

A Content-Centric Framework for Effective Data Dissemination in Opportunistic Networks

Ling-Jyh Chen, *Member, IEEE*, Chen-Hung Yu, Cheng-Long Tseng,
Hao-hua Chu, *Member, IEEE*, Cheng-Fu Chou

Abstract—In this paper, we address the challenges of content transfer in opportunistic networks, and propose techniques to better facilitate data dissemination based on the characteristics of the content. To investigate this problem from its origins, we propose three message scheduling algorithms: Sequential Forwarding (SF), Full Interleaving (FI), and Block-based Interleaving (BI). Each algorithm is embedded in a specially tailored data dissemination technique to evaluate the benefits of applying it to different types of content and data dissemination methods. Three types of content (file, video and web) are considered and evaluated, and the dissemination methods considered are Layered Multiple Description Coding (LMDC) based and file-based. Using simulations as well as both synthetic and realistic network scenarios, we evaluate the proposed schemes in terms of latency and user perceived quality, and demonstrate how the schemes can achieve much better latency performance for file transfers. Furthermore, we show that using LMDC-based techniques leads to higher user perceived quality, since the end user is allowed to “preview” video file or web content, even before the data has been completely transferred. The effectiveness and robustness of our message scheduling algorithms and their corresponding content dissemination techniques make them ideal solutions that can go a long way toward effective data dissemination in opportunistic networks.

I. INTRODUCTION

With wireless networking technologies extending into every part of our working and living environments, proper handling of intermittent wireless connectivity and network disruptions is important. As the foreseeable need for data communication escalates in challenged network environments, an increasing amount of effort is being invested in developing techniques that can address these anticipated requirements. Applications of these techniques are wide ranging. For instance, it would be quite advantageous to be able to interconnect mobile search and rescue nodes in disaster areas (where communication infrastructures have been disabled by earthquakes, hurricanes, wildfires, or flooding), allow message exchanges in underdeveloped areas (remote towns and villages interconnected by wireless networks, but not guaranteed an always-on Internet connection), and permit scientific monitoring of wilderness areas (remote monitoring of various forms of wildlife).

Formally, an opportunistic network is a type of challenged network that satisfies the following conditions: (1) network contacts (i.e., communication opportunities) are intermittent,

(2) an end-to-end path between the source and the destination rarely exists, (3) disconnection and reconnection is common, and (4) link performance is highly variable or extreme. Because of various disruptions and long delays, traditional MANET and Internet routing techniques can not be applied directly to opportunistic networks. Meanwhile, with the development of numerous opportunistic networking applications, such as wireless sensor networks (WSN) [6][39], underwater sensor networks (UWSN) [15], pocket switched networks (PSN) [10][21], people networks [35][37], and transportation networks [3][8], it is both desirable and necessary to develop an effective content dissemination framework that can better accommodate data content and the various characteristics of opportunistic networks.

Several data forwarding schemes have been proposed for opportunistic networks [8][19][23][24][27][36][38][41]. Such schemes can be divided into two main categories, *replication-based* and *coding-based*, according to their basic technical strategies. In general, coding-based schemes are more robust than replication-based schemes when the network connectivity is extremely poor, but they result in inefficiency when the network is fairly connected (due to the additional code blocks embedded in the data). Yet, all of the above schemes only consider data delivery performance (in terms of delivery latency and success ratio), and neglect the innate characteristics (e.g., content types) of applications when designing dissemination mechanisms. With the growing diversity of opportunistic network applications, a well-designed and content-centric solution for effective data dissemination remains highly desirable.

In this study, we propose techniques to better facilitate content dissemination based on the characteristics of the content. To investigate this problem from its origins, we propose three message scheduling algorithms: HEC-FI, HEC-SF, and HEC-BI. Each algorithm is embedded in a specially tailored content dissemination technique to evaluate the benefits of applying it to different types of content and content dissemination methods. Three types of content (file, video and web) are considered and evaluated, and the dissemination methods considered are Layered MDC-based and file-based. Using simulations as well as both synthetic and realistic network scenarios, we evaluate the proposed schemes in terms of latency and user perceived quality. We demonstrate how the schemes can achieve much better latency performance for file transfers. Specifically, the results show that HEC-BI and HEC-SF achieve a good performance for networks with good connectivity, while HEC-FI yields a more resilient performance in cases of poor network connectivity. Furthermore, we show

Manuscript received XXX XX, 2007.

L.-J. Chen is with the Institute of Information Science, Academia Sinica.

C.-H. Yu, C.-L. Tseng, H.-h. Chu, and C.-F. Chou are with the Department of Computer Science and Information Engineering, National Taiwan University.

that our proposed LMDC-based schemes enable the end user to “preview” video or web content at a lower quality, even before the data has been completely transferred, thereby improving the overall viewing experience. Our evaluations provide a deeper understanding of how to take advantage of content types and message scheduling to more effectively disseminate data across opportunistic networks¹.

The remainder of the paper is organized as follows. Section III provides a comprehensive overview of the HEC² routing scheme and describes the proposed enhancements in detail. In Section IV, a Layered MDC scheme with an unequal erasure protection scheme is presented, and the objective of the scheme is to facilitate more effective data transfer for video and web content delivery over opportunistic networks. Section V presents a comprehensive set of simulation results for various opportunistic network scenarios; the results are also analyzed and explained in detail. In Section II, we review related works of data forwarding schemes for opportunistic networks. We then present our conclusions in Section VI.

II. RELATED WORK

Routing in an opportunistic network is challenging and completely different from routing in a conventional network. An ideal routing scheme for opportunistic networks has to provide reliable data delivery, even when the network connectivity is intermittent or when an end-to-end path is temporally unavailable. Moreover, since ‘contacts’ in an opportunistic network may appear arbitrarily without prior information, neither scheduled optimal routing (e.g., linear programming routing in delay tolerant networks of scheduled contacts [22]) nor mobile relay approaches (e.g., Message Ferrying [42][43]) can be directly applied.

Currently, replication is the most popular design choice for opportunistic routing schemes. For instance, the *Epidemic Routing* scheme [36] sends identical copies of a message simultaneously over multiple paths to mitigate the effects of a single path failure; thus, it increases the possibility of successful message delivery. However, flooding a network with duplicate data tends to be very costly in terms of traffic overhead and energy consumption.

To address the problem of excess traffic overhead caused by flooding, Harras et al. proposed a *Controlled Flooding* scheme to reduce the flooding cost while maintaining reliable message delivery [19]. In this scheme, flooding is controlled by three parameters, namely, *willingness probability*, *Time-to-Live*, and *Kill Time*. Additionally, once a message has been delivered to the receiver successfully, a *Passive Cure* is generated to “heal” the nodes in the network that have been “infected” by the message. Therefore, by removing the excess traffic overhead problem, while providing reliable data delivery, controlled flooding can substantially reduce the network overhead.

Node mobility also impacts on the effectiveness of opportunistic routing schemes. Previous studies have shown that if

the network mobility departs from the well-known random way-point mobility model (e.g., the Pursue Mobility Model [9] or the Reference Point Group Mobility Model [20]), the overhead carried by epidemic- and/or flooding-based routing schemes can be further reduced by considering node mobility. For instance, the *Probabilistic Routing* scheme [26] calculates the *delivery predictability* from a node to a particular destination node based on the observed contact history, and forwards a message to its neighboring node if and only if that neighboring node has a higher delivery predictability value. The scheme was revised by Leguay *et al.* [23] by considering the *mobility pattern*, i.e., a message is forwarded to a neighbor node if and only if that node has a mobility pattern more similar to the destination node. [23][24] show that the revised *mobility pattern* scheme is more effective than previous schemes.

