) as shown in Fig. 2. MLRO is established based on common reading habits, i.e., the reading order is sequential and chapters close to the current reading one have higher probabilities to be read in the near future.

Let  $n$  be the current chapter ( $1 \leq n \leq N$ ), and  $k_n(i)$  be the MLRO rank of any Chapter  $i$  with respect to  $n$ . In Fig. 2, the gray-colored box indicates Chapter  $n$ , which has the rank  $k_n(n) = 1$ . Rank 2 is assigned to the next chapter in the sequential order:

Chapter number $i$	1	...	$n - j$	...	$n - 1$	$n$	$n + 1$	...	$n + j$	...	$N$
MLRO rank $k_n(i)$		...	$2j + 1$	...	3	1	2	...	$2j$	...	

Fig. 2. Most Likely Reading Order (MLRO).

$k_n(n + 1) = 2$ , rank 3 is assigned to the previous one, i.e.,  $k_n(n - 1) = 3$ , and so on. We note that user’s “backward” reading may not exactly follow the ranking. However, this ranking does capture that observed in our survey [19].

For the description purpose, we define two operators  $\oplus$  and  $\ominus$  as follows:

$$n \oplus j = (n + j) \bmod N \text{ and } n \ominus j = (n - j) \bmod N$$

where  $j$  is the jump distance (in number of chapters) forward or backward from Chapter  $n$ . In other words, Chapter 1 jumps forward 1 chapter to Chapter 2, and Chapter 2 jumps backward 2 chapters to Chapter  $N$ .

Because we transform the chapter sequence  $[1, N] = 1, 2, \dots, N$  into a ring structure, we extend the definition of chapter sequence as follows:

$$\langle a, b \rangle = \begin{cases} [a, b] & \text{if } a \leq b \\ [a, N] \cup [1, b] & \text{if } a > b \end{cases} \tag{1}$$

For example,  $\langle 2, 5 \rangle$  means Chapters 2–5, and  $\langle 5, 2 \rangle$  means Chapters 5– $N$  and Chapters 1–2.

Basically, the e-reader can jump from Chapter  $n$  to another Chapter  $n^*$  within  $N/2$  chapters, and we call it.

- a forward jump if  $n^* \in \langle n, n \oplus \lfloor \frac{N}{2} \rfloor \rangle$ , or
- a backward jump if  $n^* \in \langle n \ominus (\lfloor \frac{N}{2} \rfloor - 1), n \ominus 1 \rangle$ .

In Fig. 2, assume that  $N = 11$  and  $n = 6$ . If  $n^* = 9$ , then it is a forward jump with  $j = 3$ , and  $k_6(9) = 6$ . If  $n^* = 1$ , then it is a backward jump with  $j = 5$ , and  $k_6(1) = 11$ . It is clear that the MLRO ranking of Chapter  $n^*$  with respect to Chapter  $n$  can be expressed as

$$k_n(n^*) = \begin{cases} k_n(n \oplus j) = 2j & \text{for a forward jump} \\ k_n(n \ominus j) = 2j + 1 & \text{for a backward jump} \end{cases}$$

