Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

# Measurement and analysis of content diffusion characteristics in opportunity environments with Spark*

Xiao-hong ZHANG†1, Kai QIAN[1], Jian-ji REN†‡1, Zong-pu JIA[1], Tian-peng JIANG[2], Quan ZHANG[3]

[1]*College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China*

[2]*Beijing Anqi Zhilian Technology Co., Ltd., Beijing 100000, China*

[3]*Department of Computer Science, Wayne State University, Detroit 48202, USA*

†E-mail: xh.zhang@hpu.edu.cn; renjianji@hpu.edu.cn

Received Mar. 11, 2019; Revision accepted Apr. 18, 2019; Crosschecked Oct. 10, 2019

**Abstract:** Opportunity networks provide a chance to offload the tremendous cellular traffic generated by sharing popular content on mobile networks. Analyzing the content spread characteristics in real opportunity environments can discover important clues for traffic offloading decision making. However, relevant published work is very limited since it is not easy to collect data from real environments. In this study, we elaborate the analysis on the dataset collected from a real opportunity environment formed by the users of Xender, which is one of the leading mobile applications for content sharing. To discover content transmission characteristics, scale, speed, and type analyses are implemented on the dataset. The analysis results show that file transmission has obvious periodicity, that only a very small fraction of files spread widely, and that application files have much higher probability to be popular than other files. We also propose a solution to maximize file spread scales, which is very helpful for forecasting popular files. The experimental results verify the effectiveness and usefulness of our solution.

**Key words:** Content dissemination; Device-to-device communication; Opportunity network; Linear threshold model

https://doi.org/10.1631/FITEE.1900137 **CLC number:** TP393.0

## 1 Introduction

Social networks have attracted tremendous attention since their emergence. The rise of online services such as Facebook (Obar and Wildman, 2015), WeChat (Che and Cao, 2014), and Sina microblog (Guan et al., 2013) has significantly increased the frequency of users' online activities. Specifically, with the rapid growth of mobile technologies, mobile devices have become the preferred tools to access social networks. A very large amount of content is continuously uploaded from or downloaded to mobile devices with high frequency, which has resulted in the explosive growth of traffic loads on mobile networks. However, due to limited wireless capacity, it is difficult for mobile network operators to efficiently handle those loads, especially in areas with high user density (Andreev et al., 2014).

Fortunately, opportunity networks in which contents spread depending on human mobility and device-to-device (D2D) communication have shown the ability to offload these huge traffic loads (Ioannidis et al., 2009; Pietiläinen and Diot, 2012; Wang et al., 2014). Since most communication is generated by downloading popular content such as photos, videos, and mobile applications (Cha et al., 2007;

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61433012 and 61602156), the Project of Science and Technology in Henan Province, China (No. 142102210435), the Project of the Basic and Frontier Technology in Henan Province, China (No. 142300410147), and the PhD Foundation of Henan Polytechnic University, China (No. B2012-099)

ORCID: Xiao-hong ZHANG, http://orcid.org/0000-0003-4720-4775

Cisco, 2017), opportunity networks can release cellular networks by transmitting those popular contents through D2D links temporarily built between mobile users. A lot of effort has gone into offloading these huge loads by exploiting opportunity communications among mobile users. Some solutions focus on selecting ideal users (Lu et al., 2014; Thilakarathna et al., 2014; Zhao and Song, 2016) or best groups (Thilakarathna et al., 2012) to start content transmission, while others concentrate on designing data transmission protocols (Jiang N et al., 2016). In some work, temporal factors (Gao et al., 2016) and users' preferences (Lin et al., 2012) are also exploited to perform traffic offloading. However, most of these solutions are based on unrealistic assumptions or limited data volume. Although the characteristics of content transmission in real opportunity networks are very valuable, there has been relatively little work on them.

In this study, we perform an analysis on the dataset collected from a real opportunity environment formed by the users of Xender, which is one of the leading mobile applications (Apps) for content sharing. We aim to discover the content transmission characteristics in opportunity environments through scale, speed, and type analyses. The analysis results show that file transmission has obvious periodicity, that only a very small fraction of files spread widely, and that application files have much higher probability of being popular than other files. We also propose a solution based on the linear threshold (LT) model (Kempe et al., 2003) to maximize file spread scales, which is very important for forecasting popular files.