Another class of opportunistic network routing schemes is based on encoding techniques, which transform a message into a different format prior to transmission. For instance, an integration of *network coding* and epidemic routing techniques has been proposed to reduce the required number of transmissions in a network [41], and [38] proposes combining *erasure coding* and the simple replication-based routing method to improve data delivery for the *worst delay performance* cases in opportunistic networks.

Following the concept of erasure coding-based data forwarding [38], an Estimation based Erasure-Coding routing scheme (EBEC) has been proposed to adapt the delivery of erasure coded blocks using the Average Contact Frequency (ACF) estimate [25]. Moreover, [12] proposes a hybrid scheme that combines the strength of erasure coding and the advantages of *Aggressive Forwarding*, so that it not only remains robust in *worst delay performance cases*, but also performs efficiently in *very small delay performance cases*.

III. HEC: AN ERASURE CODING-BASED HYBRID ROUTING APPROACH

A. HEC Overview

Erasure coding is a scheme that provides better fault-tolerance by adding redundancy without the overhead of strict replication of the original data [40]. Two of the most popular erasure coding algorithms are Reed-Solomon coding and Low-Density Parity-Check (LDPC) based coding (e.g., Gallager codes, Tornado codes, and IRA codes) [28][34]. These algorithms differ in the degree of encoding/decoding efficiency, the replication factor, and the minimum number of code blocks needed to reconstruct a message. The selection of an appropriate erasure coding algorithm is not within the scope of this paper, since our work is based on the concept of generic erasure coding.

In a generic erasure coding scheme, given a message of size S bytes, a replication factor of erasure coding r , and a coded message fragmented into several blocks of identical size b bytes, one can obtain the number of coded blocks by $N = \frac{S \times r}{b}$. Moreover, this message can be successfully reconstructed as long as $\frac{1}{r}$ of the coded blocks are received, i.e., the minimal number of coded blocks required to successfully reconstruct the message is N/r .

¹Note that, unlike previous studies on opportunistic routing, we focus on data transfer scenarios that assume the source has a complete set of messages before initiating data forwarding (rather than generating messages on the fly, as in [12][38]).

²The HEC scheme was called H-EC in previous work [12].

In [38], an Erasure Code-based forwarding algorithm (EC) is proposed. In this scheme, the erasure coded blocks are split equally among n relays, which are only allowed to send messages to the destination directly (this is the well-known “two-hop” scenario used in [18]). Each relay forwards the same number of coded blocks, and the number of blocks forwarded by each relay can be obtained by³

$$\frac{N}{n} = \frac{Sr}{bn}. \quad (1)$$

As reported in [38], the EC scheme can provide the best *worst-case delay performance* with a fixed amount of overhead. However, the drawback of the EC scheme is that it can not provide good *very small delay performance* compared to other popular replication-based approaches. The reason for this inefficiency lies in its block allocation method. In the EC scheme, the number of transmitting blocks in each contact is fixed (i.e., $\frac{Sr}{bn}$, in accordance with Eq. 1) regardless of the duration of each contact. As a result, the EC scheme can only utilize each network contact effectively when the contact duration is slightly longer than the time required to send the relayed data. If most network contacts are much longer than the required time, the EC scheme tends to *waste* the residual contact period, which results in ineffectiveness.

To resolve this problem, [12] proposed an enhanced scheme called A-EC, i.e., EC with an *aggressive forwarding* feature. In this scheme, the source sends as many coded blocks as possible during each contact (totally $\frac{Sr}{bn}$ blocks, i.e., $\frac{Sr}{n}$ bytes). It has been shown that the A-EC scheme is better able to utilize network contacts; thus, it can be expected to outperform the EC scheme for *very small delay performance* cases. However, for *worst delay performance* cases, it has been shown that A-EC yields a poor delivery ratio and/or a very large delivery delay when *black-holes*⁴ are present in the network [12].

Taking advantage of the strengths of the EC and A-EC schemes, Chen et al. proposed a hybrid scheme, called HEC [12], to achieve better message delivery performance in both *worst delay performance* and *very small delay performance* cases. In the HEC scheme, two copies of EC blocks (constructed based on the erasure coding and replication techniques described previously) are transmitted by the sender. The first copy of EC blocks is sent in a similar way to how the original EC scheme sends blocks, while the second copy is sent using aggressive forwarding during the residual contact time after sending the first EC block. For general opportunistic network scenarios (i.e., without black-holes), the HEC scheme utilizes each contact opportunity better because of the aggressive forwarding feature; however, if black-holes are present in the network, the scheme’s performance is expected to be similar to that of the EC scheme, which achieves better forwarding in *worst delay performance* cases.

The performance of the HEC scheme depends to a large extent on the message scheduling algorithm used in the aggressive forwarding phase, which is not discussed in [12].

However, we are interested in assessing the impact of different message scheduling algorithms on the performance of HEC routing. To this end, we propose three message scheduling algorithms in the following subsection.

B. Message Scheduling in HEC

We propose three message scheduling algorithms, namely Sequential Forwarding (SF), Full Interleaving (FI), and Block-based Interleaving (BI), for transmitting the second copy of erasure coded blocks (i.e., using aggressive forwarding) in HEC scheme. For simplicity, we assume the data file to be transferred in an opportunistic network is L messages in size. After applying erasure coding (i.e., adding redundancy), the size of each message is N erasure code blocks. We denote the n -th block of the l -th message as $M_{l,n}$. Moreover, we assume that each message can be reconstructed when at least B out of N blocks are successfully received by the destination node. We now describe the three algorithms in detail.

1) *Sequential Forwarding (SF)*: In the Sequential Forwarding (SF), the second copy of erasure coded blocks is sent sequentially in accordance with the order of the messages. The main advantages of this scheme are (a) it is intuitive and easy to implement, and (b) it requires the minimal amount of buffer on the sender’s side (i.e., it does not need to apply erasure coding to all messages in advance).

2) *Full Interleaving (FI)*: Different from the Sequential Forwarding (SF) scheme, the Full Interleaving (FI) algorithm interleaves the second copy of the erasure coded blocks for HEC. More precisely, while performing aggressive forwarding, the FI algorithm transmits the “first” coded block of all the messages at the outset, then the second block of all the messages, and so forth. The advantages of this scheme are (a) it distributes the blocks of each message in a more diverse manner, and is thus expected to be more resilient to *black-hole* scenarios than the SF algorithm; and (b) since a message can be reconstructed after receiving just a portion of all the coded blocks, there should be less overall delivery latency than in the SF algorithm.

3) *Block-based Interleaving (BI)*: The main drawback of the FI scheme is the very long response time needed to reconstruct messages when L and/or B are large (i.e., the time between sending the first block and successfully reconstructing the first message). More specifically, the response time of the FI scheme is much greater than the time required to forward the first $L \times (B - 1) + 1$ blocks⁵. A clever solution to this problem is to send B blocks during each contact period and interleave the sending process among L messages, instead of just sending a single block, as in the FI scheme. This is called the Block-based Interleaving (BI) scheme. Note that, the FI scheme is a specialized case of the BI scheme with B equal to 1.

IV. LAYERED MULTIPLE DESCRIPTION CODING WITH UNEQUAL ERASURE PROTECTION

In this section, we propose the use of Layered Multiple Description Coding (LMDC) with unequal erasure protection

³For simplicity, following [12][38], we assume that $N=n$ for all cases.

⁴A node in a network is called a *black-hole* if it is either unreliable (e.g., it has very limited battery power and/or buffer size) or it hardly moves towards the destination [12].