and the jump distance  $j$  will be

$$j = \left\lfloor \frac{k_n(n^*)}{2} \right\rfloor \tag{2}$$

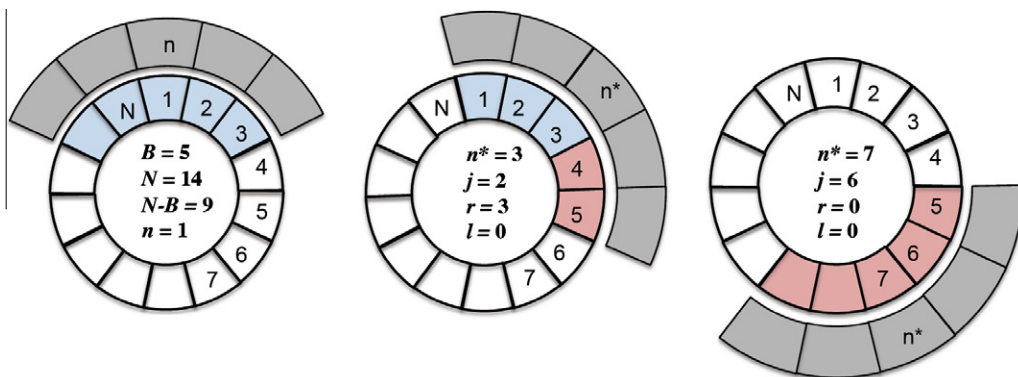
Let  $B$  be the buffer size, then Chapter  $i$  is in the buffer (i.e.,  $1 \leq k_n(i) \leq B$ ) when

$$i \in \begin{cases} \langle n \ominus (\frac{B-1}{2}), n \oplus (\frac{B-1}{2}) \rangle & \text{if } B \text{ is odd} \\ \langle n \ominus (\frac{B}{2} - 1), n \oplus (\frac{B}{2}) \rangle & \text{if } B \text{ is even} \end{cases} \tag{3}$$

Eq. (3) determines whether a chapter is in the buffer depending on  $B$ . For example, in Fig. 3a,  $n = 1$  and  $B = 5$ , then Chapters 13, 14, 1, 2 and 3 are in the buffer; in Fig. 4a,  $n = 1$  and  $B = 9$ , then Chapters 11–5 are in the buffer.

### 2.2. Chapter replacement cost

We assume that the limited buffer size cannot preload the whole e-book ( $N > B$ ), so that the e-reader must replace some *victim* chapters in the buffer. In Fig. 3, when the user jumps to Chapter 3 (see Fig. 3b), a.k.a. *cache hit*, there are 2 chapters in the buffer will be replaced; when the user jumps to Chapter 7 from Chapter 1 (see Fig. 3c), a.k.a. *cache miss*, all chapters in the buffer will be replaced in this case.



(a). Initial state with a small  $B$  (b). One-side overlapping from (a) (c). No overlapping from (a)

Fig. 3. Buffer replacement mechanism (small  $B$ ).

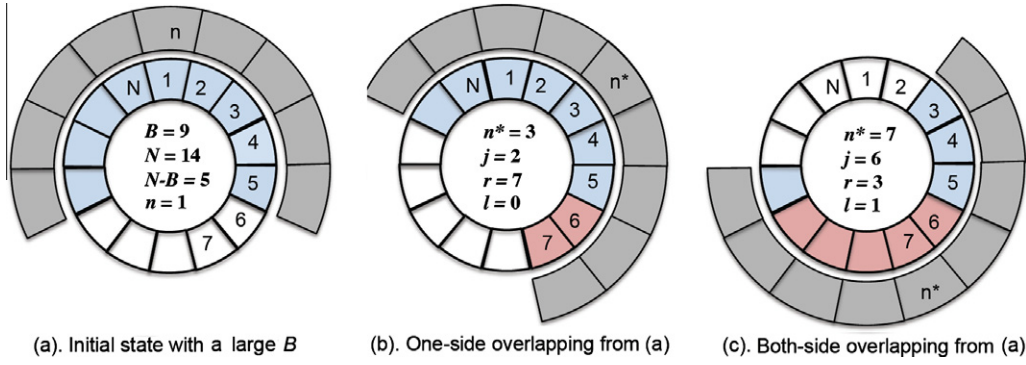


Fig. 4. Buffer replacement mechanism (large B).

After a chapter jump, the MLRO ranking is reset with respect to the new current chapter. If the MLRO ranking of a buffered chapter is larger than  $B$ , it implies that the chapter should be replaced; otherwise, the chapter stays in the buffer.

The number of chapters to be replaced is derived in the following Theorem, and scenarios of forward jumps are illustrated in Figs. 3 and 4.

