Disseminating Multilayer Multimedia Content Over Challenged Networks

Hua-Jun Hong, Tarek El-Ganainy, Cheng-Hsin Hsu, *Senior Member, IEEE*, Khaled A. Harras, *Senior Member, IEEE*, and Mohamed Hefeeda, *Senior Member, IEEE*

***Abstract*—Mobile devices are getting increasingly popular all over the world. Mobile users in developing countries, however, rarely have Internet access, which puts them at economic and social disadvantages compared to their counterparts in developed countries. We propose mBridge: A distributed system to disseminate multimedia content to mobile users with intermittent Internet access and opportunistic ad hoc connectivity. By disseminating various multimedia content, such as news reports, notification messages, targeted advertisements, movie trailers, and TV shows, mBridge aims to eliminate the digital divide. We formulate an optimization problem to compute personalized *distribution plans* for individual mobile users, to maximize the overall user experience under various resource constraints. Our formulation jointly considers the characteristics of multimedia content, mobile users, and intermittent networks. We present an efficient distribution planning algorithm to solve our problem, and we develop several online heuristics to adapt to the system and network dynamics. We implement a prototype system and demonstrate that our algorithm outperforms the existing algorithms by up to 206%, 472%, and 188% in terms of user experience, disk efficiency, and energy efficiency, respectively. In addition, we conduct trace-driven simulations to rigorously evaluate the proposed system in different environments and for large-scale deployments. Our simulation results demonstrate that the proposed algorithm substantially outperforms the closest ones in the literature in all performance measures. We believe that mBridge can allow multimedia content providers to reach out to more mobile users, and mobile users to access multimedia content without always-on Internet access.**

***Index Terms*—Challenged networks, content distribution, multimedia, mobile devices, offline access.**

Manuscript received March 20, 2017; revised June 18, 2017 and July 15, 2017; accepted July 29, 2017. Date of publication August 24, 2017; date of current version January 17, 2018. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, in part by the Qatar National Research Fund under Grant [NPRP8-519-1-108], and in part by the Ministry of Science and Technology of Taiwan under Grant 105-2221-E-007-088 and Grant 106-2221-E-007-101. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Liang Zhou. *(Corresponding author: Cheng-Hsin Hsu.)*

H.-J. Hong and C.-H. Hsu are with the Department of Computer Science, National Tsing Hua University, Hsin-Chu 30013, Taiwan (e-mail: hua.j.hong@gmail.com; cha16@alumni.sfu.ca).

T. El-Ganainy and M. Hefeeda are with the School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: tarek_elganainy@sfu.ca; mhefeeda@sfu.ca).

K. A. Harras is with the Computer Science Department, Carnegie Mellon University, Doha 24866, Qatar (e-mail: kharras@qatar.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2017.2744183

I. Introduction

Multimedia content has dominated the global Internet traffic and shows no trend of slowing down, e.g., Cisco's report predicts that video traffic alone will represent more than 80% of the Internet traffic by 2020 [1]. While such reports demonstrate the importance of disseminating multimedia content, not all *world citizens* have the luxury of high-speed Internet access. In fact, International Telecommunications Union (ITU) reports that only 7% of households in the least developed countries have the Internet access compared with the world average of 46% [2]. In addition, although global mobile subscriptions are excepted to grow to more than 7.2 billion by the end of 2016, a large fraction of these subscriptions do not have access to the Internet, especially in developing countries and rural areas. For example, even including major cities like Cairo, Mumbai, and Shanghai, only about 15%, 21%, and 30% of mobile users in Africa, India, and China have cellular data plans [3]–[5]. The above statistics reveal the so-called *digital divide*, which refers to the inequality among world citizens in accessing and using information and communication technologies. Furthermore, given the increasing trend of disseminating information over the Internet in recent years, instead of traditional media outlets (e.g., radio, TV, and newspapers), the digital divide may have more detrimental social and economic impacts on numerous people living in developing countries and rural areas.

In this article, we propose *mBridge*: a distributed system that tries to reduce the digital divide gap by disseminating multimedia content to users with limited or no Internet access. Specifically, we consider mobile users that may not have cellular data plans, but they are interested in receiving various types of multimedia content, such as news reports, notification messages, targeted advertisements, movie trailers, and TV shows. We assume that content providers, such as news agencies, advertisers, and government authorities, are interested in reaching such users for gaining more revenues (through for example ads) or impacting/informing them (through disseminating important messages and news). To do so, content providers deploy a limited number of *local proxies* in various places such as coffee shops, city halls, public markets, and schools. These local proxies have Internet access through which they can receive multimedia objects from content providers. The local proxies are also connected to local WiFi access points, which can be used by mobile users to connect to the local proxies when they are within the communication ranges.

In Fig. 1, we illustrate the high-level operations of mBridge by considering one day of a user, who owns a mobile device but without a data plan. The user is interested in specific types of multimedia content, say news and cooking TV shows. The user

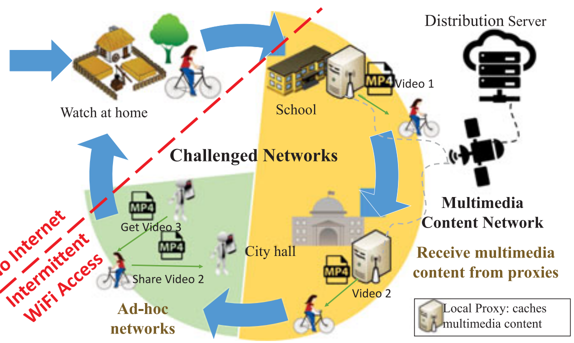

Fig. 1. High-level operations of the proposed mBridge distributed system for distributing multi-layer multimedia content over challenged networks.

takes his/her kids to the school, works at the city hall, and buys food at the market, where WiFi access is available and local proxies are deployed. When the user is in the range of a WiFi access point, the associated local proxy will transmit various multimedia objects to the user's smartphone. These multimedia objects mostly contain what the user is interested in, but they may also contain other objects to be relayed to other mobile users that the user will likely come across during the day. As the user moves throughout the day and runs into other mobile users, the user will exchange different multimedia content with others over an ad-hoc network. By the time the user gets back home, his/her mobile device would have collected (from proxies and other mobiles) most of the multimedia content the user is interested in watching, as well as helped other users in getting the content they are interested in. In addition, based on the routes the user takes, and thus the distributed proxies and the mobile user comes across and their capacities, the user may obtain the desired multimedia objects at different levels of quality.

We notice that the user in the above example has, in fact, an intermittent network access via a non-traditional network called a *challenged network* [6], which suffers from frequent link downs, long queueing delays, high dynamics, and scarce resources [7]. Disseminating multimedia content over the conventional Internet has been studied in the literature [8]–[11]. In particular, Chen *et al.* [8] propose to save bandwidth of streaming services by predicting the viewers' departure patterns. Hu *et al.* [9] build a relay network with multiple servers to reduce the streaming end-to-end latency, and solve a server selection problem. The studies in [10], [11] solve the energy-efficient video delivery problems by a rate adaptation mechanism and multiple network interfaces, respectively. Because these studies [8]–[11] assume always-on Internet access, their solutions do not work in challenged networks. A few other studies consider data communications in challenged networks, e.g., Mota *et al.* [6] and Ntareme *et al.* [12] distribute short messages, such as emails and hazard/criminal alarms, whereas Gao *et al.* [13] propose a solution not designed for multimedia content and heterogeneous user interests. In fact, to our best knowledge, the mBridge system is one of the first of its kind: *it intelligently disseminates multimedia content among users with limited or no Internet access*.

The crux of the proposed mBridge system is to create a personalized *distribution plan* for each user, in order to intelligently distribute multimedia content: (i) at the best time, (ii) to the right mobile users, and (iii) at the highest possible quality. The best time refers to the contact with the best channel condition, which in turn results in faster data transfer and lower energy consumption. The right mobile users are the ones who are likely to be interested in the given multimedia content; otherwise, the transfer energy would simply be wasted. The highest possible quality is measured as the average user experience across all watched multimedia content among all mobile users.

Computing the best distribution plan, however, is challenging due to the complex nature of the multimedia content, mobile users, and intermittent networks. We propose a scalable algorithm that solves this complex (actually NP-Complete) problem and produces near-optimal results (Section V). We implement and validate our solution in an actual proof-of-concept prototype (Section VII), and we rigorously analyze it and compare it against others using detailed, trace-driven, simulations (Section VI).

In particular, this article makes the following contributions.

1) We propose mBridge, which is a distributed system for distributing multi-layer multimedia content to mobile users over challenged networks.
2) We rigorously formulate and solve the distribution planning problem, which is the core optimization problem in mBridge.
3) We conduct extensive simulations using real datasets of online news reports, mobile user trajectories, and user interests. We also carry out a real user study to quantify the user experience of diverse multimedia content. The simulation results show that our algorithm significantly outperforms the existing algorithms: by at least: 20% in terms of user experience, 33% in terms of energy efficiency, and 39% in terms of disk efficiency.
4) We implement and deploy a complete prototype system to demonstrate the practicality and efficiency of our solution. It consists of a distribution server, local proxies, and Android mobile devices. We show performance improvements compared to previous work over a 2-week experiment using 11 local proxies and 31 mobile users.

II. Related Work

Our mBridge tries to deliver multimedia content over opportunistic networks and we deploy local proxies to cache and disseminate multimedia content. The most important and unique design aspect of mBridge is the personalized *distribution plan* which has not been considered by prior studies that only consider: (i) delivering short messages over opportunistic networks, (ii) disseminating multimedia content over wired/wireless networks, and (iii) caching Web content among mobile devices. In the following, we survey the related work in these three categories.

A. Opportunistic Networks

Opportunistic networks include delay-tolerant networks and challenged networks [6]. Delay-tolerant networks have been studied in the literature [7], [14], [15]. For example, Fall [7] proposes a network architecture composed of resource-constrained mobile devices, which is essentially an overlay network above the transport layer. Challenged networks have also been studied, such as flooding [16], message ferrying [17], and social-based forwarding [18]–[20]. Prior studies on forwarding messages in opportunistic networks can be categorized based on assumptions, such as controlled devices, unlimited resources, and predictable contacts. The naive flooding [16] and controlled flooding [21], [22] are less efficient forwarding protocols in practical settings. Message ferrying [17] improves the efficiency based on the knowledge of ferry routes and contact predictability. Device mobility [23], controlled mobility of some nodes [24], exploiting space syntax characteristics [25], [26], and contact history [27] are also used for more efficient opportunistic networks. Capturing intrinsic behavior on social networks, researchers are able to design social-aware forwarding algorithms [18] that consider ranking or centrality information of mobile devices. However, the aforementioned studies focus on short messages instead of multimedia content.

Vahdat and Becker propose Epidemic [16] routing to deliver messages over ad-hoc networks. Epidemic routing transmits all cached content at a node to any other node that gets in contact with it. Epidemic routing results in optimal performance in terms of delivery delay and delivery ratio, but it imposes the highest delivery overhead and assumes infinite buffer space. Hence, researchers, such as [28], in opportunistic networks area often use Epidemic as a baseline for the performance bound under the assumption of unlimited resources. CSI [29] disseminates messages among mobile devices, and leverages the user behavior to improve the dissemination efficiency. In particular, CSI collects mobility data of mobile users to compute their similarity and disseminates the requested messages accordingly. CSI and its variants [30] are probably the closest work to ours, although they only consider short messages. Since Epidemic [16] and CSI [29] are representative schemes in the literature, we adopt them as the baseline algorithms in the evaluations (Sections VI and VII).