⁵The response time would be even larger if we also consider data loss, data disorder, and inter-contact time.

[14] for effective data dissemination in opportunistic networks.

Layered MDC has been proposed as a means of combining Multiple Description Coding (MDC) [17] and Layered Coding [29] for emerging multicast and peer-to-peer audio/video streaming applications. More specifically, multiple descriptions are spread across multiple packets (or paths) via MDC, and transmitted to a collection of clients, thereby reducing packet loss due to network congestion or the failure of unreliable hosts. Applications of MDC include IP-level multicast [13] and application-level multicast [31][32]. Moreover, by using Layered Coding, multimedia data can be encoded into different quality levels, so that clients can play the best possible video/audio quality level according to their capabilities, such as screen resolution and link bandwidth.

By combining MDC and Layered Coding, the Layered MDC scheme spreads the layered data across multiple packets with multiple descriptions. Then, clients can play the layered data as long as the required number of descriptions are received successfully. Of course, the more descriptions a client receives, the better the reconstructed data quality will be. In practice, the Layered MDC scheme is usually implemented in conjunction with Unequal Erasure Protection (UEP) [14], which provides different levels of erasure protection to the Layered MDC blocks by adding different amounts of redundancy (i.e., the more essential the code blocks are, the more protection/redundancy is added). In this study, we propose the use of Layered Multiple Description Coding with unequal erasure protection for video file transfer and web surfing applications in opportunistic networks. Hereafter, we call the resulting scheme *LMDC*.

A. LMDC for Video Transfer

Fig. 1 illustrates the LMDC scheme for video transfer applications. From the figure, we observe that the quality of a layered video frame improves as the size of the collected video bit stream increases. More specifically, if one of the layered video frames is S bytes in size, one can split it into k equal-sized pieces and reconstruct it to Q_i quality level by using the first i out of the k pieces (i.e., the required bit stream size for reconstructing Q_i level frame is $S_i = i \times S/k$).

Each layered video frame is then split among N packets ($N \geq k$) with unequal erasure protection on each frame piece. For instance, the i -th piece of the layered video frame is erasure coded with replication factor r_i and spread among N packets, i.e., the i -th piece of the video frame can be reconstructed by any N/r_i out of the N packets ($r_1 > r_2 > \dots > r_{k-1} > r_k$ and $N \geq k$). The size of the i -th coded frame piece is

$$b_i = \frac{(S_i - S_{i-1}) \times r_i}{N} = \frac{S r_i}{kN} \quad (2)$$

and the size of the resulting N packets is

$$b_{packet} = \sum_{i=1}^k b_i = \frac{S}{kN} \sum_{i=1}^k r_i. \quad (3)$$

For simplicity, in this study, we let $r_i = N/i$ and $N = k$. The values of b_i and b_{packet} can thus be obtained by

$$b_i = \frac{S}{ik} \quad (4)$$

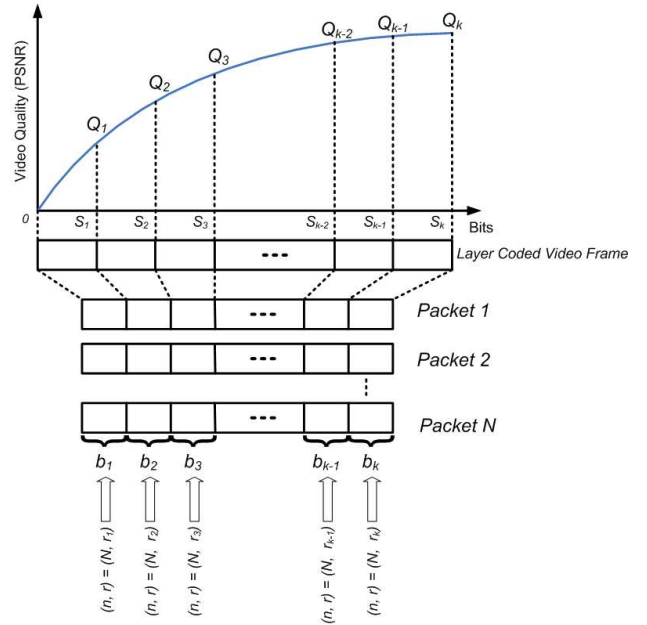


Fig. 1. Illustration of the Layered MDC scheme with unequal erasure protection. Each video frame is encoded into k quality levels using layered coding, and the i -th quality level video frame is erasure protected with replication factor r_i and split equally among N relays ($r_1 > r_2 > \dots > r_{k-1} > r_k$ and $N \geq k$).

and

$$b_{packet} = \frac{S}{k} \sum_{i=1}^k \frac{1}{i}. \quad (5)$$

Moreover, comparing with the Layered Coding scheme, the traffic overhead of the LMDC scheme is

$$b_{overhead} = N b_{packet} - S = S \sum_{i=2}^k \frac{1}{i}. \quad (6)$$

Since k is a positive integer, one can conclude that (a) $b_{overhead} = 0$ when $N = k = 1$ (i.e., no LMDC); and (b) $b_{overhead} > 0$ otherwise.

B. LMDC for Web Surfing

In addition to video file transfer applications, we apply the LMDC scheme to web surfing applications in opportunistic networks. More precisely, the LMDC scheme is applied to each MHTML document [33] that is a MIME (Multipurpose Internet Mail Extensions [7]) HTML document enclosing one or more objects, such as text, images, and videos. Unlike video transfer applications, the layered coding scheme encodes MHTML documents by looking up a pre-determined *codebook*, rather than splitting messages into equal-sized pieces. For instance, we present a codebook that is based on the MIME type of each web object, as shown in Table I.

In the codebook, Layer 1 web objects can be HTML documents (source codes only), cascading style sheets (CSS), java script, or plain text. Layer 2 contains images files in addition to Layer 1 objects. Layer 3 contains additional application objects (such as PDF files and flash games). Layer 4 contains additional video objects. Layer 5 contains other objects (e.g.,

TABLE I
LAYERED CODING FOR WEB OBJECTS

Layers	MIME types
1	text/{html,css,plain,javascript,xml}; application/javascript
2	Layer 1 + image/{gif,jpeg,png,bmp,x-icon}
3	Layer 2 + application/{pdf,octet-stream,x-shockwave-flash}
4	Layer 3 + video
5	Layer 4 + others

audio and unknown objects). Note that the design of the codebook can be customized according to an object's size, semantics, importance, and other design choices. The approach proposed in this paper is based on the generic LMDC concept.

Similar to video transfer applications, the LMDC scheme first encodes each MHTML document into k quality levels using layered coding. The layered document is then split among N packets ($N \geq k$) with unequal erasure protection on each layered piece. More precisely, the i -th layered piece contains the i -th layered document, excluding the $(i - 1)$ th layered piece. Unequal erasure protection is applied to each layered piece, such that the i -th piece is erasure coded with a replication factor equal to i and split among N packets⁶ (i.e., the i -th layered document can be successfully reconstructed from any i of the N packets). Note that, the i -th layered piece may be null if the original MHTML document does not have the corresponding types of objects as specified in the codebook.

For example, using the codebook shown in Table I, each MHTML document is encoded into 5 layers using LMDC (i.e., $k = 5$). Fig. 2 shows the quality of sample web documents in Layer 1, 2, and 3. As shown in the figure, Layer 1 provides basic text-only descriptions at the lowest quality level, Layer 2 adds images of intermediate quality to the descriptions, and Layer 3 adds flash animations of the highest quality. The quality of the web document improves incrementally as the layer number or the amount of received data increases.