## 2 Related work

Opportunity networks spread contents depending on human mobility and D2D communication. Many efforts have been devoted to exploit opportunity networks to offload the explosive traffic on mobile networks (Zhang Y et al., 2015; Tang, 2017; Wang et al., 2018). Thilakarathna et al. (2012, 2014) discovered that contents can be propagated to almost all users by properly selecting a small number of users to replicate contents, which is still the case even with required time bounds (Wang et al., 2016). Based on the discovery, Chuang and Lin (2012) and Wang et al. (2014) proposed to choose special users as initial seeds for content transmission according to factors such as social impact, mobility patterns, and communities, while Cheng et al. (2015) and Tian et al. (2016) concentrated on the selection of relaying users according to users' popularity or movements. Pietiläinen and Diot (2012) showed that the users contacting with high frequency contribute less to content dissemination when temporality is considered.

To reduce data communication latency between mobile users, Jiang N et al. (2016) designed some protocols for data dissemination. Gao et al. (2016) designed a special solution to offload deadline-sensitive data. Lin et al. (2012) aimed to handle users' preferences, while Mashhadi et al. (2012) and Rebecchi et al. (2016) emphasized load balancing during content dissemination and the reward need of users' forwarding contents, respectively. There also exist some special approaches (Zhang S et al., 2015; Bao et al., 2016; Jiang J et al., 2016) which highlight traffic offloading by cache policies. Although the above solutions are meaningful, they share the same limitation that they lack the support of real and large-scale opportunity environments.

Wang et al. (2017a, 2017b) also performed analysis on the dataset from Xender and excavated some valuable knowledge, e.g., the knowledge about user behavior and social characters, which concentrated on users' behavior. In this study, we focus on content dissemination and aim to excavate content transmission characteristics. This is quite different from Wang et al. (2017a, 2017b).

## 3 Details of datasets and the platform

Xender is a world-wide popular application for content sharing over D2D. It provides delivery services for various types of content across multiple mobile platforms without the support of cellular networks. Concretely, it implements peer-to-peer content dissemination via WiFi, which does not increase any charge of cellular traffic. Xender has been widely used in the world. It has more than 80 million active users globally covering all time zones.

Whenever a file is shared, a record is logged in the device that receives that file. When the Internet is available, all the records are uploaded to Xender servers. We collect the records of a whole month and try to excavate file spread characteristics from them. The collected records involve more than 800

million files and accumulate to more than 500 GB. As described in Wang et al. (2017b), those records consist of more than 20 attributes. In this study, we pay attention only to five attributes: content name, content type, sender identity document (ID), receiver ID, and receiver write time.

To analyze this very large dataset, a big data analysis platform has been set up. The platform consists of three layers: hardware, system, and application layers. The hardware layer comprises 20 nodes, which are connected by a gigabit ethernet switch. The system layer provides storage and computing services. It takes the Hadoop distributed file system (HDFS) as a storage subsystem and Spark as a computing subsystem. The Scala programs we have developed to perform data analysis comprise the application layer. Table 1 shows the details of the platform. For convenience of description, all notations used in the study are described in Table 2.

We use graphs to express the spread procedures of files and name these graphs "spread graphs." In Xender, file sharing activity happens only among adjacent users. Whenever a file is transmitted from one user to another, an edge is created to the spread graph of that file. For example, if $u_i$ shares one file $f$ to $u_j$, an edge is drawn from $u_i$ to $u_j$ in $\boldsymbol{G}_f$, which is the spread graph of $f$. With the diffusion of $f$, more edges are generated to $\boldsymbol{G}_f$. Finally, the total number of edges in $\boldsymbol{G}_f$ is used to describe the spread scale of $f$. Fig. 1 shows the dissemination procedure and the spread graph of $f$. According to Fig. 1, the spread scale of $f$ is 13.

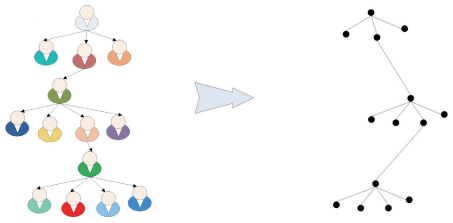

**Fig. 1 File spread graph**

## 4 File spread analysis

Understanding the file dissemination characteristics in real and large-scale opportunity environments is critical for offloading traffic efficiently. We elaborate the dissemination characteristic analysis on the dataset collected from Xender in this section.