B. Multimedia Dissemination

Changuel *et al.* [31] focus on streaming videos to a large number of users. Kang and Mutka [32] use P2P networks to reduce the cost of disseminating multimedia content using cellular networks. Xiang *et al.* [33] design a P2P topology overlay based on clustering mechanisms to improve the availability and Quality of Service (QoS). Mokhtarian and Hefeeda [34] study the resource allocation problem in P2P streaming system with multi-layer scalable videos. In Device-to-Device (D2D) networks, Zhou [35] and Zhang *et al.* [36] optimize for delivery delay and user experience when disseminating multimedia content, respectively. Zhang *et al.* [37] adopt smartphones in cellular networks as helpers to disseminate multimedia content. Unlike our work, D2D studies [35]–[37] assume that cellular infrastructure is available, and smartphone users run into one another very often. Zhang *et al.* [38] propose a hybrid approach of CDN and P2P networks to disseminate multimedia content. MicroCast [39] proposes to group multiple mobile users, to share their cellular connectivities over short-range auxiliary networks for better video streaming quality. Hanano *et al.* [40] utilize both WiFi and cellular networks to disseminate video advertisements. Do *et al.* [41] and HybCast [42] concurrently leverage cellular and ad-hoc networks for (video) file dissemination. Other multimedia dissemination studies [43]–[46] focus on multiple representations of multimedia content for dissemination to clients with heterogeneous resources, such as network, energy, and computing power. Different from mBridge, these multimedia studies are not customized for challenged networks, where clients are often without the Internet access.

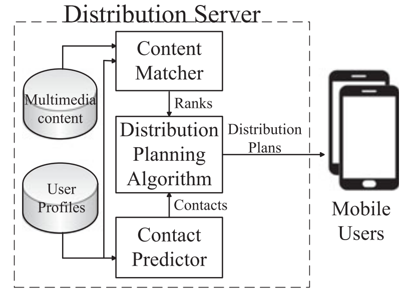

Fig. 2. Architecture of the proposed mBridge distributed system. The core part is the Distribution Planning Algorithm. Other components leverage off-the-shelf machine learning algorithms.

C. Caching

Qian *et al.* [47] show that caching can eliminate redundant network traffic while disseminating Web content. Users' browsing behaviors and machine learning algorithms [48], [49] are adopted by proactive Web content caching. Cooperative caching improves performance of Web applications in opportunistic networks. The technique proposed in [13] caches data in a set of easily accessible mobile devices and exercises the tradeoff between data accessibility and caching overhead. Wang *et al.* [50] leverage the popularity ranks to support cooperative caching under opportunistic networks via BlueTooth or WiFi. Besides, a cooperative caching system [51] is proposed for interactive Web applications over challenged networks. Unlike our distribution planning algorithm that jointly considers multimedia content, mobile users, and intermittent networks, prior studies on caching only consider limited aspects, e.g., Isaacman and Martonosi [51] do not consider the detailed distribution plans.

III. Overview of mBridge

As illustrated in Fig. 1, mBridge operates over two networks: (i) Multimedia Content Network and (ii) Challenged Network. The first network is used to deliver multimedia content over the Internet from a distribution server to local proxies. Proxies are connected to WiFi access points, and they are installed at popular locations, such as coffee shops, city halls, schools, and markets. The second network is used by mobile devices that rarely have Internet access. Mobile users obtain multimedia objects from the proxies as well as other mobile devices.

Fig. 2 shows the three main components of the distribution server: (i) Content Matcher, (ii) Contact Predictor, and (iii) Distribution Planning Algorithm. The first two components leverage multiple off-the-shelf machine learning tools and we briefly describe them in the following. The third component contains our novel algorithm for managing the distribution of multi-layer multimedia content to mobile users over challenged networks; it is described in details in Section V.

The Content Matcher collects user interests, which may be manually specified by mobile users, or derived from collecting recently watched multimedia content. It then determines a ranked-list of multimedia content that matches users interests. This is done by extracting keywords that represent each multimedia object. In our implementation, we use Topia Term Extractor [52], which is a Python package to extract terms using a Parts-of-Speech algorithm. Then, a user-centric ranking is computed using, for example, Google Bayesian [53], RankingSVM [54], or LambdaMART [55].

The Contact Predictor estimates the future contact locations of each user using, e.g., trajectory patterns [56], social networks [57], or a frequency-based approach. Upon the contact locations are determined, the contact durations can be predicted using the techniques proposed in [58]. We note that human mobility is highly predictable, and 85% of the time a mobile user stays at his/her top 5 favorite locations [59].

The distribution server executes the Distribution Planning Algorithm to compute the distribution plans for all known users. The distribution server then pushes the distribution plan of each mobile user along with the user's profile to the local proxies that are on the user's contacts. The mobile user fetches the distribution plan when being within proximity of any of these local proxies. Some mobile users may fail to find their own distribution plans, because they are new to the system or dramatically change their daily trajectories. In this case, the nearest or the current proxy server to which the user is attached to will assign that user the plan of the closest mobile user profile. Therefore, even if the exact distribution plan is not available at a local proxy, the most suitable one can be sent to the mobile user.

We consider different types of multimedia content, including news reports, video clips, notification messages, targeted advertisements, movie trailers, and TV shows. Each multimedia content has different representations that are suitable under different circumstances. For example, for mobile users with a few short contacts, distributing videos to them may not always be possible. In contrast, a well-connected tablet computer user may allocate more energy and disk budgets for high-resolution videos. In fact, each multimedia content can be rendered in the following representations: text (if applicable, e.g., for news), images, left-channel audio, stereo audio, and low-, medium-, and high-resolution videos. Advanced codecs may be used to exploit the redundancy across adjacent representations, e.g., Scalable Video Coding (SVC) [60], [61] allows us to *incrementally* encode videos in different resolutions (and other scalability modes), in order to reduce the bandwidth consumption. With these scalable codecs, higher layers are not decodable without lower layers; for example, the high-resolution video representation contains medium- and low-resolution video layers. Hence, there exists a linear dependency among the layers. This is because lower layers typically have much smaller sizes, e.g., if we already request for the image layer, the bandwidth consumed by the text layer is relatively negligible. Last, different user experience is observed when watching multimedia content in different representations. In Section IV, our user study indicates that the user experience improvement follows a saturated function, when the received data amount increases. For example, moving from nothing to the text of a news report is a huge jump, while moving from medium- to high-resolution videos is less dramatic. Such observation also motivates our multiple representation approach.

In this article, we opt for a centralized distribution server for its global view of the whole system, which typically results in better distribution plans compared to distributed approaches with partial views. For example, by adopting centralized Software-Defined Network (SDN) controllers, Google [62] achieves 2–3 times performance improvement. Such superior performance is crucial to mobile devices in challenged networks, where the number of contacts is rather limited. Nevertheless, having a single distribution server may lead to scalability concern, which can be solved by deploying hierarchical distribution servers. In this way, we may deploy more distribution servers based on the numbers of users. One last concern is the *handover* issue that happens when users move across several geographical regions. This, however, is not a serious concern in our usage scenario, because citizens living in challenged networks often have no access to modern transportation, and are more likely to stay in the same geographical region most of the time.

IV. Crowdsourced User Experience of Different Representations

We carefully design a user study by using a crowdsourcing platform [63] to quantify the user experience of different representations of multimedia content. The user experience of different media types may be quantified in different metrics, such as latency (ms), energy (J), and resolution (pixels). For example, latency is the most important user experience metric for interactive multimedia applications, like teleconferencing calls and online games. In this article, we focus on disseminating news reports to eliminate digital divide in challenged networks, and thus we adopt *understanding level* as the user experience metric. The understanding level is quantified through questionnaires in Mean Opinion Score (MOS) between 1 and 5. We note that the understanding levels may be affected by not only media types and audio/video quality, but also news structure and complexity. To avoid biased results, we retrieve news reports from a reputable news agency (Apple Daily in Taiwan, as an example), and thus the news reports have comparable structure and complexity. We present the detailed user study design below. We emphasize that understanding level is just a sample user experience metric; other user experience metrics should be adopted for other multimedia applications, while our user study procedure is also applicable.

In the user study, each participant watches several random news reports from Apple Daily using a mobile device. After watching each news report, participants fill out their ages and genders. They then answer 5 questions related to the news report. The first question asks the participant to input their understanding level in MOS scores. The scores are normalized to between 0 and 100% in our analysis. The next 4 challenging questions are multiple-choice questions on the news report for checking if a participant really comprehends the news report. We keep track of the number of correctly-answered challenging questions for filtering purpose detailed below. We also allow each participant to skip a news report anytime. A sample questionnaire and our Web interface opened with a smartphone are shown in Fig. 3.

We recruit 182 participants throughout our user study. There are 24 news reports chosen from the following news categories: sports, society, health, finance, politics, history, nature, and life. We generate 5 representations: text, audio, 240p, 360p, and 480p videos, for each news report. For each participant, we play a random news report at a random representation, and a participant may opt to watch more news reports. In total, we gather 2108 user experience scores, and we filter out the scores: (i) with zero correctly-answered challenging questions and (ii) with inconsistent answers in (intentionally) duplicated challenging questions. Eventually, we get 587 valid user experience scores from 120 participants. The average, minimum, and maximum ages of these participants are 29, 16, and 63 years old, respectively. Moreover, 45% of the participants are male. We report the average user experience scores in Table I, and we make two observations out of this table. First, more layers lead to better user experience. Second, the improvement on user

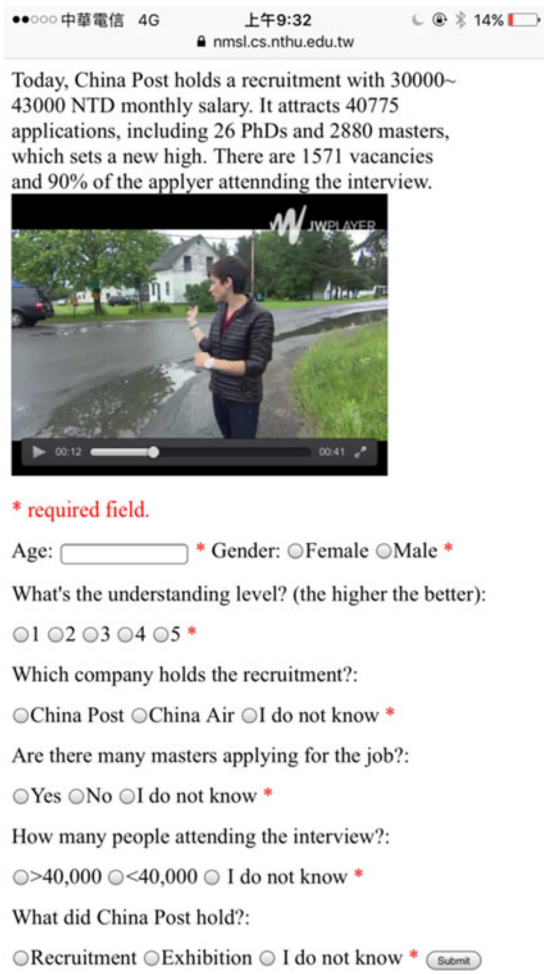

Fig. 3. Sample questionnaire used in our user study.

TABLE I
CROWDSOURCED USER EXPERIENCE SCORES

	Article	Audio	240p Video	360p	480p
Average	55%	68%	71%	74%	77%
No. Samples	112	112	134	104	125

experience is diminishing, e.g., articles give 55% improvement, while 480p videos give 77%-74% = 3% improvement.

The user study results inspire our solution presented in Section V, and are used in our evaluations. We note that some studies [64]–[66] optimize multimedia disseminating system in different QoS metrics, including delivery delay and energy consumption. Compared to these studies, in this article, we aim to eliminate digital divide. Although delivery delay and energy consumption are not our user experience metrics, we carefully design our algorithms and systems to achieve reasonable delivery delay and energy consumption.