V. EVALUATION

In this section, we evaluate the delay performance under file transfer workload in opportunistic networks. We implemented the EC, HEC-SF, HEC-FI, and HEC-BI schemes and performed simulations in DTNSIM [2], a java based DTN simulator. For video transfer applications, we applied Layered Coding to a 2000-frame video clip using JPEG2000 [4] codec, and added unequal erasure protection to each video layer so that the video was LMDC encoded. For simplicity, the number of video quality levels, k , and the resulting packets for each frame, N , were both set to 10 in the simulation. Moreover, we assume there are no black-holes in the evaluating scenarios.

In addition, for web surfing applications, we assume there exists a gateway (say, Performance Enhancing Proxy, PEP) bridging the Internet and the opportunistic network (i.e., similar to the second HTTP scenario in [30]). The gateway archives web documents (in MHTML format) and delivers them to the opportunistic network after LMDC coding. To

⁶Note that the replication factor of each layered piece should be carefully configured according to the performance requirements of the network system.

TABLE II
OBJECT PROPERTIES OF THE SELECTED WEB DOCUMENTS.

Object types	Request (%)	Avg. Size (bytes)
Layer 1	27.81	9,082
Layer 2 excluding Layer 1 objects	63.11	6,974
Layer 3 excluding Layer 2 objects	8.17	66,725
Layer 4 excluding Layer 3 objects	0.15	2,028,783
Layer 5 excluding Layer 4 objects	0.76	182,289

TABLE III
DISTRIBUTION OF THE REQUIRED LAYER NUMBERS, WHICH CAN RESULT IN FULL QUALITY DOCUMENT, OF THE SELECTED WEB DOCUMENTS.

Layer #	Number of documents	% of documents
Layer 1	0	0%
Layer 2	240	48%
Layer 3	240	48%
Layer 4	15	3%
Layer 5	5	1%

be realistic, for the simulation, we selected a set of web documents comprised of the top 500 requested documents according to the hit-count statistics of our campus proxy server for the period Apr.'06 to Sept.'06, and applied LMDC to the selected documents in the simulation. Table II shows the embedded object properties of the selected web documents, and Table III shows the distribution of the selected documents that achieved the best possible quality with limited layers (i.e., a web document may achieve the best possible quality with fewer layers if it does not contain objects located in higher layers). The average number of layers required for the best possible quality documents in this study was 2.57. Moreover, we also assumed that data transmission is wireless at a fixed rate of 1Mbps. The simulation results were all obtained by taking the average performance of 200 simulation runs. In each simulation run the source and the destination pair was randomly selected from all participating nodes.

A. Evaluating Scenarios

We evaluate three network scenarios. One is generated according to the power-law distribution by setting both inter-contact time and contact duration of the network with power-law distributed values of coefficient 0.6 (as reported in [21]). The scenario consists of 34 participating nodes. The other two scenarios are based on realistic campus wireless network traces, i.e., the Dartmouth [1] and UCSD [5] traces, which are publicly available for research references. Table IV outlines the basic properties of the three network scenarios⁷, and Fig. 3 shows the distributions of the UCSD and Dartmouth traces⁸.

More specifically, the UCSD trace is a client-based trace that records the availability of WiFi-based access points (APs) for each participating portable device (e.g., PDAs and laptops) on the UCSD campus. The network trace covers a two and half-month period, and there are 273 participating devices. Similar to [10][21], we assume that two participating devices in ad hoc

⁷In the Dartmouth network trace, there were 13,888 devices in the network, but only 5,148 of them had contact with other devices.

⁸The inter-contact time distributions of the UCSD and Dartmouth traces have been shown to be heavy-tailed with the parameter α equal to 0.26 and 0.47 respectively, and both of them have also been proved to be self-similar [11].



Fig. 2. Examples of the MLB.com web page after layered coding (Layer 1: the web document contains text only; Layer 2: the web document contains images in addition to Layer 1 objects; Layer 3: the web document contains flash animations in addition to Layer 2 objects).

TABLE IV
THE PROPERTIES OF THE NETWORK SCENARIOS IN THIS STUDY.

Trace Name	Power-Law	UCSD	Dartmouth
Device	N/A	WiFi-based PDA	WiFi Adapter
Duration (days)	16	77	1,177
Devices participating	34	273	5,148
Number of contacts	25,959	195,364	172,308,320
Avg # Contacts/pair/day	2.89205	0.06834	0.01105

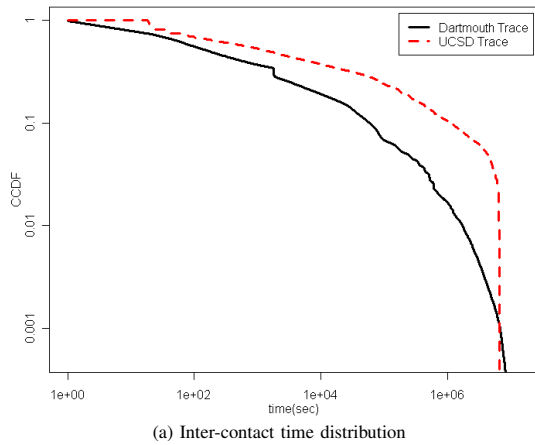
mode encounter a communication opportunity (i.e., a network contact) if and only if they are both associated with the same AP during the same time period.

The Dartmouth trace is an interface-based trace that recorded the APs that associated with a particular wireless interface during a three-year (1,177 day) period. However, it should be noted that, wireless interfaces can be used by different devices at different times, and each device may use multiple wireless interfaces. For simplicity, we assume that each network interface represents a single mobile user in the network. Moreover, like in the UCSD scenario⁹, a network contact is encountered when two mobile users are associated with the same access point.

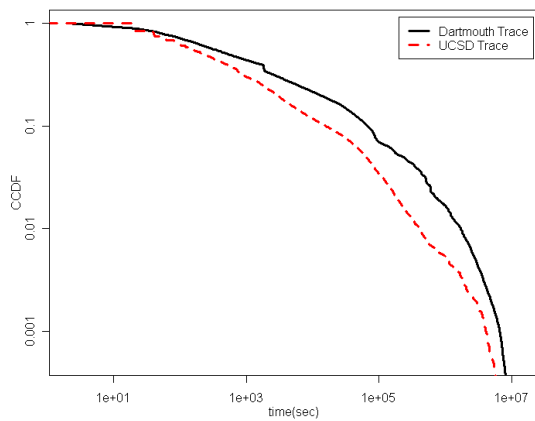
B. Evaluation: Data File Transfer

In the first set of simulations, we evaluate the delay performance of our proposed message scheduling algorithms (i.e., HEC-SF, HEC-FI, and HEC-BI) for data file transfer in opportunistic networks. Three network scenarios are examined via simulations, and a huge data file (i.e., 100MBytes) is selected for the file transfer. For all schemes, the code block size b is set to 125 bytes, the replication factor of the erasure coding r is set to 2, and each coded message is fragmented into 16 equal-sized blocks (i.e., $N = 16$), which is consistent with the settings used in [12][38]. The simulation results are obtained by taking the average performance of 200 simulation runs, and in each simulation run the source and the destination pair is randomly selected from all participating nodes. For simplicity, we assume there are no black-holes in the evaluating scenarios. Fig. 4 depicts the average data latency distribution results in Complementary CDF (CCDF) curves.