**Table 1 Configuration of the big data platform**

| Layer | | Configuration | Value |
|---|---|---|---|
| Hardware layer | | Number of cores per node | 24 |
| | | Memory space per node | 32 GB |
| | | Disk space per node | 3 TB |
| | | Total number of nodes | 20 |
| System layer | HDFS | Block size | 256 MB |
| | | Number of replicas | 3 |
| | | Number of data | 19 |
| | | Number of nodes | 19 |
| | Spark | Number of tasks per executor | 7 |
| | | Number of worker nodes | 19 |
| | | Number of executors per node | 1 |
| | | Number of workers per node | 1 |

**Table 2 Notations**

| Notation | Description |
|---|---|
| $u$ | User $u_i$ represents the $i^{\text{th}}$ user |
| $v$ | Vertex $v_i$ represents the $i^{\text{th}}$ vertex, and each vertex corresponds to a user |
| $\boldsymbol{V}$ | Vertex set $\boldsymbol{V}_f$ is the set corresponding to file $f$ |
| $e$ | Edge $e_{ij}$ expresses the edge drawn from $v_i$ to $v_j$ |
| $\boldsymbol{E}$ | Edge set $\boldsymbol{E}_f$ is the set corresponding to $f$ |
| $\boldsymbol{G}$ | Spread graph $\boldsymbol{G}_f$ denotes the spread graph of $f$, and $\boldsymbol{G}_f =< \boldsymbol{V}_f, \boldsymbol{E}_f >$ |
| $\boldsymbol{N}$ | Neighbor node set $\boldsymbol{N}_i$ describes the neighbor node set of $v_i$ |
| $F_{\text{scale}}(s)$ | Total number of files in the same scale $s$ |
| $\text{factor}_{\text{degree}}$ | Degree factor $\text{factor}_{\text{degree}}(i)$ expresses the degree factor of $v_i$ |
| $\text{factor}_{\text{seed}}$ | Seed factor $\text{factor}_{\text{seed}}(i)$ describes the seed factor of $v_i$ |
| $\text{degree}_{\text{out}}$ | Outdegree $\text{degree}_{\text{out}}(i)$ signifies the outdegree of $v_i$ |
| $\text{degree}_{\text{in}}$ | Indegree $\text{degree}_{\text{in}}(i)$ denotes the indegree of $v_i$ |
| closeness | Closeness $\text{closeness}(i,j)$ describes the closeness between $v_i$ and $v_j$ |
| Inf | Influence $\text{Inf}(i)$ describes the influence from the neighbors of $v_i$ |
| activeness | Activeness $\text{activeness}(i)$ depicts the activeness of $v_i$ |

### 4.1 Scale analysis

The spread scale of each file is determined by the number of edges in the spread graph of that file. To construct spread graphs for files efficiently, a series of Scala programs are developed to partition all records by files. The scale of each file can be captured by calculating the number of related records.

We form each scale $s$ and its $F_{\text{scale}}(s)$ into tuples described as $< s, F_{\text{scale}}(s) >$ and sort these tuples in ascending order of $s$. Fig. 2 shows the sorting results. In Fig. 2, the $x$ and $y$ axes represent all the possible

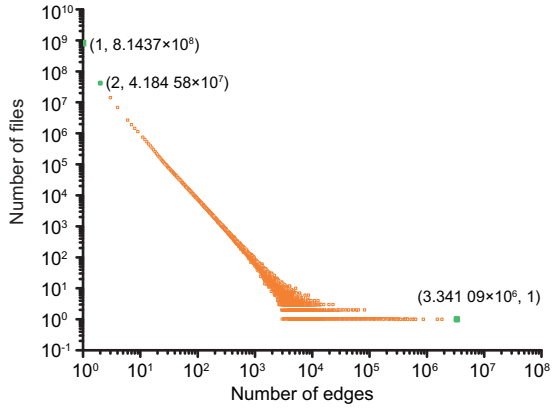


**Fig. 2 Distribution of $F_{\mathrm{scale}}(s)$**

values of $s$ and $F_{\mathrm{scale}}(s)$, respectively. According to the figure, the distribution of $F_{\mathrm{scale}}(s)$ has a strong property of power law. The vast majority of files spread in relatively small scales. Only a very small proportion of files spread in large scales. Similar to the very small number of very important people who have strong influence in social networks, the files transmitted most widely have the strongest impact in the opportunity environments.