V. DISTRIBUTION PLANNING PROBLEM AND SOLUTION

A. Notations

Table II lists all symbols used in this article. We consider a distributed system that delivers multimedia content to U mobile users. Mobile users communicate with S local proxies. Let N be the total number of multimedia objects and L be the number of layers of each multimedia content. Layer l $(1 \leq l \leq L)$ is only decodable/meaningful if all layers $l' \leq l$ have also been received. We define the delivery ***unit*** as a layer of a multimedia content, and unit $i = nL + l$ is a unique identifier pointing to layer l of multimedia content n. We let ρ_i $(1 \leq i \leq NL)$ be the user experience improvement when receiving unit i in addition to all layers beneath it. We let b_i be the size of unit i, and $\psi_{u,n}$ be the probability of mobile user u to watch multimedia content n. We let $\bar{\psi}$ be the minimal viewing probability: a mobile user would not request a multimedia content from another mobile user who is unlikely to watch it.

We let T be the number of time slots that are considered in our formulation, and $t = 0$ be the starting time slot. We assume that mobile users' trajectories are given, i.e., each user's location at every time slot is provided by some localization and prediction techniques. With mobile users' trajectories and local proxies' locations, the sequence of contacts during time $[0, T]$ is determined. Each contact happens between two parties, which can be either mobile users or local proxies. We let C_u be the number of contacts for user u, and $C = \max_{u=1}^{U} C_u$. A mobile user can have multiple concurrent contacts. In this case, it equally divides the contacts into disjoint contacts along the time domain, where each contact has exactly two parties. That is, simple time-division multiplexing is done to avoid interference due to concurrent transfers. We let $p_{u,c}$ be the other party of contact c $(1 \leq c \leq C_u)$ of user u $(1 \leq u \leq U)$, where $1 \leq p_{u,c} \leq U + S$. When $p_{u,c} \leq U$, it points to mobile user $p_{u,c}$, while $p_{u,c} > U$, it points to local proxy $p_{u,c} - U$. Last, we write the duration of contact c of user u as $\kappa_{u,c}$.

TABLE II
SYMBOLS USED THROUGHOUT THIS ARTICLE

Sym.	Description
S	Number of local proxies.
U	Number of mobile users.
N	Number of multimedia content.
L	Number of layers of each multimedia content.
C_u	Number of contacts of user u.
b_i	Size of unit i.
$\psi_{u,n}$	Viewing probability on multimedia content n of a mobile user u.
ρ_i	User experience improvement of unit i.
$\bar{\psi}$	Minimal viewing probability.
$p_{u,c}$	Contacted party of user u in contact c.
$\kappa_{u,c}$	Contact duration of user u in contact c.
$r_{u,c}$	Throughput of user u in contact c.
$\hat{e}_{u,c}$	Per-byte transmitting energy consumption of user u in contact c.
$\check{e}_{u,c}$	Per-byte receiving energy consumption of user u in contact c.
q_u	Energy budget of user u.
d_u	Disk budget of user u.
$r_{u,c}\kappa_{u,c}$	Network budget of user u in contact c.
$\hat{q}'_{u,c}$	Upload energy budget of user u in contact c
$\check{q}'_{u,c}$	Download energy budget of user u in contact c.
$d'_{u,c}$	Disk budget of user u in contact c.
a	Number of days of historical data for training.
τ_u	Contribution potential of user u.
ι_u	Number of unit be selected of user u in each round with our algorithm.
Z_u	A list to store possible units can be downloaded by user u.
F	Maximal segment size.

Combining the contacts with trajectories, we can estimate the throughput and energy consumption of each contact. In particular, we write the receiving throughput of contact c of user u as $r_{u,c}$, the transmitting per-byte energy consumption as $\hat{e}_{u,c}$, and the receiving per-byte energy consumption as $\check{e}_{u,c}$. Last, we use q_u and d_u to represent the energy and disk budgets of mobile user u during $t \in [1, T]$. q_u and d_u are user-specified parameters.

B. Problem Formulation

We first write the distribution plans as $x_{u,n,l,c}$, where $1 \leq u \leq U$, $1 \leq n \leq N$, $1 \leq l \leq L$, and $1 \leq c \leq C$. $x_{u,n,l,c} = 1$ if mobile user u requests unit $nL + l$ during contact c; $x_{u,n,l,c} = 0$ otherwise. Then, we need to keep track of various types of resources: disk space, battery level, and network traffic. We make an important observation: the amounts of resource consumptions are proportional to unit sizes. Hence, we derive a unified budget $R_{u,c}$ for each contact c of user u, which is the resource cap imposed by the rarest resource among disk, battery, and network. In particular, we define $R_{u,c} = \min(\check{q}'_{u,c}, \hat{q}'_{u,c}, d'_{u,c}, r_{u,c}\kappa_{u,c})$, where $\check{q}'_{u,c}$ is the download energy budget, $\hat{q}'_{u,c}$ is the upload energy budget, $d'_{u,c}$ is the disk budget, and $r_{u,c}\kappa_{u,c}$ is the network budget.

The precise derivation of the resource budgets is as follows. We first divide the energy budget q_u into $\check{q}'_u$ and $\hat{q}'_u$ based on the number of contacted local proxies. Specifically, we let $\check{q}_u = \frac{q_u}{2}(1 + \frac{\sum_{c=1}^{C_u} \max(\min(p_{u,c} - U, 1), 0)}{C_u})$, where the term in parentheses is the weight on download energy: running into more local proxies means this user has more chances to download than upload content. We then have $\hat{q}_u = q_u - \check{q}_u$. Next, we allocate the energy budgets to individual contacts, by setting $\check{q}'_{u,c} = \check{q}_u \frac{r_{u,c}\kappa_{u,c}}{\sum_{c=1}^{C_u} r_{u,c}\kappa_{u,c}}$ and $\hat{q}'_{u,c} = \hat{q}_u \frac{r_{u,c}\kappa_{u,c} \max(\min(U - p_{u,c} + 1, 1), 0)}{\sum_{c=1}^{C_u} r_{u,c}\kappa_{u,c} \max(\min(U - p_{u,c} + 1, 1), 0)}$. We notice that we do not allocate upload energy for users who run into local proxies since mobile users never send any multimedia content to local proxies. Last, we allocate the disk budget by setting $d'_{u,c} = d_u \frac{r_{u,c}\kappa_{u,c}}{\sum_{c=1}^{C_u} r_{u,c}\kappa_{u,c}}$, which is also proportional to the network budget normalized across all contacts.

With all the notations developed above, we write our distribution planning problem as

$$\max \sum_{u=1}^{U} \sum_{n=1}^{N} \sum_{l=1}^{L} \sum_{c=1}^{C_{u'}} x_{u,n,l,c}\ \rho_{nL+l}\ \psi_{u,n} \tag{1a}$$

$$\text{st}: \psi_{p_{u',c'},n'} \geq \bar{\psi} x_{u',n',l',c'}; \tag{1b}$$

$$\sum_{n=1}^{N} \sum_{l=1}^{L} b_{nL+l} x_{u',n,l,c'} \leq R_{u',c'}; \tag{1c}$$

$$\sum_{c=1}^{C_{u'}} x_{u',n',l',c} \geq \sum_{c=1}^{C_{u'}} x_{u',n',l'',c}; \tag{1d}$$

$$\sum_{c=1}^{C_{u'}} x_{u',n',l',c} \leq 1; \tag{1e}$$

$$\forall u' \in [1, U], n' \in [1, N], (l' < l'') \in [1, L], c' \in [1, C_{u'}].$$

$$x_{u,n,l,c} \in \{0, 1\}\ \forall u, n, l, c.$$

The objective function in (1a) maximizes the expected overall user experience with viewing probabilities across all mobile users. Equation (1b) makes sure that mobile users never request multimedia content from someone who is unlikely to watch it. Equation (1c) ensures that for each contact: (i) the contact duration is long enough to complete the planned unit transfer under the given transmission throughput, (ii) the total size of planned transmission does not exceed the user's disk budget, and (iii) the total energy does not exceed the energy budget. Equation (1d) captures the layer dependency, i.e., higher layer l'' is only decodable/meaningful when all its lower layers l' are received. Equation (1e) ensures that users do not receive the same unit multiple times, which results in wasted resources.

Lemma 1 (Hardness): The considered distribution planning problem (Problem 1) is NP-Complete.

Proof: We reduce the Multiple Knapsack Problem (MKP) to our Problem 1. The MKP problem is written as

$$\max \sum_{j=1}^{J} \sum_{k=1}^{K} v_j y_{j,k} \tag{2a}$$

$$\text{st}: \sum_{j=1}^{J} w_j y_{j,k} \leq O_k\ \forall k = 1, 2, \ldots, K \tag{2b}$$

$$\sum_{k=1}^{K} y_{j,k} \leq 1; \forall j = 1, 2, \ldots, J \tag{2c}$$

$$y_{j,k} \in \{0, 1\}\ \forall j = 1, 2, \ldots J, k = 1, 2, \ldots, K. \tag{2d}$$

In the MKP problem, we consider J objects and K knapsacks. The boolean decision variable $y_{j,k}$ indicates whether we want to put object j into knapsack k, while v_j represents the profit of having object j. Each knapsack k has its capacity O_k, which is a resource limit and each object j consumes a given amount of resource $w_{j,k}$. The MKP problem strives to pick a subset of objects, so that the total profit is maximized, while none of the constraints are violated.

Given an MKP problem, we generate a corresponding Problem 1, as follows. First, we let $U = 1$, $L = 1$, and $\psi_{u,n} = 1$, which means only one user would like to receive multimedia content from local proxies. Moreover, each content has only one layer and the viewing probability is 100%. Second, because we let $U = 1$, we write the number of contacts C_u of user u as C and the resource constraint $R_{u,c}$ as R_c. We then let $C = K$ and $R_c = O_k$. Third, because we let $L = 1$, we write the size of units b_{nL+l} as b_n and user experience improvement ρ_{nL+l} as ρ_n. We then let $b_n = w_j$ and $\rho_n = v_j$. This results in a proper instance of Problem 1 in polynomial time. In addition, a solution of Problem 1 can be verified in polynomial time. Because we let $U = 1$ and $L = 1$, the decision variable $x_{u,n,l,c}$ can be written as to $x_{n,c}$. Hence, we let $x_{n,c} = y_{j,k}$, which yields a solution of the MKP problem. Since MKP problem is NP-Complete, Problem 1 is also NP-Complete.

C. Optimal Algorithm: DP

We solve the formulation in (1) using a Dynamic Programming (DP) algorithm, which systematically skips redundant computations. DP memorizes the computed user experience of a user u while downloading unit i during contact c with remaining unified budget $R'_{u,c}$, where $0 \leq R'_{u,c} \leq R_{u,c}$. Fig. 4

Input: User profiles, such as viewing probability $\psi_{u,n}$, multimedia content information, such as size of a unit b_i, and resource budgets $R_{u,c}$
Output: Distribution plan M

let M as the table to memorize the computed values
for all user u, contact c, and unit i **do**
    Call Procedure DP $(i, c, R_u, R_{u,c})$
procedure DP$(i, c, R_u, R'_{u,c})$ $R_{u,c} = R'_{u,c}$
    if $R'_{u,c} \leq 0$ **then** //remaining resource is not enough
        return $-\infty$ //return negative user experience
    if $i = 0$ **then** //all the units have been put into the plan
        return 0
    if $M[i][R_{u,1}][R_{u,2}]...[R_{u,C_u}]$ exists **then**
        return $M[i][R_{u,1}][R_{u,2}]...[R_{u,C_u}]$
    let $M[i][R_{u,1}][R_{u,2}]...[R_{u,C_u}] =$
    //download from contact c
    $\max(DP(i-1, c, R_u, R_{u,c} - b[i]) + \rho_i \psi_{u,i/L})\ \forall c \in [1, C_u]$
    let $M[i][R_{u,1}][R_{u,2}]...[R_{u,C_u}] =$
    // do not download
    $\max(DP(i-1, 0, R_u, R_{u,c}), M[i][R_{u,1}][R_{u,2}]...[R_{u,C_u}])$

Fig. 4. Pseudocode of our DP algorithm for solving the distribution planning problem.

shows the pseudocode of our algorithm, which recursively decides whether user u downloads unit i from one of the contacts or does not download the unit, until the unified budget is used up or all units have been considered. The proposed DP algorithm essentially conducts a grid search, and goes through all possible solutions. Hence, the DP algorithm gives optimal solution at an expense of high computational complexity, which is discussed in Lemma 2.