From Fig. 4, we observe that HEC-based schemes (i.e., SF, BI, and FI) outperform the EC scheme in almost all test cases, which further affirms the results of the previous studies [12]. Specifically, the three variants of HEC perform nearly identically in the Power-Law scenario. They can successfully



(a) Inter-contact time distribution



(b) Contact duration distribution

Fig. 3. Distributions (CCDF) of the UCSD and Dartmouth traces in Complementary CDF (CCDF) curves.

⁹Note that, although the Dartmouth trace is longer than the UCSD trace and has more participating mobile nodes, its network connectivity is actually very poor. This is because network contacts (for each source-destination pair) occur much less frequently than in other scenarios (only one-sixth of the UCSD scenario and 0.4% of the Power-Law scenario).

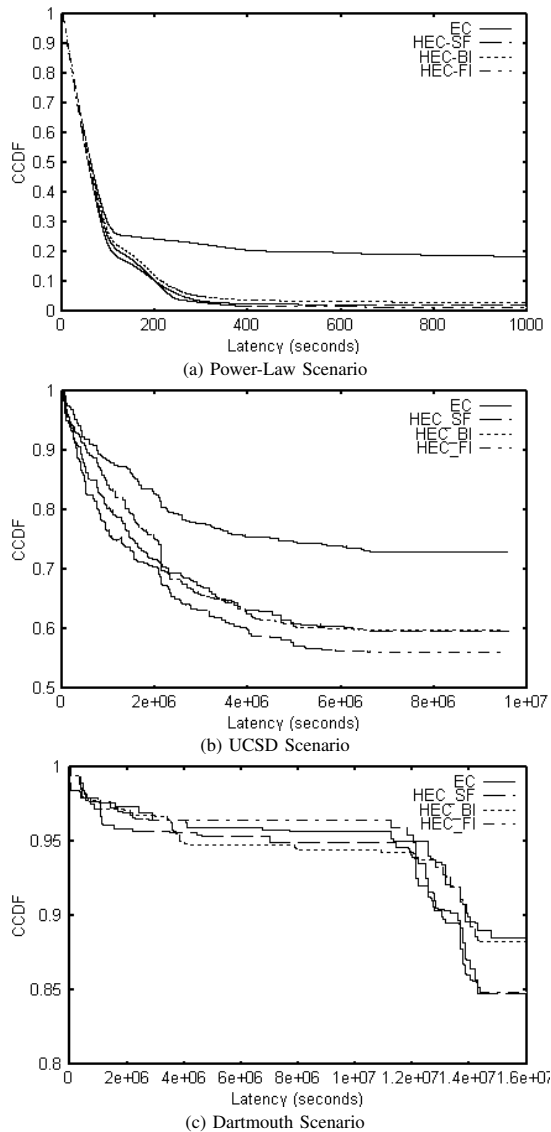


Fig. 4. Distribution (CCDF) of average latency performance of the EC, HEC-SF, HEC-FI, and HEC-BI schemes ($N = 16$ and $r = 2$).

transfer more than 96% of the data file in the simulation, compared to 80% transferred by the EC scheme.

However, we note that the *completion ratio* degrades as the network connectivity (i.e., the average number of network contacts per node pair, per day) decreases. For instance, while HEC-based schemes can achieve about 96% completion ratio in Power-Law scenario, they can only achieve about 45% in the UCSD scenario and 15% in the Dartmouth scenario. Thus, it is quite difficult to transfer complete data files in an opportunistic network, or it requires a much longer transfer time than that employed in the simulation scenarios. Therefore, from an end user's point of view, transferring data files in an opportunistic network may not be feasible, unless the receiver has the ability to read *partial* information from an incomplete file. This finding motivates us to investigate transferring video files with the LMDC technique. We present the evaluation of LMDC-based video file transfer in the next subsection.

Additionally, the HEC-SF and HEC-BI schemes perform

similarly in all scenarios, and perform consistently better at the beginning of the three examined scenarios. This is in contrast to the HEC-FI scheme, which only performs better above a certain latency value. These results seem to contradict our initial intuition that HEC-BI should achieve the best performance at all times. The reason is that the HEC-SF and HEC-BI schemes achieve better *contact efficiency*, since they allow the receiver to reconstruct the original message as long as a certain number of coded blocks are received. In contrast, the interleaving nature of the HEC-FI scheme requires the receiver to wait for at least a few contacts (depending on the erasure coding parameter, r) before it can collect sufficient coded blocks to reconstruct the message. As a result, as shown in Fig. 4, HEC-SF and HEC-BI usually perform better or more aggressively in cases with small latency, which represents good network connectivity (i.e., the left portion of the figure), and HEC-FI usually performs better or more resiliently in cases with the worst latency, which represents poor network connectivity (i.e., the right portion in the figure).

C. Evaluation: Video File Transfer

In the second set of evaluations, we investigate the performance of video file transfer (in terms of Peak Signal-to-Noise Ratio, PSNR) in opportunistic networks with and without LMDC-based coding schemes. We applied Layered Coding on a 2000-frame video clip using JPEG2000 [4] codec, and we also added unequal erasure protection to each video layer in order to have the video become LMDC encoded (i.e., the i -th equal-sized piece of the layered video frame is erasure coded with a replication factor r_i equal to M/i). In the simulation, the number of video quality levels, k , and the resulting packets for each frame, N , are both set to 10. Note that the LMDC scheme (with unequal erasure protection) is similar to the EC scheme, except that LMDC employs multiple redundancy levels for each individual video quality layer, instead of using just one redundancy level for the whole message. Moreover, we apply the concepts of HEC-SF/FI algorithms to the LMDC scheme by sending the second copy of the LMDC blocks during the remaining contact time as in the HEC-SF/FI scheme. We call the resulting schemes LMDC-SF and LMDC-FI respectively. Note that there is no LMDC-BI scheme in our evaluation, since $N = k$, i.e., the minimal number of blocks required to reconstruct an original quality video is exactly the same as the number of blocks transferred over the network¹⁰.

We only evaluated the performance of video file transfer in Power-Law and UCSD scenarios, since the network connectivity of Dartmouth scenario is very sporadic and the data delivery is very poor, as discussed in the previous subsection. For the Power-Law and UCSD scenarios, we took the average performance results of 200 runs and, in each run, we randomly selected one node as the video source and one node as the video destination. The video file was also 2000-frames in length, and all simulation parameters (i.e., transmission rate and erasure coding parameters) were the same as the previous

¹⁰When $N > k$, one can derive the LMDC-BI scheme by simply applying the HEC-BI algorithm with the number of blocks set equal to k .

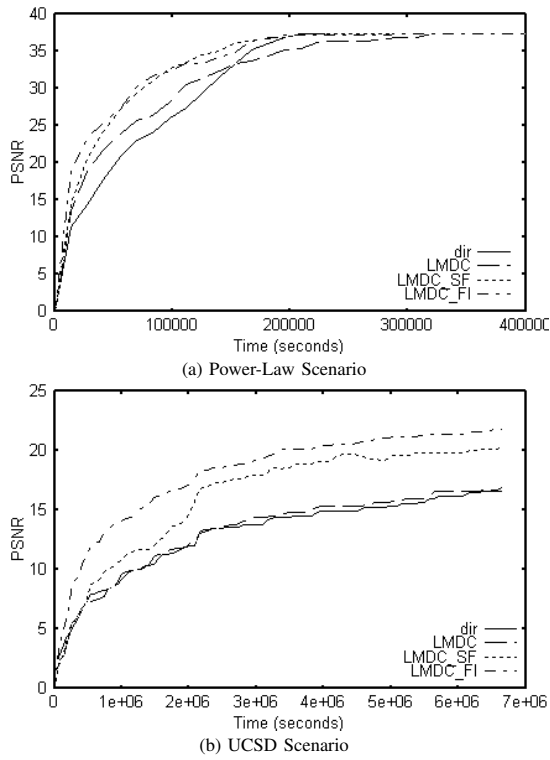


Fig. 5. Average quality of video transfer using direct contact and the LMDC-based schemes. (Both N and k parameters are set to 10)

experiments. Fig. 5 shows the average PSNR performance of the 2000-frame video file using different coding and forwarding schemes.