We find that almost 90% of files are spread with only one edge, and that about 99% of them are transmitted with no more than 100 edges. These discoveries are very important for system optimization, especially for the capacity design of file cache servers. Based on these results, cache servers need only to keep the replicas of very few files.

According to spread scales, files are partitioned into seven classes: tiny, small, small-medium, medium, medium-large, large, and huge files. The scales of different classes of files are distributed as follows: [1, 9] for tiny files (T), [10, 99] for small files (S), [100, 999] for small-medium files (SM), [1000, 9999] for medium files (M), [10 000, 99 999] for medium-large files (ML), [100 000, 999 999] for large files (L), and [1 000 000, $+\infty$) for huge files (H). Tiny files account for about 91% of all files. Huge files spread in the largest scales among all files. However, their total number is a lot smaller than that of any other class. In this study, popular files are those files that spread most widely; in other words, popular files are huge files.

### 4.2 Speed analysis

In the Xender dataset, all sharing activities of files are recorded with timestamps. Those timestamps are very important since they can reveal the temporal characteristics of file diffusion and Xender usage. All those timestamps are stored in the attribute named "receiver write time." Hence, the temporal characteristics can be excavated by analyzing the attributes. However, a large number of timestamps are invalid since they are not generated in the expected period. This happens because the system clocks on a non-negligible number of mobile devices are not proofread correctly. In this subsection, all the records including invalid timestamps are excluded.

We calculate the edges generated for files in different times and depict the results in Fig. 3. According to Fig. 3, all files share similar spread trend regardless of spread scales. They all spread to peak from 10:00 to 12:00 p.m. when people stay with family while drop to a trough around 7 am. It is obvious that all files spread with a strong periodicity, which is consistent with the discovery described in Wang et al. (2017b). Moreover, about 27% of spread activities take place during working time, while 73% of them happen in rest time. It means that families and friends use Xender to share files more often than colleagues.

We also analyze file spread speeds which are calculated according to the edges generated at different times. All speeds are grouped by week (Fig. 4). It is obvious that files diffuse at different speeds in different weeks. Concretely, they diffuse at the lowest average speeds in the first week while at the highest average speeds in the last week. Even in the same week, files spread at growing average speeds day by day. It can be observed easily that the daily peak speeds and the daily valley speeds of different classes of files are continuously growing week by week. This