**Theorem 1.** Let  $N_L(j)$  be the number of replaced chapters. Then

$$N_L(j) = \begin{cases} j & \text{if } j < B \text{ and } j < N - B & \text{(a)} \\ N - B & \text{if } N - B \leq j < B & \text{(b)} \\ B & \text{if } B \leq j \leq N - B & \text{(c)} \end{cases} \quad (4)$$

**Proof 1.** Suppose that  $B$  is odd (we omit the similar proof for the case when  $B$  is even). From (3), the chapter sequence  $S$  in the buffer (see Figs. 3a and 4a) is

$$S = \left\langle n \ominus \left( \frac{B-1}{2} \right), n \oplus \left( \frac{B-1}{2} \right) \right\rangle \quad (5)$$

Without loss of generality, we assume that the user jumps  $j$  chapters forward (we omit the similar proof for a backward jump). After the jump, the chapter sequence  $S^*$  in the buffer is

$$S^* = \left\langle n^* \ominus \left( \frac{B-1}{2} \right), n^* \oplus \left( \frac{B-1}{2} \right) \right\rangle = \left\langle (n+j) \ominus \left( \frac{B-1}{2} \right), (n+j) \oplus \left( \frac{B-1}{2} \right) \right\rangle \quad (6)$$

Let  $r$  and  $l$  be the numbers of overlapping chapters on the right-hand side and the left-hand side in  $S$  and  $S^*$ , respectively. For example, in Fig. 4c, we have  $r = 3$  (Chapters 3, 4 and 5) and  $l = 1$  (Chapter 11). The number of overlapping chapters is  $r + l$ . Therefore, the number of replaced chapters in the buffer is

$$N_L(j) = B - (r + l) \quad (7)$$

Now, we prove the hypothesis in two scenarios:  $j < B$  and  $j \geq B$ .

*Scenario 1:  $j < B$*  (see Figs. 3b and 4b and c) We first consider the overlapping chapters on the right-hand side. Since  $j < B$ , from (5) and (6), we know that the overlapping chapters on the right-hand side must be in

$$\left\langle (n+j) \ominus \left( \frac{B-1}{2} \right), n \oplus \left( \frac{B-1}{2} \right) \right\rangle \quad (8)$$

From (8),

$$r = \left[ n + \left( \frac{B-1}{2} \right) \right] - \left[ (n+j) - \left( \frac{B-1}{2} \right) \right] + 1 = B - j \quad (9)$$

Now, we consider the left-hand side: If  $l \neq 0$ , (see Fig. 4c), it implies that the chapters not in  $S$  must be loaded in  $S^*$ , and

$$N_L(j) = N - B \quad (10)$$

Similarly to (8), we have the overlapping chapters on the left-hand side from (5) and (6) if  $l \neq 0$ . Then we find that

$$(n+j) + \left( \frac{B-1}{2} \right) \geq n - \left( \frac{B-1}{2} \right) + N \iff j \geq N - B \quad (11)$$

Therefore,  $l \neq 0$  if and only if  $j \geq N - B$ . When  $j < N - B$  (see Figs. 3b and 4b),  $l = 0$  and from (7) and (9),  $N_L(j)$  is

$$N_L(j) = B - (r + l) = B - ((B - j) + 0) = j \quad (12)$$

From (10) and (11), we prove Eq. (4)b, and Eq. (4)a is also proved from (12).