Lemma 2 (Time Complexity): The DP algorithm has an exponential time complexity.

Proof: The dynamic programming algorithm recursively solves the problem. Each recursive function creates other $C + 1$ recursive functions as its children. Thus, the worst-case time complexity of creating a plan for a user is $O((C + 1)^{NL})$. We need to create U plans, thus $O(U(C + 1)^{NL})$ is the time complexity of our DP algorithm. This exponential time complexity may render the DP algorithm not feasible for some large problems. Hence, we develop a heuristic algorithm in the following section. ■

D. Efficient Algorithm: CDRR

We propose an efficient heuristic algorithm, called Contact-Driven Round Robin (CDRR), for larger distribution planning problems. CDRR is based on *three major intuitions*:

1) Deliver higher user experience improvement using less resources (energy, disk, and network budgets).
2) Send multimedia content to mobile users who have more chances to relay the content to others.
3) Download multimedia content from mobile users who have fewer chances to send content to others.

In particular, we first define the resource efficiency as $\nu_{u,nL+l} = \psi_{u,n} \times \rho_{nL+l}/b_{nL+l}$. For each user u, we sort all units that exist on any contact user $p_{u,c}$ in the descending order on resource efficiency. To avoid allocating too much resource to a single user, users take turns to choose units from the sorted lists. In addition, we calculate the number of contacts C_u and the contact duration $\kappa_{u,c}$, and we define the *contribution potential*

Input: User profiles, such as contact duration $\kappa_{u,c}$, multimedia content information, such as size of a unit b_i, and resource budgets, such as the disk budget d_u
Output: The distribution plan $x_{u,n,l,c}$

1: **let** $\mathbf{H} = \{1, 2, 3, \ldots, U\}$ // index set of unfinished users
2: **for** $u = 1$ to U **do**
3:     **store** units existing on any $p_{u,c}$ in a list $\mathbf{Z}_u$
4:     **sort** $\mathbf{Z}_u$ by resource efficiency in the dsc. order
5: **let** $B_{u,c} = \gamma_{u,c}\kappa_{u,c}\ \forall 1 \leq u \leq U, 1 \leq c \leq C_u$ // network budget
6: **while** $\mathbf{H}$ is not empty **do**
7:     **for** $u \in \mathbf{H}$ **do** // iterate in round robin fashion
8:         **let** $\mathbf{R} = \iota_u$ units removed from the head of $\mathbf{Z}_u$
9:         **for** $r \in \mathbf{R}$ **do**
10:             **for** $k = 1$ to C_u **do**
11:                 **if** $d_u \geq b_r$, $q_u \geq \check{e}_{u,k} b_r$, $q_{p_{u,k}} \geq \hat{e}_{u,k} b_r$, and $\gamma_{u,k}\kappa_{u,k} \geq b_r$ **then**
12:                     **if** $\tau_{p_{u,k}} < \tau_{p_{u,c}}$ **then**
13:                         **let** $c = k$ // least contribution potential
14:             **if** c exists and Eq. (1e) is satisfied **then**
15:                 **let** $d_u = d_u - b_r$ // get r from c
16:                 **let** $q_u = q_u - \check{e}_{u,c} b_r$
17:                 **let** $q_{p_{u,c}} = q_{p_{u,c}} - \hat{e}_{u,c} b_r$
18:                 **let** $B_{u,c} = B_{u,c} - b_r$
19:                 **let** $x_{u,\lfloor r/L \rfloor, r \bmod L, c} = 1$
20:         **if** $d_u \leq 0$ or $q_u \leq 0$ or $\mathbf{Z}_u$ is empty **then**
21:             **remove** u from $\mathbf{H}$

Fig. 5. Pseudocode of our CDRR algorithm for solving the distribution planning problem.

of mobile user u as $\tau_u = C_u \times \sum_{c=1}^{C_u} \kappa_{u,c}$. In each round, user u chooses ι_u units. We let $\iota_u = \lceil \frac{\tau_u}{\hat{\tau}_u} \bar{C} \rceil$, where $\hat{\tau}_u$ is the highest contribution potential and $\bar{C} = \frac{1}{U} \sum_{u=1}^{U} C_u$. In this way, users with higher contribution potential choose more units in each round. Then, for each mobile user, we know the units to be downloaded. Last, we determine which contact to download each unit. For each unit, user u selects the contact user $p_{u,c}$ with the smallest contribution potential $\tau_{p_{u,c}}$, as long as the resources of users u and $p_{u,c}$ are enough for transferring the unit. Receiving the unit from this contact user reduces negative impacts on other users.

Fig. 5 gives the pseudocode of the CDRR algorithm. Lines 1–4 implement the first intuition, and save all the units that may be downloaded by mobile user u $(1 \leq u \leq U)$ in a list $\mathbf{Z}_u$, which is sorted on resource efficiency. Lines 6–9 realize the second intuition, where users choose units in a round robin fashion. Each user u gets to select up to ι_u units in each round. Lines 10–19 implement the third intuition to select the contact users to download individual units. Lines 20–21 check if there are residue resources, and terminate once resources are saturated.

Lemma 3 (Time Complexity): The CDRR algorithm terminates in polynomial time.

Proof: We first define $\mathbf{H}$ as the index set of unfinished users. Next, we create the sorted units list $\mathbf{Z}_u$ in lines 2–4, which has a complexity of $O(UCNL\log(CNL))$ since the maximal number of units of any contact user is NL. The for-loops starting from lines 6 and 7 both go through user $u \in \mathbf{H}$ until $\mathbf{H}$ is empty. In each iteration, we take turns to check the units in $\mathbf{Z}_u$. We also make sure that at least one unit is removed and the user u will be removed from $\mathbf{H}$ once $\mathbf{Z}_u$ is empty. Since the maximal number of units in $\mathbf{Z}_u$ is CNL, the complexity of the loop is $O(UCNL)$. Hence, $O(UCNL\log(CNL))$ dominates and thus is the complexity of our CDRR algorithm. ■

TABLE III
SAMPLE RUNNING TIME AND TOTAL USER EXPERIENCE, GEOLIFE

No. Content	Running Time (sec)				No. Content	Total User Experience		
	DP		CDRR			DP	CDRR	Perf. Gap
	Mean	Max	Mean	Max				
1	0.08	0.08	0.05	0.05	1	0.43	0.42	3%
2	0.11	0.11	0.06	0.07	2	0.36	0.34	6%
3	0.19	0.19	0.63	0.66	3	0.33	0.31	6%
4	0.6	0.61	0.07	0.07	4	0.31	0.29	6%
5	1.8	1.8	0.07	0.08	5	0.27	0.25	7%
6	110.2	111.2	0.08	0.08	6	0.29	0.27	7%
7	269.4	457.2	0.08	0.09	7	0.29	0.27	7%

E. Near-Optimality of the Proposed Algorithm

We perform numerical analysis to quantify the performance of CDRR. We use our DP algorithm to optimally solve the formulation in (1), and use it as a benchmark. We adopt real datasets: GeoLife [67], San Francisco [68], and SIGCOMM [69] trajectory traces. The details of the datasets are given in Section VI. The datasets give us number of users and contacts per user. We then vary the number of multimedia content, and solve the distribution planning problem using the CDRR and DP algorithms. We repeat each experiment 5 times and report their running time and user experience.

Table III gives the sample running time and user experience from the GeoLife trace. We draw three observations from the tables. First, the proposed CDRR algorithm runs in real time, and scales to more multimedia content. Second, the DP algorithm leads to prohibitively long running time with large number of multimedia content. Third, we observe that CDRR achieves at least 93% of the user experience compared to DP. In summary, DP can optimally solve our problem under small problem size, but it is not feasible for large problems because of the long running time. In contrast, CDRR can solve larger problem with near-optimal user experience in short time. We note that we only report the *expected* user experience in this subsection, and more detailed simulation results are provided in Section VI. In that section, less-than-perfect optimality of CDRR is further compensated by the practical heuristics presented in Section V-F.

F. Practical Considerations

Our system implementation contains the following practical optimizations.

1) Determining the downloading order: For each contact, a mobile user downloads units based on the plan. After the units planned for a contact are all finished, the units planned for other contacts are downloaded. Once all units on the distribution plan are downloaded (or the plan is empty, which means the mobile user has not received his/her plan), the mobile device shows available units to the user, and allows him/her to select the contents to request. When requesting multiple units, the order is crucial: it is preferred to devote resources to those units that result in higher user experience improvement normalized to unit size. Therefore, in each contact, each mobile device computes ρ_i/b_i of the next outstanding layer of each content. The mobile device requests unit i^* with the highest ρ_{i^*}/b_{i^*}. This is repeated until the contact is over or resources (energy and disk) are used up.

2) Segmenting Video Layers: Because multimedia content could be quite large, we define a maximal segment size F as a system parameter. For unit with size b larger than F, we divide it into $\lceil b/F \rceil$ segments, where the first $\lceil b/F \rceil - 1$ segments are in the size of F. Doing segmentation is to avoid unnecessary retransmission after interrupted transfers. The user experience improvement of a unit is equally split among all segments in that unit; more comprehensive approaches are also possible. Segmentation is done after the distribution plans are computed, because incorporating the concept of segments in the distribution planning problem increases the problem size, which leads to high computational overhead.

VI. TRACE-DRIVEN SIMULATIONS

In this section we use real datasets to analyze the performance of the distribution planning algorithm in a detailed simulator and show that it outperforms other algorithms by wide margins. various metrics including: (i) user experience, (ii) watched user experience, (iii) missed units, (iv) disk efficiency, (v) energy efficiency, and (vi) watched units.

A. Datasets

We employ three datasets: (i) user contacts, (ii) multimedia content, and (iii) user interests to drive our simulator. In order to evaluate our distributed system in different environments, we adopt three user contact datasets: GeoLife [67], San Francisco [68], and SIGCOMM [69]. In GeoLife dataset, there are multiple transportation modes, including walk, bike, bus, taxi, train, and subway. In San Francisco dataset, the transportation mode is taxi, and the participants of SIGCOMM dataset walk in the conference. The 4-year GeoLife and 30-day San Francisco datasets contain the GPS trajectories of 178 and 500 participants, respectively. Using the locations of the participants, we estimate the contact duration using a WiFi range of 55 m, measured under the setup proposed in Wang *et al.* [70] with HTC Desire 620 smartphone. We use the average throughput measured by Friedman *et al.* [71] for each contact. The SIGCOMM dataset contains 76 mobile users' BlueTooth contacts for 3 days. The characteristics of these three datasets are diverse. Therefore, the evaluation results using these three datasets will shed some lights on the performance of our solution in *diverse* environments, including challenged networks.

For the multimedia content dataset, we collect 300 news reports from NBC in 2015, and divide each news report into five layers: text, audio, low-, medium-, and high-resolution videos. The size of each layer is calculated. The shortest and longest news reports last for 12 and 287 secs, respectively. We adopt Topia Term Extractor [52] to extract keywords from the articles. The resulting keywords are used by the Content Matcher. Last, we derive the user interests by leveraging the user queries in the LETOR [72] dataset. In particular, we randomly pick a user query, and take the keywords in it to mimic the user's interests. The keywords in LETOR dataset are different from our NBC dataset, and we map the keywords using their orders of appearance.

B. Simulator Implementation

We have implemented a detailed mBridge simulator using a mixture of Python, Java, and Matlab. The distribution planning algorithm is executed once a day in our simulations. Once the distribution plans are computed, we carry out the simulations following the ground truth given in the datasets.

1) State-of-the-art algorithms: For comparisons, we have implemented two algorithms: (i) Epidemic that transmits all the units when a contact occurs [16] and (ii) CSI that sends the units of interested multimedia content to mobile users based on mobile similarity [29].