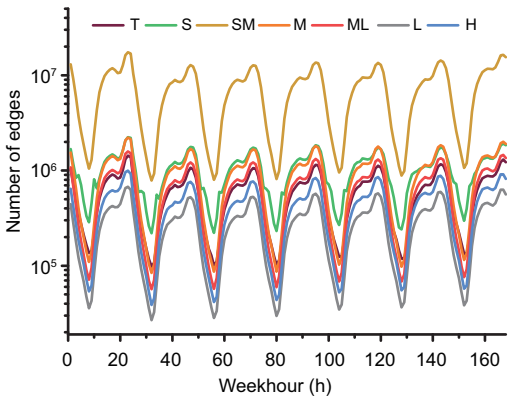


**Fig. 3 Temporal characters**

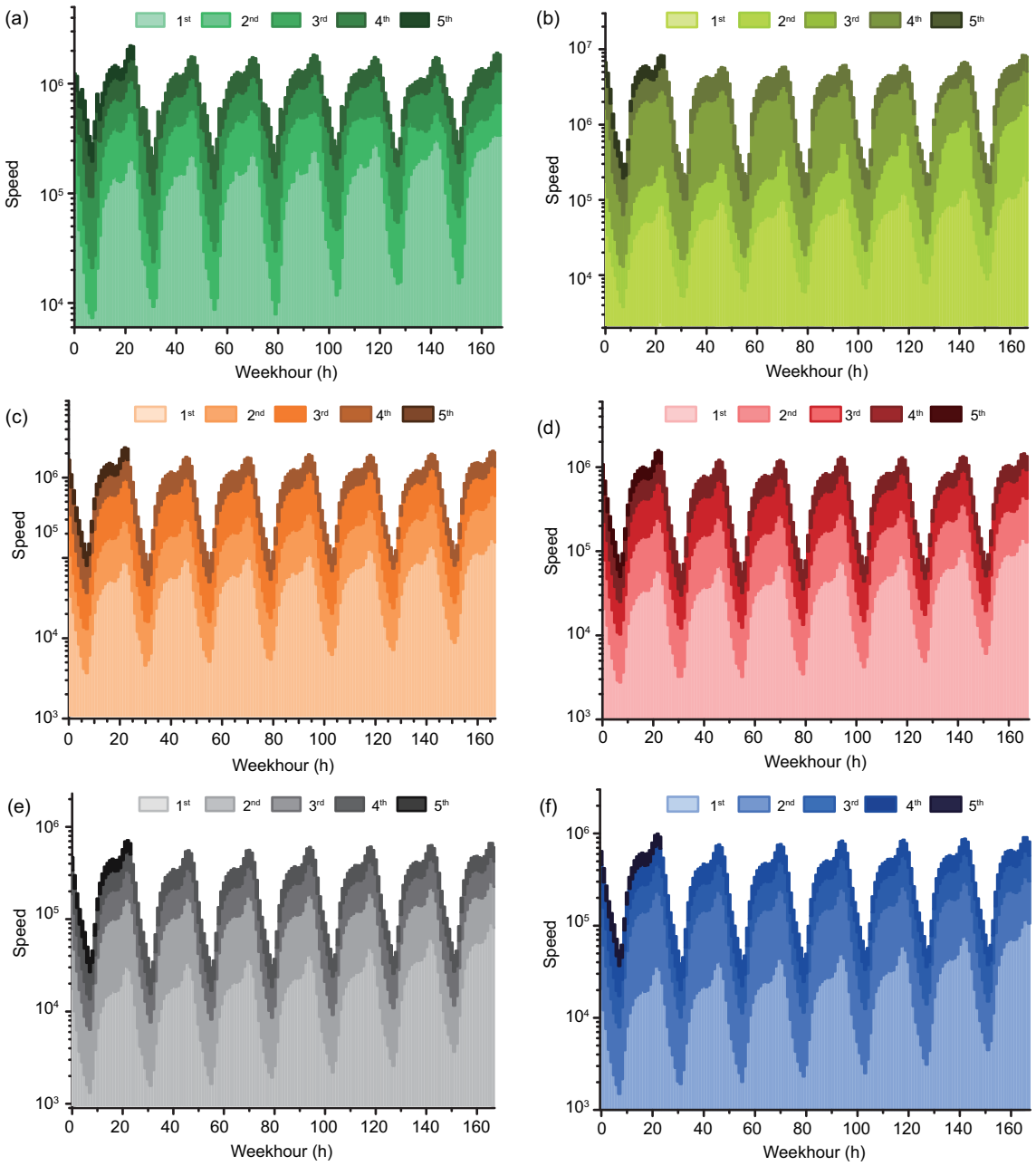


**Fig. 4 File spread speeds: (a) small files; (b) small-medium files; (c) medium files; (d) medium-large files; (e) large files; (f) huge files**

happens because more and more people participate in file dissemination with time flying, which increases the spread speeds.

According to the results in Fig. 4, we calculate an average weekly spread speed for each class of files, and normalize all the average speeds according to

$$\mathrm{Speed}_{\mathrm{Nor}}(i)=\frac{\mathrm{Speed}_{\mathrm{Ave}}(i)}{\max\limits_{j\in[1,5]}\left(\mathrm{Speed}_{\mathrm{Ave}}(j)\right)}, \tag{1}$$

where $\mathrm{Speed}_{\mathrm{Nor}}(i)$ describes the normalized average speed of a certain class of files in the $i^{\mathrm{th}}$ week. $\mathrm{Speed}_{\mathrm{Ave}}(i)$ and $\mathrm{Speed}_{\mathrm{Ave}}(j)$ represent the average speed of the same class of files in the $i^{\mathrm{th}}$ and $j^{\mathrm{th}}$ weeks, respectively. Fig. 5 shows the normalized average weekly speeds. According to Fig. 5, the gap between the normalized average speed in the first week and that in the second week is wide for large and huge files, while it is relatively narrow for other

files. It is also observed that the normalized average speeds of large files and huge files spread in the second week are closer to those in the third week, while other files (S, SM, M, and ML) do not. These discoveries give useful clues for forecasting popular files.