*Scenario 2:*  $j \geq B$  (see Fig. 3c) This scenario implies  $r = 0$ . We show  $l = 0$  by contradiction. Suppose  $l \neq 0$ , then (11) must be satisfied. When  $j > N - B$ , by definition of jump,  $j \leq N/2$ , we have

$$\frac{N}{2} > N - B \iff B > \frac{N}{2} \quad (13)$$

Since  $j \geq B$ , Eq. (13) implies that  $j > N/2$  which contradicts with the definition of jump, i.e.,  $j \leq N/2$ . When  $j = N - B$ ,  $B$  and  $j$  will be  $N/2$  and finally  $l$  must be 0. Therefore, Scenario 2 holds if and only if  $B \leq j \leq N - B$ ,

$$N_L(j) = B \quad (14)$$

From (14), Eq. (4)c is proved.  $\square$

### 3. Analysis and discussion

This section shows how to select appropriate  $B$  values according to the *reload probability*  $p$  and the *replacement cost*  $C_L$ . Let  $p$  be the probability that the next Chapter  $n^*$  is not in the buffer (cache miss). Let  $C_L$  be the expected number of replaced chapters in the buffer due to a chapter jump. Note that for DRM-protected e-books, the buffer of e-reader is a limited resource, and for the practical exercise, it suffices to consider  $B < N/2$ .

We derive  $p$  and  $C_L$  assuming that the probability distribution of e-book chapter reading follows Zipf's law [4,18]. This law predicts the frequency of the chapter of MLRO rank  $k$  to appear. The Zipf's probability of the chapter of rank  $k$  is

$$P_Z(k) = \left[ k^s \left( \sum_{i=1}^N \frac{1}{i^s} \right) \right]^{-1} \quad (15)$$

where  $s$  describes the reading behavior, and more specifically, the locality of the read chapters. A large  $s$  means that the user is inclined to read neighboring chapters; on the contrary, a small  $s$  means that there is a higher probability to jump further away from the current chapter. Zipf's law gives the probability of any chapter to be read depending on its MLRO rank.

From (15), the reload probability  $p$  is expressed as

$$p = \sum_{k_n(n^*) > B} P_Z[k_n(n^*)] = 1 - \sum_{1 \leq k \leq B} \left[ k^s \left( \sum_{i=1}^N \frac{1}{i^s} \right) \right]^{-1} \quad (16)$$

Now we derive  $C_L$ . For the practical exercise, we assume  $B < N/2$ , so Theorem 1 can be simplified as

$$N_L(j) = \begin{cases} j & \text{if } j < B \\ B & \text{if } B \leq j \end{cases}$$

Let  $C_1$  be the replacement cost corresponding to forward jumps. Since  $k_n(n^*) = 2j$  and  $j \leq \lfloor N/2 \rfloor$ , we have

$$C_1 = \sum_{1 \leq j \leq \lfloor \frac{N}{2} \rfloor} N_L(j) P_Z(2j) = \sum_{1 \leq j < B} j P_Z(2j) + B \left( \sum_{B \leq j \leq \lfloor \frac{N}{2} \rfloor} P_Z(2j) \right) \quad (17)$$

Let  $C_2$  be the replacement cost corresponding to backward jumps. Since  $k_n(n^*) = 2j + 1$  and  $j \leq \lfloor N/2 \rfloor - 1$ , we have

$$C_2 = \sum_{1 \leq j \leq \lfloor \frac{N}{2} \rfloor - 1} N_L(j) P_Z(2j + 1) = \sum_{1 \leq j < B} j P_Z(2j + 1) + B \left( \sum_{B \leq j \leq \lfloor \frac{N}{2} \rfloor - 1} P_Z(2j + 1) \right) \quad (18)$$

From 17, 18, 15 and 2, the replacement cost  $C_L$  is

$$C_L = C_1 + C_2 = \sum_{1 \leq l < 2B} \left\lfloor \frac{l}{2} \right\rfloor P_Z(l) + B \left( \sum_{2B \leq l \leq N} P_Z(l) \right) = \sum_{1 \leq l < 2B} \left\lfloor \frac{l}{2} \right\rfloor \left[ l^s \left( \sum_{i=1}^N \frac{1}{i^s} \right) \right]^{-1} + B \left( \sum_{2B \leq l \leq N} \left[ l^s \left( \sum_{i=1}^N \frac{1}{i^s} \right) \right]^{-1} \right) \quad (19)$$

From (16), we observe that a larger  $B$  implies a lower  $p$ . On the contrary, from (19), a larger  $B$  implies a higher  $C_L$ . To select an appropriate  $B$  to balance against  $p$  and  $C_L$ , we define a net cost  $C$  as follows:

$$C = p + \alpha C_L$$

where  $\alpha$  is a factor that weights the importance of  $C_L$  against  $p$ .