2) Performance metrics: We consider the following performance metrics, and report the average performance with 95% confidence intervals whenever applicable.

1) *User experience:* The average user experience (between 0 and 100%) of all the user demanded news reports. We also report *watched user experience* that only considers watched multimedia content.
2) *Missed units:* The number of unavailable news units among all the user demanded ones.
3) *Disk efficiency:* The ratio of per-user user experience and per-user disk consumption.
4) *Energy efficiency:* The ratio of per-user user experience and per-user energy consumption.
5) *Watched units:* The number of watched multimedia content units.

We note that performance metrics used for live video streaming, such as initial buffering time, number of rebuffering instances, and number of dropped frames, are not directly applicable to our mBridge system. This is because mBridge only shows *fully-downloaded* content to users. Therefore, none of the aforementioned, and unfortunate, situations that are common to live video streaming ever occur in mBridge.

3) Content Matcher and Contact Predictor: We have also implemented several machine learning algorithms in Content Matcher and Contact Predictor. The algorithms use historical data up to the past a days to predict contacts and user interests. The Contact Predictor keeps track of the historical contacts, and predicts the future contacts based on frequencies. The Content Matcher first classifies the news reports into several categories and calculates the viewing probability using a Bayesian approach inspired by Google news recommendation [53]. We utilize the BBC dataset [73], which classifies 2225 news reports in 5 categories, to train a classification model following a frequency-based approach, which uses numbers of keyword occurrences in each category for classification. We note that these machine learning algorithms are not developed by us, nor are the most advanced ones. We adopt them to be conservative: the mBridge distributed system will achieve even better performance with updated machine learning algorithms.

4) Simulation Scenarios: We run 3-day simulations using GeoLife, San Francisco, and SIGCOMM datasets. The GeoLife dataset is very sparse as only 3.33% of user-day GPS trajectories are non-empty and the dataset spans over the greater Beijing area. Therefore, we focus on the 88 km^2 downtown area, and create a 3-day trace by choosing the top 30 active days of each mobile user. We assume that only 10% of the residents participated in GeoLife data collection, so that we aggregate 30-day traces into 3-day ones. We remove the mobile users who never get into the downtown area, which yields a trace with 870 mobile users. 80 local proxies are randomly deployed in the crowded locations. For San Francisco and SIGCOMM, we promote 50 and 10 mobile users with the most contacts to be the local proxies, respectively. Table IV shows some statistics of the generalized datasets. We observe that GeoLife dataset has the lowest connectivity, while SIGCOMM and San Francisco have 7 and 96 times more contacts compared to GeoLife. The contact duration of San Francisco is lower than other datasets because the speed of taxi is much higher than other transportation modes. SIGCOMM has the highest contact duration because the attendees in SIGCOMM conference walk and discuss with each other in a small area. These three datasets with different characteristics help us to evaluate our distributed system under diverse environments.

TABLE IV
STATISTICS OF USER CONTACT DATASETS

	Contacts Per Day		Contact Duration (sec)	
	Mean	Std	Mean	Std
GeoLife	2.7	7.1	91	399
SIGCOMM	19	29	526	2606
San Francisco	258	209	25	50

5) Parameters: We vary the number of random NBC daily news reports within $\{10, 25, 50, 100, 200\}$. We use the least disk and energy capacity among the top five smartphones in South Africa [74] as the upper bounds of the disk and energy budgets. We also assume that $\frac{1}{3}$ energy is used by communications, and $\frac{1}{3}$ of communication energy is used by our mBridge app. The upper bound of disk budget is 500 MB and the upper bound energy budget is 1500 J. We then consider the disk budget in $\{15, 30, 60, 125, 250, 500\}$ MB, and the energy budget in $\{100, 200, 400, 800, 1500\}$ J. By default, we pick 100 news reports everyday and set disk (energy) budget to be 125 MB (400 J). Moreover, the per-byte WiFi energy consumption is 9×10^{-7} J [71], and the maximal segment size is 5 MB. For prediction, we let $a = 3$ (days) for predicting the viewing probability and contacts. Last, we set the user experience following Table I.

C. Results

1) The Performance of Our CDRR Algorithm is Near Optimal Under Unlimited Resources: Fig. 6 shows the service quality and resource usage under unlimited resources. We use Epidemic algorithm as a benchmark which gives optimal results in terms of user experience and delivery delay with unlimited resources. To achieve the optimality, Epidemic uses excessive energy and disk to flood news reports. Fig. 6(a) and 6(b) show that our CDRR algorithm achieves near optimal (<3% gap) user experience and low delivery delay (<6% gap), compared to Epidemic algorithms. Regarding system overhead reported in Fig. 6(c) and 6(d), CDRR saves up to 8% energy consumption and 8% used disk space compared to Epidemic algorithm. CSI saves about 67% energy and disk consumption on average, but suffers from 10% longer delivery delay and 20% lower user experience compared to Epidemic. In summary, while CDRR is not designed for environments with unlimited resources, it performs almost optimally in terms of user experience, yet achieves short delivery delay. CSI has lower resource usage, but suffers from higher delivery delay and lower user experience. In real life, the resources are *always* limited, and thus we take limited energy, disk, and network budgets into considerations in the rest of this article.

2) The Proposed CDRR Algorithm Improves the Service Quality: Fig. 7 reports the service quality of individual mobile

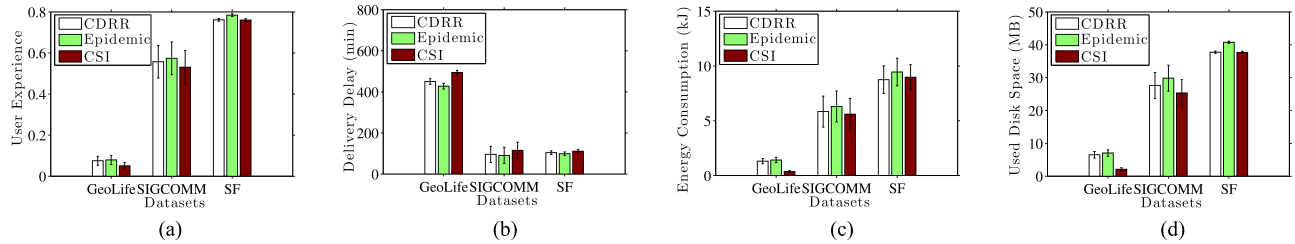

Fig. 6. CDRR with unlimited resources is near optimal, average: (a) user experience, (b) delivery delay, (c) energy consumption, and (d) used disk space over three days.

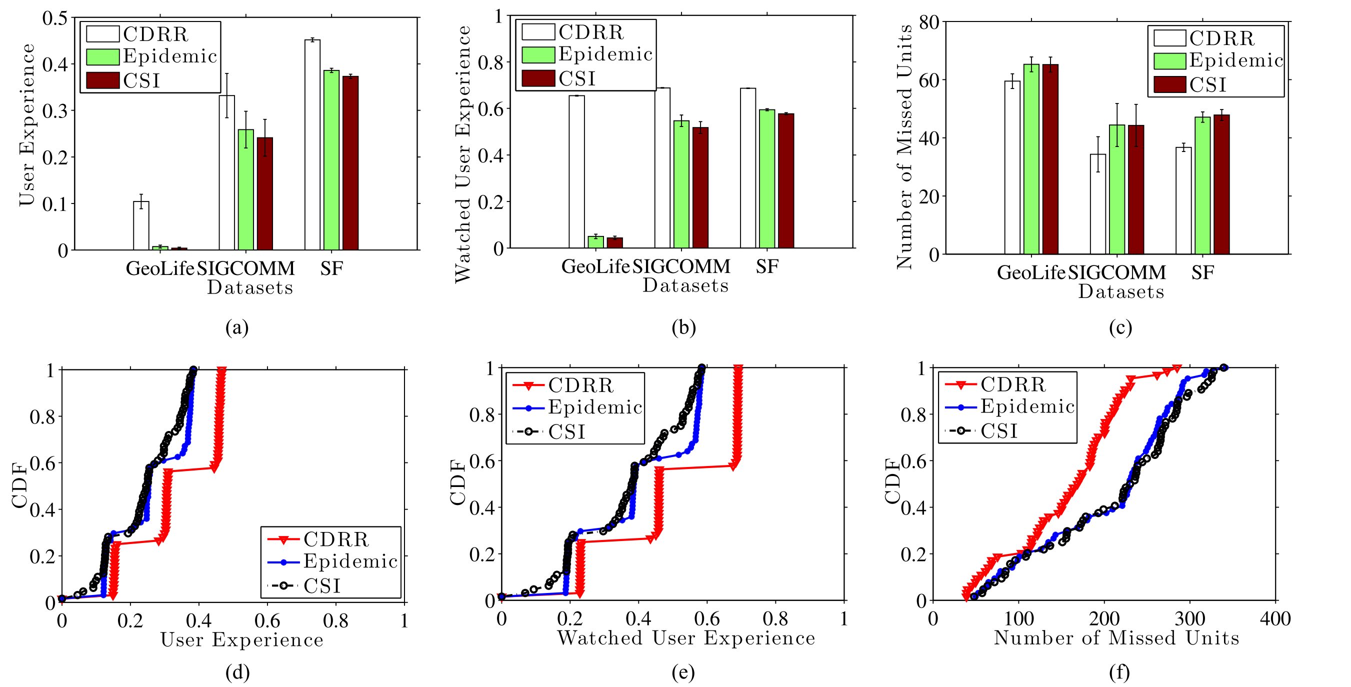

Fig. 7. Service quality improvement of our CDRR algorithm: (a) user experience, (b) watched user experience, (c) missed units, (d) user experience CDF, (e) watched user experience CDF, and (f) missed units CDF over three days. Sample CDFs in (d), (e), and (f) are from SIGCOMM dataset.

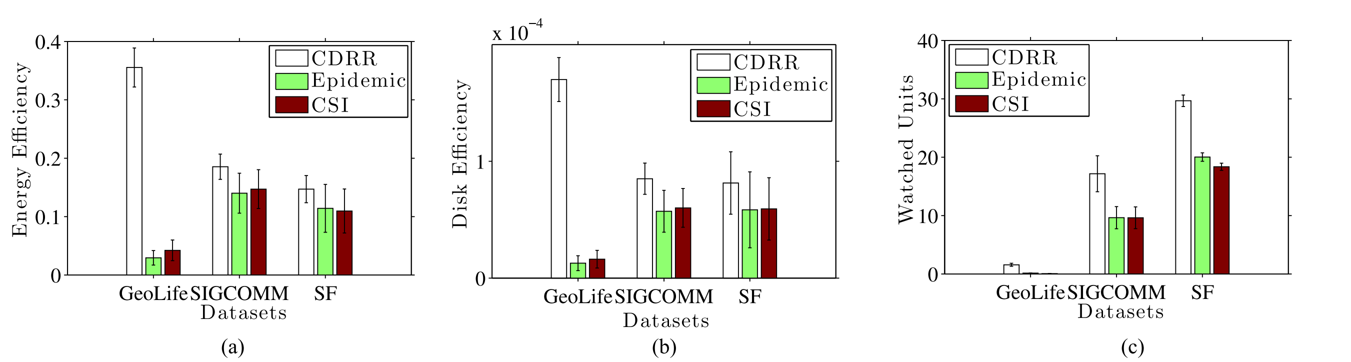

Fig. 8. Resource efficiency of our CDRR algorithm: (a) energy efficiency, (b) disk efficiency, and (c) watched units over three days.

users in user experience, watched user experience, and missed units among three datasets over 3 days. Fig. 7(a) and 7(b) show that our algorithm outperforms all other algorithms in terms of user experience and watched user experience. The gap of Epidemic and CSI to our algorithm is at least 20% in terms of user experience and watched user experience. This is because Epidemic and CSI do not take the characteristics of multi-layer news reports into considerations. Fig. 7(c) gives the number of missed units, which shows that Epidemic and CSI miss at least 10% and 11% more demanded units than our CDRR algorithm, respectively. Next, we plot sample empirical CDF (Cumulative Distribution Function) curves from SIGCOMM in Fig. 7(d)–7(f), which clearly show that our algorithm results in higher user experience and fewer missed units. In summary, Fig. 7

TABLE V
PERFORMANCE COMPARISONS OF CDRR OVER EPIDEMIC AND CSI

Metric	GeoLife		SIGCOMM		SF	
	Epidemic	CSI	Epidemic	CSI	Epidemic	CSI
User Experience	14X	26X	1.28X	1.38X	1.2X	1.21X
Watched User Experience	13X	15X	1.26X	1.33X	1.56X	1.19X
Missed Units	1.1X	1.11X	1.24X	1.26X	1.21X	1.25X
Watched Units	11X	26X	1.78X	1.78X	1.7X	1.76X
Energy Efficiency	13X	12X	1.33X	1.36X	1.33X	1.33X
Disk Efficiency	14X	15X	1.5X	1.52X	1.39X	1.41X

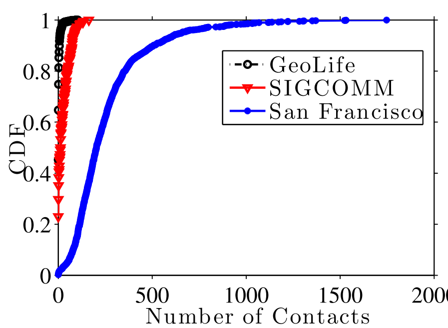

Fig. 9. Number of per-user contacts in three datasets.

demonstrates that our CDRR algorithm significantly improves the service quality.