Fig. 5 shows that, as expected, the HEC based schemes (i.e., LMDC-SF and LMDC-FI) outperform the LMDC scheme and the *dir* scheme, which employs the *direct contact* algorithm [38] to transfer video file directly without LMDC coding. More precisely, the LMDC-FI scheme would performs comparatively better than the LMDC-SF scheme in the simulation. This is because the FI-based strategy combines both aggressive forwarding and interleaving techniques that not only make aggressive use of precious network contacts, but also do their best to alleviate the negative effects of possible black-holes.

The performance of the LMDC scheme is only slightly better than that of the *dir* scheme. This result contradicts our intuition about the LMDC scheme’s added resilience to error-prone and/or poorly-connected networks [14]. The similarity in performance is due to the amount of overhead carried by LMDC (i.e., layered coding and erasure protection). The overhead is too high to achieve a performance gain in extremely challenging network scenarios.

Apart from looking at the overall average PSNR of the video, it is also important to consider the frame-by-frame PSNR performance of video file transfers, since the variance in the frame-by-frame PSNR can also have a strong impact on the perceived video quality for the end user. Fig. 6 shows the frame-by-frame PSNR performance of three time points in the UCSD scenario, and it is clear that the frame-by-frame PSNR quality consistently improves as each encoding/forwarding scheme is given more time (also evident in Fig. 5). Moreover,

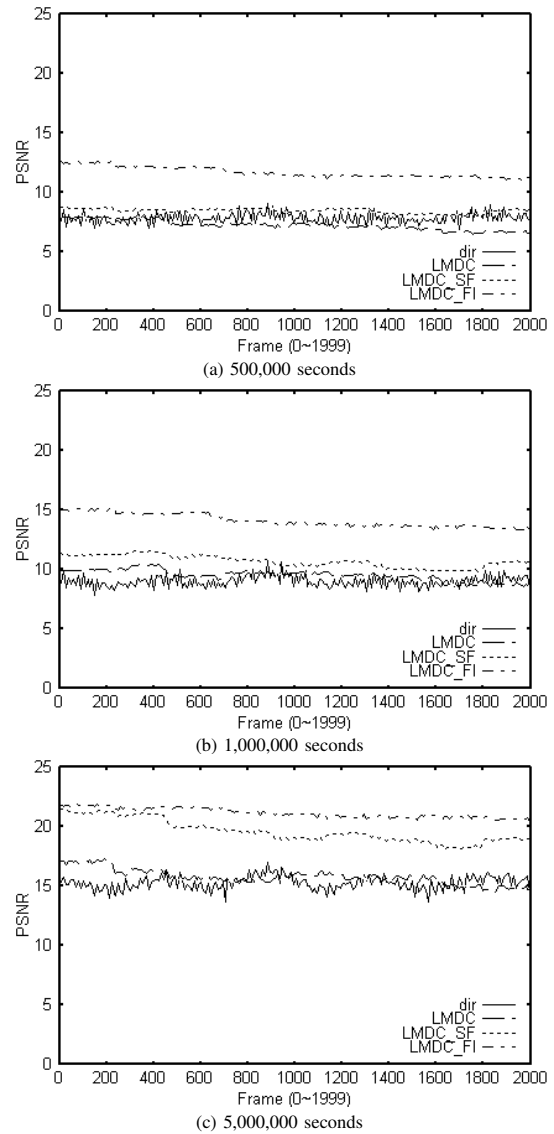


Fig. 6. Comparison of average video quality (i.e., PSNR for each video frame) for the UCSD scenario after 500,000, 1,000,000 and 5,000,000 seconds.

we note that the LMDC-FI scheme consistently outperforms the other schemes for all three time points; whereas the LMDC-SF scheme performs similarly to the LMDC and the *dir* schemes in the beginning, but clearly outperforms those two schemes after 1,000,000 seconds. The results confirm that the *aggressive forwarding* phase can significantly enhance the performance of data forwarding in opportunistic networks. They also demonstrate that the interleaving technique is useful for spreading LMDC blocks over the network, which alleviates the influence of potential black-holes.

Another noteworthy point evident in Fig. 6 is that the average PSNR value of all LMDC-based schemes degrades slightly as the number of frames increases. This is because the schemes all basically send video frames (or coded blocks) in sequence, regardless of whether it is in the first regular EC sending phase or in the second aggressive forwarding phase. This problem can be easily solved by sending video frames in a uniformly random order; however, the tradeoff in

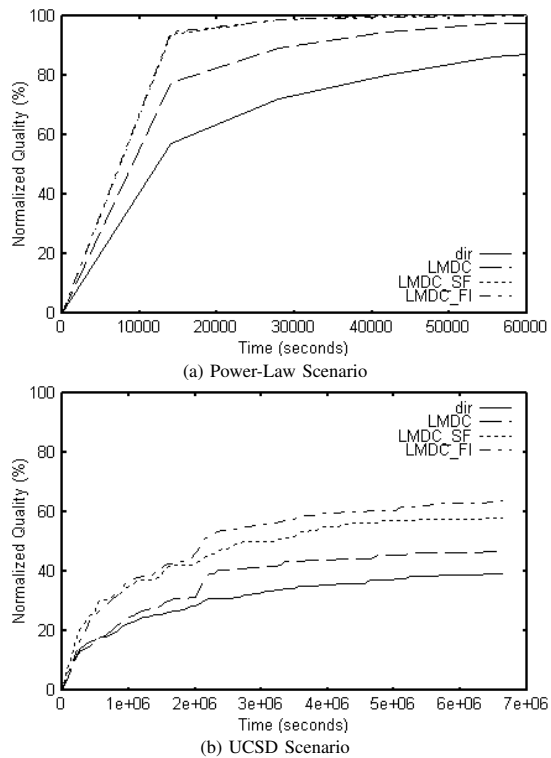


Fig. 7. Average quality of web surfing results using the *dir* and LMDC-based schemes. ($N = k = 5$, and the results have been normalized to the average quality of the employed web documents, i.e., 2.57)

this instance is the computation overhead and memory space overhead increase.

It should also be noted that, in Fig. 6, the PSNR value of the *dir* scheme oscillates heavily in terms of frequency and amplitude, whereas the curves of LMDC-based schemes are much smoother (especially the curve for the LMDC-FI scheme). This is because, in the *dir* scheme, each video frame is either successfully received or completely lost; there is no intermediate quality video that can be played by the end user. As a result, the per frame PSNR performances of the *dir* scheme is subject to large variations. As noted in [44], such drastic PSNR variations are detrimental to the end users' perceived video quality. Therefore, based on our observations above, LMDC-based schemes can indeed yield higher frame-by-frame PSNR performance for video transfer in opportunistic networks, as well as providing better perceived quality video to the end users.

D. Evaluation: Web Surfing

Here, we evaluate the performance of web surfing applications in opportunistic networks with and without LMDC-based coding schemes. Similar to the video file transfer cases discussed in the previous subsection, the LMDC-based schemes (i.e., LMDC, LMDC-SF, and LMDC-FI) are in fact equivalent to the EC-based schemes (i.e., EC, HEC-SF, and HEC-FI schemes), except that they provide different levels of erasure protection to each layered piece of a web document. Moreover, there is no LMDC-BI scheme, since $N = k = 5$ in our evaluation (i.e., using the codebook shown in Table I).