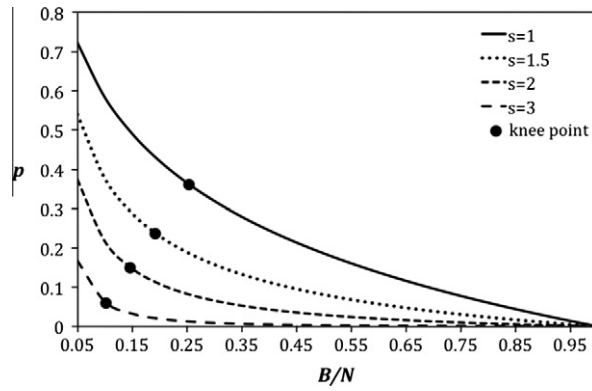


Fig. 5. Effect of  $s$  and  $B/N$  on  $p$  ( $N = 20$ ).

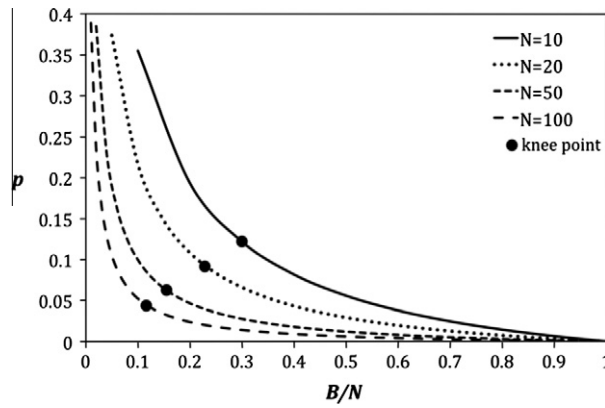


Fig. 6. Effect of  $N$  and  $B/N$  on  $p$  ( $s = 2$ ).

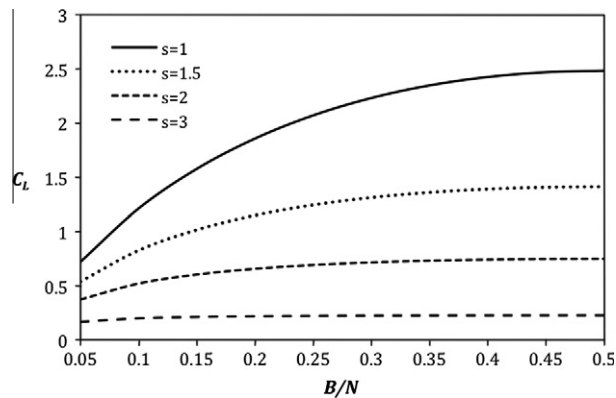


Fig. 7. Effect of  $s$  and  $B/N$  on  $C_L$  ( $N = 20$ ).

Fig. 5 shows the reload probability  $p$  as a function of  $B/N$  and  $s$ , where  $N = 20$ . The figure indicates the intuition that  $p$  decreases as  $B/N$  increases. We observe a non-trivial result that there is a knee point (the bullets in Fig. 5) such that before this point,  $p$  significantly decreases as  $B/N$  increases. This phenomenon implies that after this knee point, selecting a larger buffer size does not improve  $p$  performance. Such knee effect is more significant for a larger  $s$ . Fig. 5 also indicates that to achieve the same  $p$  performance,  $B/N$  decreases as  $s$  increases.