3) The Proposed CDRR Algorithm is Resource Efficient: We report the resource efficiency of our proposed CDRR in Fig. 8. Fig. 8(a) and 8(b) present the energy efficiency and disk efficiency of our CDRR, which are the ratios between user experience and energy/disk consumption. The figures show that our CDRR algorithm is more energy-efficient than all other algorithms. In particular, at least 33% higher energy efficiency is observed compared to Epidemic and CSI algorithms. Moreover, the disk efficiency of our CDRR algorithm outperforms others by at least 39%. Fig. 8(a) and 8(b) reveal that CDRR delivers high service quality in a resource-efficient manner. Next, we plot the watched units in Fig. 8(c), which reveals that Epidemic and CSI suffer from lower watched units: at least 70% and 76%, compared to our CDRR algorithm. This partly explains why the CDRR algorithm is resource-efficient: it downloads more *useful* units. In summary, Fig. 8 shows that our CDRR algorithm is resource efficient.

4) Implications of Different Datasets: In Table V, we report the average performance comparisons between our CDRR algorithm and the other two baseline algorithms across all considered simulations. This table shows that our CDRR algorithm significantly outperforms others in all the aspects using all three datasets. A closer look reveals that our CDRR algorithm has its limitations with San Francisco dataset. We only outperform others by up to 21% in terms of user experience. This can be explained by Fig. 9, which shows the mobile users in San Francisco dataset are better connected, compared to GeoLife and SIGCOMM datasets. Since the number of contacts is very high, *any* distribution plans will work reasonably well. Such limitation is however not an issues, because typical challenged networks, especially those in developing countries and rural areas are not well-connected.

5) The Proposed CDRR Algorithm is Scalable: Next, we vary the disk budget, energy budget, and number of daily news reports to compare the performance of our CDRR algorithm against the other algorithms with SIGCOMM dataset in Fig. 10. Fig. 10(a) presents the user experience under different disk budgets, which shows that higher disk budgets lead to higher user experience with our CDRR algorithm, but it is not the case with Epidemic and CSI algorithms. This can be explained by Fig. 10(b), which shows that the CDRR algorithm utilizes higher disk budgets efficiently to reduce missed units. The other two algorithms, however, do not leverage the additional disk budgets. Compared with our CDRR algorithm, Epidemic and CSI miss 40% and 42% more units under 500 MB disk budget. Next, we plot the user experience under different energy budgets in Fig. 10(c). This figure shows that our CDRR algorithm capitalizes higher energy budget for better user experience, while CSI and Epidemic algorithms do not result in the same trend. This can be explained by Fig. 10(d), which reveals that the CDRR algorithm utilizes higher energy budgets efficiently to reduce the number of missed units. Compared with our CDRR algorithm, Epidemic and CSI algorithms miss 24% and 26% more units under 1500 J energy budget. Finally, we plot the user experience under different numbers of news reports in Fig. 10(e). This figure shows that more news reports lead to lower user experience. This can be explained by Fig. 10(f), which reveals that more news reports will degrade the user experience because of more missed units. However, our CDRR algorithm outperforms Epidemic and CSI by at least 17% and 27% under any number of news reports, respectively. In Fig. 10(f), compared to our CDRR algorithm, Epidemic and CSI algorithm miss up to 20% and 26% more units. In summary, Fig. 10 shows that our CDRR algorithm scales better with more resources and news reports, compared to other algorithms.

6) Effectiveness of Our Content Matcher and Content Predictor: We quantify the effectiveness of the machine-learning algorithms implemented in the Content Matcher and Content Predictor as follows. We augment our simulator to prohibit mobile users from requesting for any content that are not on their distribution plans, so as to focus on the impact of the machine learning algorithms. For comparisons, we assume perfect predictions using the user contact datasets, and refer to it as Oracle in the figures. We emphasize that Oracle is an impractical upper bound for benchmarking purpose only. We report the empirical CDF curves of user experience from the 3 datasets in Fig. 11. This figure reveals that our machine learning algorithms achieve similar performance with Oracle with SIGCOMM and San Francisco datasets: on average, only 26% and 17% gaps are observed, respectively. For the GeoLife dataset, our algorithms suffer from a larger gap of 69%, which can be attributed to more challenging scenarios as indicated by the inferior connectivity reported in Fig. 9. We note that, in such challenging scenarios, our CDRR algorithm significantly outperforms Epidemic and CSI as summarized in Table V.

VII. REAL IMPLEMENTATION

This section presents a complete prototype system[1] that distributes multimedia news reports to multiple users. We also compare our algorithm against others under real-life settings.

[1]Parts of the prototype system were presented in Hong *et al.* [75].

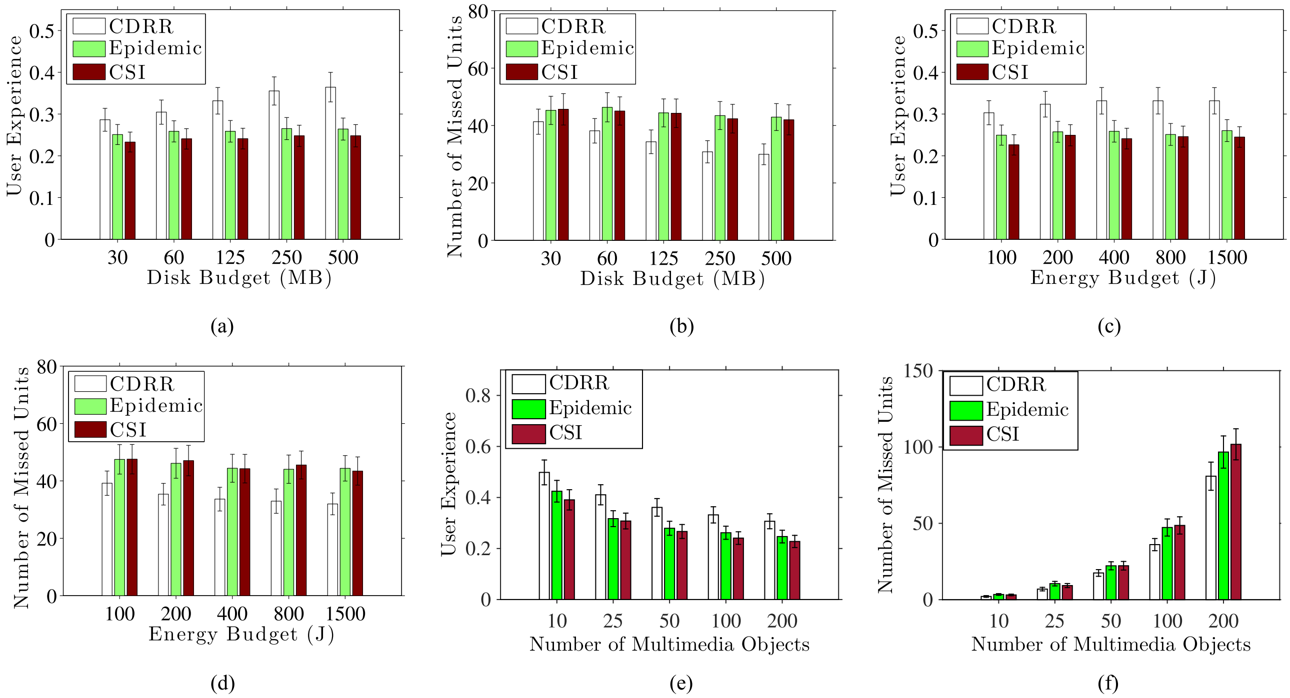

Fig. 10. Scalability of our CDRR algorithm under different resource budgets and number of multimedia objects: (a) user experience and (b) number of missed units with diverse disk budgets; (c) user experience and (d) number of missed units with diverse energy budgets; (e) user experience and (f) number of missed units with diverse number of news reports. Sample results from SIGCOMM dataset.

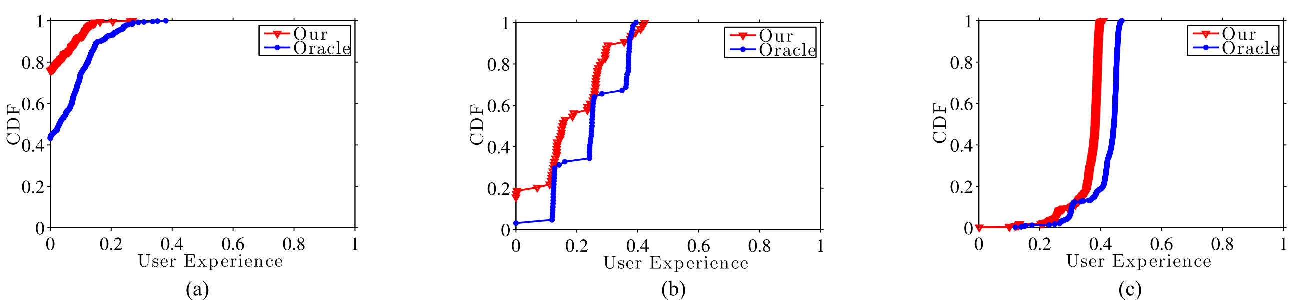

Fig. 11. Effectiveness of our content matcher and content predictor: (a) GeoLife, (b) SIGCOMM, and (c) San Francisco datasets.

A. Implementation on Linux and Android

We have implemented a complete testbed of the proposed mBridge system using Linux machines and Android mobile devices. The distribution server is built on a Linux workstation, and we realize our CDRR algorithm on it. Local proxies can be built on any Linux embedded devices and general-purpose computers. We adopt Raspberry PI for local proxies as shown in Fig. 12. Using Raspberry PI as the local proxy results in the following benefits: (i) smaller form factor, (ii) more cost effective, and (iii) easier deployments. The size of a Raspberry PI is only 100 cm^2. Each Raspberry PI, including a WiFi dongle, a case, a network cable, a memory card, and a power line, only costs about 55 US dollars at the time of writing. Compared to PCs, Raspberry PIs are easier to be transported to different places and countries.

The local proxies are connected to the distribution server via wired networks. We configure the local proxies to be WiFi access points, and program them to bridge the distribution server and mobile devices. We implement an mBridge Android app, which follows the distribution plan to download multimedia content whenever it runs into local proxies. It also records timestamped events and sends them via local proxies to the distribution server as profiles. The distribution server analyzes the profiles for various inputs, such as contacts, and computes distribution plans at 5 a.m. as a `cron` job. The computed plans are sent to mobile users via local proxies. To preserve user privacy, we anonymize the collected profiles, and allow mobile users to opt out from the data collection any time. We also allow users to configure several settings, such as the disk and energy budgets.