We evaluated the performance of web surfing in the Power-Law and UCSD scenarios, based on the average performance results of 200 runs. In each run, we randomly selected one node as the Internet gateway (which also served as the LMDC encoder) and one node as the web surfer. We used 500 web documents based on the top-500 hit-count statistics of our campus proxy server for the period Apr.'06 to Sept.'06 (the basic properties of the selected web documents are shown in Tables II and III). Fig. 7 shows the normalized average quality of web surfing (with respect to 2.57, which is the average number of layers required for the best possible quality documents) using different coding and data forwarding schemes.

The results in Fig. 7 clearly show that, regardless of the coding and forwarding schemes employed, the average surfing quality improves over time, and eventually converges after a certain period. The reason for this is very straightforward: over time, a web surfer has more chances of making more *contacts* in the network; thus, he/she is more likely to receive his/her requested web documents. The results also show that the LMDC-based schemes consistently outperform the *dir* scheme in both scenarios; more specifically, the HEC-based schemes (i.e., LMDC-FI and LMDC-SF) perform comparatively better than the non-HEC scheme (i.e., LMDC) in all test cases. The reason, not surprisingly, is the same as for video transfers, i.e., LMDC-based schemes spread web documents more widely over the network, and allow the receiver to *preview* part of a document, even before it has been transferred completely. Moreover, since the HEC-based schemes allow more redundancy in data forwarding, the data forwarding performance should be better. Finally, it is worth mentioning that LMDC-FI slightly outperforms LMDC-SF in the simulations, which confirms our previous findings in LMDC-based video transfer applications.

In addition to evaluating the average web document quality in web surfing, we compare the surfing quality of each received web document, which is more representative of the surfing experience of end users. Fig. 8 shows the CDF distribution of the web document quality received by a surfer, based on different coding and data forwarding schemes, at three time points in the UCSD scenario¹¹.

In Fig. 8, it is apparent that, in spite of the coding and data forwarding schemes employed, the CDF curve falls over time, i.e., the curve for 5,000,000 seconds is lower than that for 1,000,000 seconds, and the curve for 1,000,000 seconds is lower than that for 500,000 seconds in all schemes. The results are consistent with our intuition that the quality of web document transfer should improve as the overall delivery latency increases. Moreover, the curves of the LMDC-based schemes are consistently lower than the one of *dir* scheme's curve, and the HEC-based schemes have lower curves than the LMDC scheme. The results again confirm our previous evaluation results that HEC-based schemes outperform all the other schemes considered in this study.

Additionally, Fig. 8 shows that, when the *dir* scheme is employed, about 70% of web documents are unobtainable (i.e.,

¹¹The evaluation results of the Power-Law scenario are omitted due to space limitations.

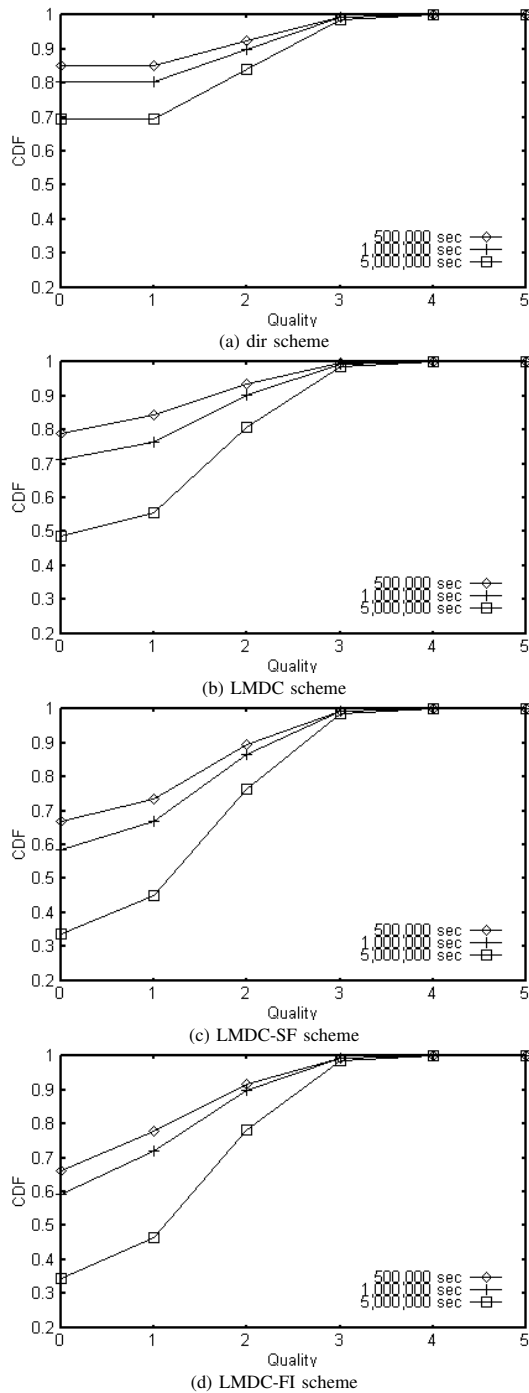


Fig. 8. Comparison of received web document quality distribution (i.e., CDF) after 500,000, 1,000,000, and 5,000,000 seconds using different coding and data forwarding schemes in the UCSD scenario.

the quality level is equal to 0), even after 5,000,000 seconds. In contrast, only about 35% of documents are unobtainable at the same time point when the LMDC-SF or LMDC-FI schemes are used. Moreover, since the dir scheme only provides “one-or-zero” delivery of web documents, it does not allow scalable web surfing when only partial data is received; thus, there are no received Layer 1 quality web documents in the simulation (using the same set of 500 web documents shown in Table III). However, the LMDC-based schemes allow web surfers to

browse lower quality (Layer 1 quality) web documents before they have been transferred completed.

To sum up, the evaluation results demonstrate that the LMDC scheme enhances web document transfer in opportunistic networks by enabling surfers to preview lower quality web documents, even though the data has only been partially transferred. Moreover, in conjunction with the HEC data forwarding strategy, the LMDC-SF/FI schemes are able to better cope with link outages and provide scalable web surfing in challenging network environments.

E. Discussion

As we have examined earlier, the proposed schemes are promising in providing effective data dissemination in opportunistic networks; however, the main drawback of these schemes is the traffic overhead they input into the network, which is considered costly when network resources are not rich (e.g., limited bandwidth and/or buffer size). In this subsection, we present two possible solutions which can potentially reduce traffic overhead caused by the proposed schemes.

1) *Explicit ACK or Passive Cure*: Firstly, the traffic overhead can be possibly reduced by ‘healing’ the network with receivers’ feedbacks upon successful receipt of a complete message. For instance, [8] and [19] have proposed *Explicit ACKs* and *Passive Cure* schemes to alleviate traffic overhead carried by replication-based data forwarding schemes. However, it should also be mentioned that, unlike the scenarios employed in [8][19], where raw messages are inputted into the network, the proposed schemes encodes all messages using erasure coding before injecting them into the network. As a result, the implementation of either Explicit ACK or Passive Cure scheme must be aware of the encoding scheme, i.e., the scheme should acknowledge/cure N erasure coded blocks at a time once $\frac{N}{r}$ of the blocks have been successfully delivered. In addition, the forwarding of receivers’ feedbacks also needs to be carefully controlled. For instance, the *Time-To-Live* [19], *Kill Time* [19], or *Reverse Path Forwarding* [16] like methods should be employed to reduce the traffic overhead caused by receivers’ feedbacks.