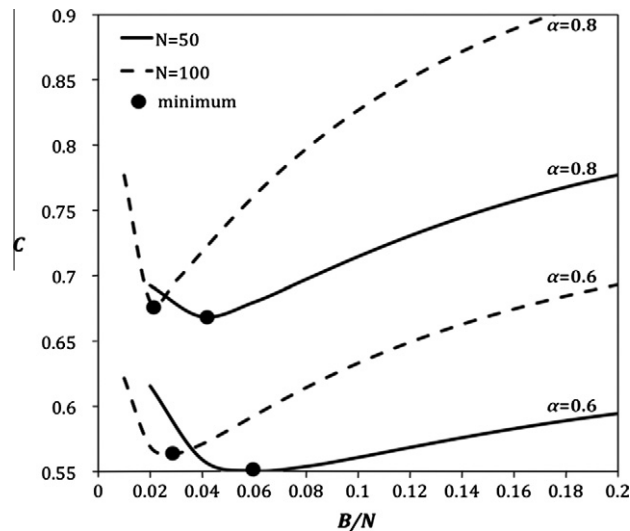


Fig. 8. Effect of  $N$ ,  $\alpha$  and  $B/N$  on  $C$  ( $s = 2$ ).

Fig. 6 illustrates that to achieve the same  $p$  performance; a larger  $B/N$  is required for a smaller  $N$ . That is, a larger portion of e-book needs to be stored in the buffer for an e-book of small volume. Similar to Fig. 5, a knee point can be determined on every  $p$  curve.

Fig. 7 shows the loading cost for  $N = 20$  under various  $s$  values. We observe that  $C_L$  increases as the locality parameter  $s$  or the e-book size  $B/N$  increases; this effect becomes less significant with large  $s$  values.

The net cost for  $s = 2$  is illustrated in Fig. 8. For  $\alpha = 0.6$ , we observe a minimum value on each  $C$  curve for  $N = 50$  and  $N = 100$ , respectively. In both cases, this minimum occurs when  $B = 3$  ( $B/N = 0.06$  for  $N = 50$  and  $B/N = 0.03$  for  $N = 100$ ). By increasing the factor  $\alpha$ , the protection of the e-book is higher and the minimum occurs for a smaller  $B$  value (for  $\alpha = 0.8$ ,  $B = 2$ ). We note that the  $B/N$  value of the minimum is always smaller than that of the knee point for  $p$ .

The above performance study indicates that by slightly increasing the buffer size, the user experience is significantly improved. For example in Fig. 8, by buffering two more chapters (i.e.,  $B/N$  increases from 0.02 to 0.06 for  $N = 50$  and  $\alpha = 0.6$ ), we can reduce the reload probability  $p$  from 0.38 to 0.17 (Fig. 6), which is a 56% improvement. Our chapter preloading mechanism provides a method for e-reader designers to select the appropriate buffer size according to the requirements in term of user experience and e-book protection.

#### 4. Conclusions

This paper proposed a chapter preloading mechanism for e-reader that significantly enhances the user experience by slightly increasing the buffer resource with the lower reload probability. Furthermore, we also guided the e-reader developers to allocate the appropriate buffer resource to balance the cache-hit ratio and replacement cost.

Our scheme preloads more chapters in the buffer than the existing scenario (i.e., no preload). Since in both approaches, the decrypted chapters are loaded into the secure paths that are not accessible by other processes, our scheme is compatible with the existing DRM-protection and assures the required protection level for the e-book.

The forthcoming ePub 3.0 will enrich the e-book contents, and different reading sequences are expected for various contents. A potential future work is to extend the MLRO ranking to be more precise for various classifications of e-book and the features of e-readers. Same technique proposed in this paper can be used to model the access performance of the new reading features.