B. Experimental Setup

We set up a distribution server and eleven local proxies at four different locations: three different rural villages and a city (on our university campus). There are 15 users,[2] including university students, university employees, and farmers who use our

[2] In total, there are 31 users, but 16 of them decide to use our app without uploading their profiles.

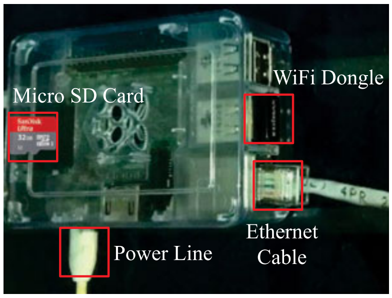

Fig. 12. Local proxy built on Raspberry PI.

Android app. In particular, they install our app from Google Play. The app comes with the default disk budget of 200 MB and the default energy budget of 80% battery capacity.

Every midnight, the distribution server automatically downloads the latest news reports, including text, audio, and videos from CNN, BBC, and Apple Daily. It transcodes the news videos to three resolutions: 240p, 360p, and 480p using `FFmpeg`. For each user, we use his/her profile collected in the past 7 days to predict his/her behavior. We then use the predicted behaviors to compute the distribution plans and send the plans to mobile users when they have the first contact with a local proxy. Our mobile app downloads news reports following the distribution plan, and the user may watch news reports anytime.

C. Experimental Scenarios

In the first two weeks of our experiment, the distribution server downloads 46 news reports on average every day, which is equivalent to 900+ MB total size. On average, each user spends 2.8 hours within the coverage of local proxies everyday. The average number of contacts of each user in each day is 7.8, the average contact duration is 101 seconds, and each user moves 3.6 km on average everyday. The distance between the university to the villages is 38.8 km and there are no users commuting between the campus and the villages. The size of the university and the villages are about 1.2 km^2 and 4 km^2, respectively. There are 67% and 87% of users using the default disk and energy budgets, respectively. With the default disk budget, each user can download news reports for up to about three layers. We note that when the app runs out of the disk budget, it pops up a dialog reminding the user about the possibility of increasing the disk budget for better video quality. We observe that more than 2/3 of users stick with the default disk budget, which shows that many users are satisfied with the low-resolution videos. This is inline with our observation of user experience improvements decreasing along with the number of layers in Section IV.

D. Emulation Setup

One limitation of our deployed mBridge testbed is that the distribution server can only execute a *single* distribution planning algorithm at a time and we only implement our CDRR algorithm in the testbed. To compare our algorithm against the baseline algorithms, we use the collected profiles to drive our simulator multiple times, and save the *distribution plans* generated by different algorithms. We then set up an *emulation testbed*, which consists of a Linux FTP server, a smartphone, and a power meter, so as to measure the *actual* energy consumption resulted by different algorithms. We configure the FTP server to limit the bandwidth following the traces in Friedman *et al.* [71]. We then augment our mobile app to download the multimedia news reports following the distribution plans. We run the mobile app on a Samsung Galaxy J7 smartphone connected to an Agilent 66321D power meter. The power meter is configured to serve as a power source at 3.85 V, and connected to a PC via an USB cable. We record the voltage and current at 200 Hz. The records are then used to calculate the energy consumption.

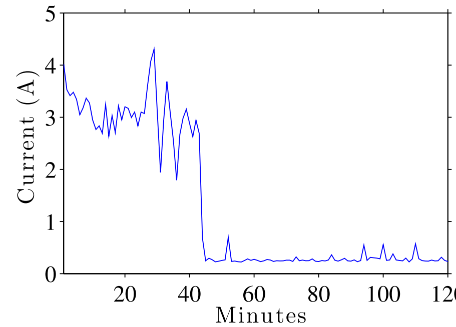

Fig. 13. Sample measured current levels from user 1.

TABLE VI
STATISTICS OF DAILY COMMUNICATION ENERGY CONSUMPTION (J) OVER A ONE-WEEK EMULATION

	CDRR		CSI		Epidemic	
Total Energy Consumption						
	Mean	Std	Mean	Std	Mean	Std
User 1	79.5	21.7	82.2	28.7	79.1	19.8
User 2	99.1	23.6	100.2	24.8	91.4	18.6
User 3	55.5	4.1	50.4	2.6	50.2	7.3
Per MB Energy Consumption						
	CDRR		CSI		Epidemic	
	Mean	Std	Mean	Std	Mean	Std
User 1	0.46	0.04	0.46	0.1	0.44	0.02
User 2	0.45	0.04	0.44	0.04	0.43	0.01
User 3	0.43	0.02	0.40	0.01	0.41	0.02

In addition, we use the distribution plans from the diverse algorithms to conduct a user study for real user experience. Questionnaires, similar to the ones used in Section IV, are prepared for the user study. Running 2-week emulations with 3 different algorithms for all 15 users is time consuming without revealing too many insights. For example, for a less active user, he/she doesn't have too many contacts, and none of the algorithms work for him/her. Therefore, we focus on top three active users and their top seven active days for a 1-week emulation, which is equivalent to 21 days in totals. We repeat each day of emulation with three algorithms, resulting in 63 1-day news reports. We recruit 63 participants, among them 70% are males and the average age is 24 years old. To avoid overloading the participants, each participant watches 1-day news report and fills in the questionnaire via our Web interface shown in Fig. 3. Each 1-day news report takes a participant about 10-40 mins. The participants then fill the questionnaire to give MOS scores.

E. Emulation Results

1) Our Implementation is Energy Efficient: Fig. 13 shows 2-hour sample measurements of the current from user 1. In

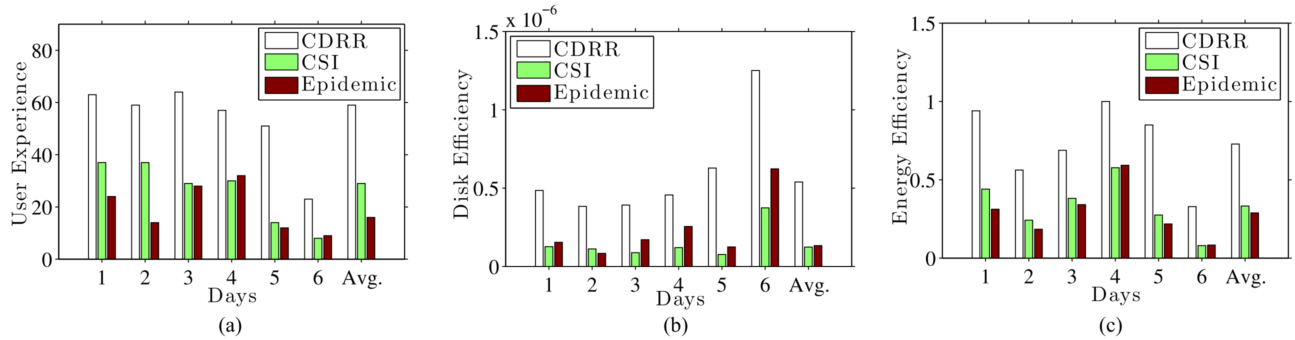

Fig. 14. Better service quality and resource efficiency of our CDRR algorithm: (a) quality improvement, (b) disk efficiency, and (c) energy efficiency over a one-week emulation. Sample results from user 1 are shown.

TABLE VII
PERFORMANCE IMPROVEMENTS OF CDRR OVER EPIDEMIC AND CSI

Metric	User 1		User 2		User 3	
	CSI	Epidemic	CSI	Epidemic	CSI	Epidemic
User Experience	128%	206%	89%	66%	143%	65%
Disk Efficiency	349%	43%	42%	143%	69%	472%
Energy Efficiency	148%	188%	129%	87%	177%	77%

this figure, the user's smartphone downloads news reports for 43 minutes before being idling. We compute the communication energy consumption by deducting the idling energy consumption from the total energy consumption. We report the mean and standard deviation of daily energy consumption resulted by different algorithms in Table VI. This table shows that completing a 1-day distribution plan only consumes up to 153 J, which is 0.3% of the battery capacity (3300 mAh, 3.85 V). We also report the energy consumption per MB in the same table. It shows that the gaps among different algorithms is at most 7%, which is insignificant.

2) Our CDRR Algorithm Leads to Better Service Quality and Resource Efficiency: Although different distribution planning algorithms result in similar energy consumption, they may lead to diverse user experience. We plot the sample user experience from user 1 in Fig. 14(a), which shows that our CDRR algorithm outperforms CSI and Epidemic by 1.1 and 2.7 times on average in terms of user experience. Fig. 14(b) and 14(c) report the sample disk efficiency and energy efficiency from user 1. In terms of disk efficiency, our CDRR algorithm outperforms CSI and Epidemic by 3.4 and 4.1 times; in terms of energy efficiency, our CDRR algorithm outperforms them by 1.2 and 1.5 times. In Table VII, we give the average performance improvement of our CDRR algorithm over the other two baseline algorithms for individual users. Among the three users, our CDRR algorithms outperforms others by up to 206%, 472%, and 188% in terms of user experience, disk efficiency, and energy efficiency.

VIII. CONCLUSION

In this article, we studied the problem of distributing multimedia content over challenged networks to mobile devices. We proposed the mBridge system, which carefully plans the distribution of multimedia content to mobile users. The critical component of mBridge is the distribution planning algorithm, which intelligently distributes multi-layer multimedia objects over challenged networks using opportunistic communications. Our CDRR achieves near-optimal results in terms of user experience, despite the complexity of the distribution planning problem (NP-Complete). We conducted extensive simulations using real datasets. The simulation results indicate that our proposed CDRR algorithm results in: (i) better user experience, which outperforms other algorithms by at least 20%, (ii) higher energy and disk efficiency, which outperforms other algorithms by at least 33% and 39%, respectively, (iii) fewer missed units which outperforms other algorithms by at least 10%, and (iv) more watched units, which are at least 70% more than other algorithms. In addition, we implemented and deployed a prototype testbed in a university and three rural villages. Experiments reveal that our mBridge system outperforms the baseline algorithms by up to 206%, 472% and 188% in terms of user experience, disk efficiency, and energy efficiency, respectively. The mBridge prototype implementation can be improved in multiple directions. For example, the mobile app does not support ad-hoc connectivity. This does not have clear impacts in our experiments, because of the limited number of ad-hoc contacts. Nonetheless, our simulation results show that our mobile app *will* work even better once the ad-hoc connectivity is implemented.

ACKNOWLEDGMENT

The authors would like to thank Y.-C. Lin and S.-T. Wang for their contributions to their preliminary implementation.

REFERENCES

[1] "Cisco visual networking index: Forecast and methodology," Jun. 2016. [Online]. Available: http://tinyurl.com/zzt4chh
[2] "ITU ICT facts and figures the world in 2015, " May 2015. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx
[3] "How do we accelerate Internet access in Africa?," 2013. [Online]. Available: http://tinyurl.com/q9ksjwg
[4] "Social, digital and mobile in India," 2014. [Online]. Available: http://tinyurl.com/nqptj6m
[5] "It's time to take a closer look at China's mobile industry," 2014. [Online]. Available: http://tinyurl.com/pgx6q94
[6] V. Mota, F. Cunha, D. Macedo, J. Nogueira, and A. Loureiro, "Protocols, mobility models and tools in opportunistic networks: A survey," *Comput. Commun.*, vol. 48, pp. 5–19, 2014.
[7] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, 2003, pp. 27–34 .
[8] L. Chen, Y. Zhou, and D. Chiu, "Smart streaming for online video services," *IEEE Trans. Multimedia*, vol. 17, no. 4, pp. 485–497, Apr. 2015.