2) *Adaptive Coding and Forwarding*: Additionally, the traffic overhead can be reduced by using *adaptive coding and/or forwarding* enhancements. For instance, the adaptive coding enhancement adapts the replication factor, r , of the employed erasure coding scheme in accordance with the observed network status. When the network connectivity is very poor, a larger r can provide better reliability for message delivery with the price of higher traffic overhead; whereas when the network connectivity is decent, choosing a smaller r can greatly reduce the traffic overhead.

Moreover, the adaptive forwarding enhancement adapts the forwarding volume in each network contact based on a *probabilistic estimate*, instead of in a best-effort manner (i.e., sending as much as possible in each network contact). The *probabilistic estimate* can be a function of several factors, such as the mobility pattern [23], the remaining battery life, and the available buffer space of the node. The higher the *probabilistic estimate*, the more likely the encountered node is able to forward the data to the destination.

One should also note that the online measurement (and/or monitoring) of the network is required for cooperating the adaptive coding and forwarding enhancements, e.g., measuring the mobility pattern, and probing the battery life and buffer space. As a result, additional computation and traffic overhead are still necessary. Thus, these enhancements must be carefully designed so that the overall traffic overhead can be reduced.

VI. CONCLUSION

There is an urgent need for an effective data dissemination scheme for opportunistic networks, as communication opportunities in such challenged networks are opportunistic in nature and thus very precious. In this paper, using three types of content as examples (file, video, and web documents), we have presented three message scheduling algorithms that enhance the data delivery capabilities of the HEC scheme for file transfers. In addition, we have proposed a content-centric framework that combines layered coding and multiple descriptions coding to facilitate data dissemination for video transfers and web surfing applications in opportunistic networks. The proposed schemes were evaluated using simulations as well as both synthetic and realistic network scenarios. The results show that the schemes can achieve a much better latency performance for file transfers. Specifically, the results indicate that HEC-BI and HEC-SF achieve a good performance for networks with good connectivity, while HEC-FI yields a more resilient performance in cases of poor network connectivity. Moreover, we have shown that our proposed techniques enable the end user to “preview” video or web content at a lower quality, even before the data has been completely transferred, thereby improving the overall viewing experience. The effectiveness and robustness of our message scheduling algorithms and their corresponding content dissemination techniques make them ideal solutions that can go a long way toward effective data dissemination in opportunistic networks.

REFERENCES

- [1] Crawdad project. <http://crawdad.cs.dartmouth.edu/>.
- [2] Delay tolerant network simulator. <http://www.dtnrg.org/code/dtnsim.tgz>.
- [3] The disruption tolerant networking project at umass. <http://prisms.cs.umass.edu/diesel/>.
- [4] Jpeg 2000. <http://www.jpeg.org/jpeg2000/>.
- [5] Ucsd wireless topology discovery project. <http://sysnet.ucsd.edu/wtd/>.
- [6] The zebrant wildlife tracker. <http://www.princeton.edu/mrm/zebrant.html>.
- [7] N. Borenstein and N. Freed. Mime (multipurpose internet mail extensions) part one: Mechanisms for specifying and describing the format of internet message bodies. Technical report, IETF RFC 1521, Sept. 1993.
- [8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In *IEEE Infocom*, 2006.
- [9] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing Journal*, 2(5):483–502, 2002.
- [10] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *IEEE Infocom*, 2006.
- [11] L.-J. Chen, Y.-C. Chen, T. Sun, P. Sreedevi, K.-T. Chen, C.-H. Yu, and H.-H. Chu. Finding self-similarities in opportunistic people networks. In *IEEE Infocom*, 2007.
- [12] L.-J. Chen, C.-H. Yu, T. Sun, Y.-C. Chen, and H. hua Chu. A hybrid routing approach for opportunistic networks. In *ACM SIGCOMM CHANTS*, 2006.
- [13] P. A. Chou and K. Ramchandran. Clustering source/channel rate allocations for receiver-driven multicast under a limited number of streams. In *IEEE ICME*, 2000.
- [14] P. A. Chou, H. J. Wang, and V. N. Padmanabhan. Layered multiple description coding. In *IEEE Packet Video Workshop*, 2003.
- [15] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou. Challenges: Building scalable mobile underwater wireless sensor networks for aquatic applications. *IEEE Network, Special Issue on Wireless Sensor Networking*, May 2006.
- [16] Y. Dalal and R. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21:1040–1048, December 1978.
- [17] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, pages 74–93, Sept. 2001.
- [18] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *IEEE Infocom*, 2001.
- [19] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer. Delay tolerant mobile networks (dtmns): Controlled flooding in sparse mobile networks. In *IFIP Networking*, 2005.
- [20] X. Hong, M. Gerla, R. Bagrodia, and G. Pei. A group mobility model for ad hoc wireless networks. In *ACM MSWIM*, 1999.
- [21] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *ACM SIGCOMM WDTN*, 2005.
- [22] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *ACM SIGCOMM*, 2004.
- [23] J. Leguay, T. Friedman, and V. Conan. Dtn routing in a mobility pattern space. In *ACM SIGCOMM Workshop on Delay Tolerant Networks*, 2005.
- [24] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for dtms. In *IEEE Infocom*, 2006.
- [25] Y. Liao, K. Tan, Z. Zhang, and L. Gao. Estimation based erasure-coding routing in delay tolerant networks. In *IWCMC*, 2006.
- [26] A. Lindgren and A. Doria. Probabilistic routing protocol for intermittently connected networks. Technical report, draft-lindgren-dtnrg-prophet-01.txt, IETF Internet draft, July 2005.
- [27] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, July 2003.
- [28] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2):569–584, February 2001.
- [29] S. R. McCanne. *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, University of California, Berkeley, 1996.
- [30] J. Ott and D. Kutscher. Bundling the web: Http over dtn. In *WNEPT*, 2006.
- [31] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. Resilient peer-to-peer streaming. In *IEEE ICNP*, 2003.
- [32] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *IEEE NOSSDAV*, 2002.
- [33] J. Palme, A. Hopmann, and N. Shelness. Mime encapsulation of aggregate documents, such as html (mhtml). Technical report, IETF RFC 2557, March 1999.
- [34] J. S. Plank and M. G. Thomason. A practical analysis of low-density parity-check erasure codes for wide-area storage applications. In *The International Conference on Dependable Systems and Networks*, 2004.
- [35] D. Snowdon, N. Glance, and J.-L. Meunier. Pollen: using people as a communication medium. *Computer Networks*, 35(4):429–442, Feb. 2001.
- [36] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, 2000.
- [37] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the postal system into a generic digital communication mechanism. In *ACM SIGCOMM*, 2004.
- [38] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure coding based routing for opportunistic networks. In *ACM SIGCOMM WDTN*, 2005.
- [39] Y. Wang and H. Wu. Dft-mns: The delay fault tolerant mobile sensor network for pervasive information gathering. In *IEEE Infocom*, 2006.
- [40] H. Weatherspoon and J. D. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *IPTPS*, 2002.
- [41] J. Widmer and J.-Y. L. Boudec. Network coding for efficient communication in extreme networks. In *ACM SIGCOMM WDTN*, 2005.
- [42] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *ACM MobiHoc*, 2004.
- [43] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *IEEE Infocom*, 2005.
- [44] M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz. Subjective impression of variations in layer encoded videos. In *IWQoS*, 2003.