#### References

- [1] Adobe-Reader Developer Center—Adobe Developer Connection. <<http://www.adobe.com/devnet/reader.html>> (accessed 25.03.12).
- [2] Amazon.com: Kindle eBooks, Newspapers, Magazines, Blogs. <<http://www.amazon.com/>> (accessed 25.03.12).
- [3] App Store-Stanza. <<http://itunes.apple.com/us/app/stanza/id284956128?mt=8>> (accessed 25.03.12).
- [4] L. Breslau, C. Pei, F. Li, G. Phillips, S. Shenker, Web caching and Zipf-like distributions: evidence and implications, in: Proceedings of IEEE INFOCOM'99, March 1999, pp. 126–134.
- [5] F.-C. Chang, H.-C. Huang, A refactoring method for cache-efficient swarm intelligence algorithms, Information Sciences 192 (2012) 39–49.
- [6] C.-Y. Chuang, Y.-C. Wang, Y.-B. Lin, Digital right management and software protection on Android phones, in: Proceedings of the 71st Vehicular Technology Conference (VTC 2010-Spring), 2010.
- [7] Freda: the free ebook reader. <<http://www.turnip.demon.co.uk/jim/freda/>> (accessed 25.03.12).
- [8] HamiBook (in Chinese) — Chunghwa Telecom Co., Ltd. <<http://bookstore.emome.net/>> (accessed 25.03.12).

- [9] W.-S. Han, W.-K. Loh, K.-Y. Whang, Type-level access pattern view: enhancing prefetching performance using the iterative and recursive patterns, *Information Sciences* 180 (2010) 4118–4135.
- [10] G. He, Analysis of e-book security, Publications in Telecommunications Software and Multimedia, 2001. <<http://www.tml.tkk.fi/Studies/Tik-110.501/2001/papers/he.pdf>> (accessed 25.03.12).
- [11] International Digital Publishing Forum – Trade and Standards Organization for the Digital Publishing Industry. <<http://idpf.org/>> (accessed 25.03.12).
- [12] iOS Developer Library: UIWebView Class Reference. <[http://developer.apple.com/library/ios/#documentation/uikit/reference/UIWebView\\_Class/Reference/Reference.html](http://developer.apple.com/library/ios/#documentation/uikit/reference/UIWebView_Class/Reference/Reference.html)> (accessed 25.03.12).
- [13] Q. Liu, R. Safavi-Naini, N. Sheppaer, Digital rights management for content distribution, in: Proceedings of the Australasian Information Security Workshop, 2003.
- [14] M.A. Martinez-Prieto, J. Adiego, P. Fuente, Natural language compression on edge-guided text preprocessing, *Information Sciences* 181 (2011) 5387–5411.
- [15] L. Masinter, The “data” URL Scheme, RFC 2397, IETF, August 1998.
- [16] L.M. Moore, At Your Leisure: Assessing Ebook Reader Functionality and Interactivity, Master Thesis, University College, London, UK, 2009.
- [17] MSDN Library: WebBrowser Class (Microsoft.Phone.Controls). <[http://msdn.microsoft.com/en-us/library/ff402601\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff402601(v=vs.92).aspx)> 25.03.12).
- [18] M.E.J. Newman, Power laws, Pareto distributions and Zipf’s law, *Contemporary Physics* 46 (2005) 323–351.
- [19] Z.J. Ren, Performance Evaluation and User Experience of E-book Reader, Master Thesis, National Chiao-Tung University, Taiwan, 2011.
- [20] Z.J. Ren, Y.-B. Lin, N. Crespi, Evaluation of e-book reader on mobile devices, in: Proceedings of the 15th Mobile Computing Workshop, Taichung, Taiwan, 2010.
- [21] P.K.C. Tse, Multimedia Information Storage and Retrieval: Techniques and Technologies, IGI Global, 2008.
- [22] WebView – Android Developers. <<http://developer.android.com/reference/android/webkit/WebView.html>> (accessed 25.03.12).
- [23] R. Wilson, E-book readers in higher education, *Journal of Educational Technology Society* 6 (2003) 8–17.