[9] Y. Hu, D. Niu, and Z. Li, "A geometric approach to server selection for interactive video streaming," *IEEE Trans. Multimedia*, vol. 18, no. 5, pp. 840–851, May 2016.
[10] R. Guruprasad and S. Dey, "Battery aware video delivery techniques using rate adaptation and base station reconfiguration," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1630–1645, Sep. 2015.
[11] Y. Go, O. Kwon, and H. Song, "An energy-efficient HTTP adaptive video streaming with networking cost constraint over heterogeneous wireless networks," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1646–1657, Sep. 2015.
[12] H. Ntareme, M. Zennaro, and B. Pehrson, "Delay tolerant network on smartphones: Applications for communication challenged areas," in *Proc. Extreme Conf. Commun.*, Manaus, Brazil, 2011, Art. no. 14.
[13] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, "Cooperative caching for efficient data access in disruption tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 3, pp. 611–625, Mar. 2014.
[14] "DTN2," 2012. [Online]. Available: http://sourceforge.net/projects/dtn/files/DTN2/dtn-2.9.0/
[15] "IBR-DTN," 2012. [Online]. Available: http://trac.ibr.cs.tu-bs.de/project-cm-2012-ibrdtn
[16] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep., 2000. [Online]. Available: http://issg.cs.duke.edu/epidemic/epidemic.pdf
[17] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. ACM MobiHoc*, Tokyo, Japan, 2004, pp. 187–198.
[18] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, 2011.
[19] A. Mtibaa and K. A. Harras, "CAF: Community aware framework for large scale mobile opportunistic networks," *Comput. Commun.*, vol. 36, no. 2, pp. 180–190, 2013.
[20] A. Mtibaa and K. Harras, "Social forwarding in large scale networks: Insights based on real trace analysis," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, 2011, pp. 1–8.
[21] K. Harras and K. Almeroth, "Controlled flooding in disconnected sparse mobile networks," *Wireless Commun. Mobile Comput.*, vol. 9, no. 1, pp. 21–33, 2009.
[22] A. Mtibaa and K. A. Harras, "Select&spray: Towards deployable opportunistic communication in large scale networks," in *Proc. ACM Int. Symp. Mobility Manage. Wireless Access*, 2013, pp. 1–8.
[23] P. Cheng, K. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Netw. Appl.*, vol. 15, no. 1, pp. 61–82, 2010.
[24] B. Burns, O. Brock, and B. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proc. IEEE Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Miami, FL, USA, vol. 1, Mar. 2005, pp. 398–408.
[25] A. Al Jarhi, H. A. Arafa, K. A. Harras, and S. G. Aly, "Rethinking opportunistic routing using space syntax," in *Proc. ACM Workshop Challenged Netw.*, pp. 21–26, 2011. [Online]. Available: http://doi.acm.org/10.1145/2030652.2030660
[26] A. Mtibaa and K. A. Harras, "Exploiting space syntax for deployable mobile opportunistic networking," in *Proc. IEEE Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Oct. 2013, pp. 533–541.
[27] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Barcelona, Spain, Apr. 2006, pp. 1–11.
[28] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "A performance comparison of delay-tolerant network routing protocols," *IEEE Trans. Netw.*, vol. 30, no. 2, pp. 46–53, Mar./Apr. 2016.
[29] W. Hsu, D. Dutta, and A. Helmy, "CSI: A paradigm for behavior-oriented profile-cast services in mobile networks," *Ad Hoc Netw.*, vol. 10, no. 8, pp. 1586–1602, 2012.
[30] E. Pagani, L. Valerio, and G. Rossi, "Weak social ties improve content delivery in behavior-aware opportunistic networks," *Ad Hoc Netw.*, vol. 25, pp. 314–329, 2015.
[31] N. Changuel, B. Sayadi, and M. Kieffer, "Control of distributed servers for quality-fair delivery of multiple video streams," in *Proc. ACM Int. Conf. Multimedia*, Nara, Japan, 2012, pp. 269–278.
[32] S. Kang and M. Mutka, "Efficient mobile access to internet data via a wireless peer-to-peer network," in *Proc. IEEE Annu. Conf. Pervasive Comput. Commun.*, Orlando, FL, USA, Mar. 2004, pp. 197–205.
[33] Z. Xiang, Q. Zhang, W. Zhu, Z. Zhang, and Y. Zhang, "Peer-to-peer based multimedia distribution service," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 343–355, 2004.
[34] K. Mokhtarian and M. Hefeeda, "Capacity management of seed servers in peer-to-peer streaming systems with scalable video streams," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 181–194, Jan. 2013.
[35] L. Zhou, "Mobile device-to-device video distribution: Theory and application," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 3, 2016, Art. no. 38.
[36] A. Zhang, J. Chen, L. Zhou, and S. Yu, "Graph theory-based QoE-Driven cooperation stimulation for content dissemination in Device-to-Device Communication," *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 4, pp. 556–567, Oct.–Dec. 2016.
[37] A. Zhang, L. Wang, and L. Zhou, "Location-based distributed caching for device-to-device communications underlaying cellular networks," *Wireless Commun. Mobile Comput.*, vol. 16, pp. 1859–1875, 2015.
[38] G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling Xunlei Kankan: Understanding hybrid CDN-P2P video-on-demand streaming," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 229–242, Feb. 2015.
[39] L. Keller *et al.*, "MicroCast: Cooperative video streaming on smartphones," in *Proc. ACM MobiSys*, Low Wood Bay, U.K., 2012, pp. 57–70.
[40] H. Hanano, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito, "Video ads dissemination through WiFi-cellular hybrid networks," in *Proc. IEEE Annu. Conf. Pervasive Comput. Commun.*, Galveston, TX, USA, Mar. 2009, pp. 1–6.
[41] N. Do, C. Hsu, and N. Venkatasubramanian, "Video dissemination over hybrid cellular and ad hoc networks," *IEEE/ACM Trans. Mobile Comput.*, vol. 13, no. 2, pp. 274–286, Nov. 2014.
[42] N. Do, C. Hsu, and N. Venkatasubramanian, "Hybcast: Rich content dissemination in hybrid cellular and 802.11 ad hoc networks," in *Proc. IEEE Symp. Rel. Distrib. Syst.*, Irvine, CA, USA, Oct. 2012, pp. 352–361.
[43] Y. Nimmagadda, K. Kumar, and Y. Lu, "Adaptation of multimedia presentations for different display sizes in the presence of preferences and temporal constraints," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 650–664, Nov. 2010.
[44] R. Mohan, J. Smith, and C. Li, "Adapting multimedia internet content for universal access," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 104–114, Mar. 1999.
[45] Y. Wang, J. Kim, S. Chang, and H. Kim, "Utility-Based video adaptation for universal multimedia access (UMA) and content-based utility function prediction for real-time video transcoding," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 213–220, Feb. 2007.
[46] X. Wu, J. Yang, Y. Ran, and H. Xi, "Adaptive scalable video transmission strategy in energy harvesting communication system," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2345–2353, Dec. 2015.
[47] F. Qian *et al.*, "Web caching on smartphones: Ideal vs. reality," in *Proc. ACM MobiSys*, Low Wood Bay, U.K., 2012, pp. 127–140.
[48] D. Lymberopoulos, O. Riva, K. Strauss, A. Mittal, and A. Ntoulas, "Pocketweb: Instant web browsing for mobile devices," in *Proc. ACM ASPLOS*, London, U.K., 2012, pp. 1–12.
[49] Y. Zhang, C. Tan, and L. Qun, "Cachekeeper: A system-wide web caching service for smartphones," in *Proc. ACM UbiComp*, Zurich, Switzerland, 2013, pp. 265–274.
[50] T. Wang, P. Hui, S. Kulkarni, and P. Cuff, "Cooperative caching based on file popularity ranking in delay tolerant networks," in *Proc. Extreme Conf. Commun.*, Zurich, Switzerland, 2012, pp. 1–6.
[51] S. Isaacman and M. Martonosi, "Low-infrastructure methods to improve Internet access for mobile users in emerging regions," in *Proc. Int. Conf. Companion World Wide Web*, Hyderabad, India, 2011, pp. 473–482.
[52] "Topia's term extractor," 2015. [Online]. Available: https://pypi.python.org/pypi/topia.termextract/
[53] J. Liu, P. Dolan, and E. Pedersen, "Personalized news recommendation based on click behavior," in *Proc. ACM Int. Conf. Intell. User Interfaces*, Hong Kong, China, 2010, pp. 31–40.
[54] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. ACM Int. Conf. Knowledge Discovery Data Mining*, Edmonton, AB, Canada, 2002, pp. 133–142.
[55] C. Burges, "From Ranknet to LambdaRank to LambdaMART: An overview," Microsoft Res., Cambridge, U.K., Tech. Rep. MSR-TR-2010-82, 2010.
[56] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext: A location predictor on trajectory pattern mining," in *Proc. ACM Int. Conf. Knowledge Discovery Data Mining*, Paris, France, 2009, pp. 637–646.

[57] E. Cho, S. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. ACM Int. Conf. Knowledge Discovery Data Mining*, San Diego, CA, USA, 2011, pp. 1082–1090.

[58] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye, "Mining periodic behaviors for moving objects," in *Proc. ACM Int. Conf. Knowledge Discovery Data Mining*, Washington, DC, USA, 2010, pp. 1099–1108.

[59] M. Gonzalez, C. Hidalgo, and A. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, pp. 779–782, 2008.

[60] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[61] J. Boyce, Y. Ye, J. Chen, and A. Ramasubramonian, "Overview of SHVC: Scalable extensions of the high efficiency video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 20–34, Jan. 2016.

[62] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *Proc. ACM SIGCOMM*, 2013, pp. 3–14.

[63] "Bountry workers (in Chinese)," 2017. [Online]. Available: http://bountyworkers.net/

[64] Y. Shen, C. Jiang, Q. Quek, and Y. Ren, "Device-to-device-assisted communications in cellular networks: An energy efficient approach in downlink video sharing scenario," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1575–1587, Feb. 2016.

[65] J. Wang, C. Jiang, Z. Bie, Q. Quek, and Y. Ren, "Mobile data transactions in device-to-device communication networks: Pricing and auction," *IEEE Wireless Commun. Lett.*, vol. 5, no. 3, pp. 300–303, Jun. 2016.

[66] Y. Shen, C. Jiang, T. Quek, and Y. Ren, "Location-aware device communication design: Exploration and exploitation on energy," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 46–52, Apr. 2016.

[67] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Ma, "Understanding mobility based on GPS data," in *Proc. ACM UbiComp*, Seoul, South Korea, 2008, pp. 312–321.

[68] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A parsimonious model of mobile partitioned networks with clustering," in *Proc. IEEE Int. Commun. Syst. Netw. Workshops*, Bangalore, India, Jan. 2009, pp. 1–10.

[69] A. Pietilainen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "MobiClique: Middleware for mobile social networking," in *Proc. ACM WOSN*, Barcelona, Spain, 2009, pp. 49–54.

[70] S. Wang, C. Fan, Y. Huang, and C. Hsu, "Toward optimal crowdsensing video quality for wearable cameras in smart cities," in *Proc. IEEE SmartCity*, Hong Kong, China, Apr.-May 2015, pp. 624–629.

[71] R. Friedman, A. Kogan, and Y. Krivolapov, "On power and throughput tradeoffs of WiFi and Bluetooth in smartphones," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 1363–1375, Jul. 2013.

[72] "LETOR 4.0 dataset," 2009. [Online]. Available: http://research.microsoft.com/en-us/um/beijing/projects/letor/letor4dataset.aspx

[73] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proc. ACM Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006, pp. 377–384.

[74] "Best-selling smartphones in South Africa," [Online]. Available: https://mybroadband.co.za/news/smartphones/89031-best-selling-smartphones-in-south-africa.html

[75] H. Hong *et al.*, "Challenged content delivery network: Eliminating the digital divide," in *Proc. ACM Int. Conf. Multimedia*, Brisbane, Australia, 2015, pp. 799–800.

Authors' photographs and biographies not available at the time of publication.