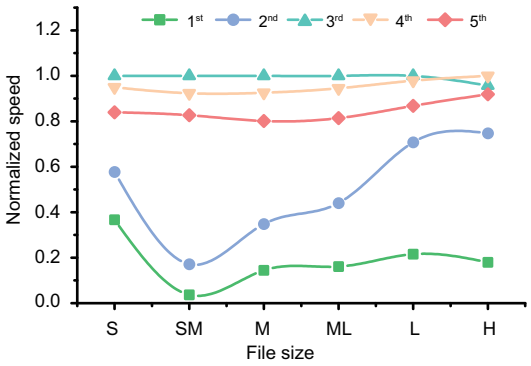


**Fig. 5 Normalized average speeds**

### 4.3 Type analysis

Xender provides delivering services for multiple types of files. It supports the transmition of Apps, audio, files, folds, images, music, and videos. Figuring out which types of files can be transmitted most widely is important, which can provide valuable information for popular file prediction and system optimization like resource allocation policies and cache policies. For each type of file, we calculate the proportions of files in different spread scales and depict the results in Fig. 6. According to Fig. 6, most files are tiny files regardless of file type. This means most files are non-popular files. Only very few App files are transmitted in huge scales, and hence they are popular files. The distribution of popular and non-popular files is consistent with the long tail effect.

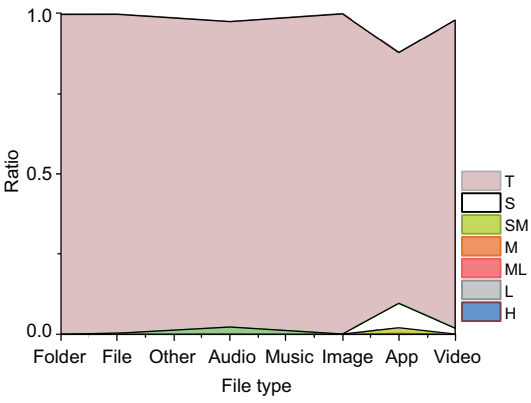


**Fig. 6 Ratio of different file scales**

We analyze the proportions of different types of files in each class and describe the results in Fig. 7. According to Fig. 7, image, video, and audio files are shared most. However, the proportions of these files decrease as spread scales expand. For example, image, video, and audio files account for 1%, 36%, and 43% in small-medium files, 0%, 24%, and 32% in medium files, and 0%, 8%, and 0% in medium-large files, respectively. However, App files are quite different. The proportion of App files increases with the growth of the spread scale. For example, App files account for 8% in small files, while reaching 20% in small-medium files and 43% in medium files. Moreover, all huge files are App files.

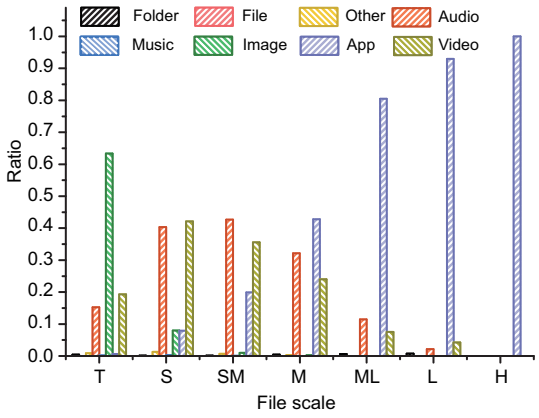


**Fig. 7 Ratio of different file types**

Based on the above analysis, app files are most possible to be transmitted most widely and hence have the chance to become popular. It is very important to analyze the spread trends of popular files since it can provide valuable information for offloading traffic and forecasting information diffusion. Some further steps can be taken, such as caching files to be popular, to accelerate file spread or release traffic loads of mobile networks.

## 5 Spread scale maximization

The spread scales of files determine whether they can become popular. In this section, we elaborate our solution to maximize the spread scales of files. The solution selects some nodes as seeds to initiate spread and then deduce a spread scale depending on the LT model.

### 5.1 Seed selection

It is critical to select the proper nodes as seeds to initiate propagation. High-quality seeds can not only expand spread scales, but also speed up spread processes. Files will be propagated to different scales if different seeds are selected. To facilitate seed selection, we introduce the following definitions:

**Definition 1** (Degree factor) $\forall\ v_i \in \boldsymbol{V}$, the degree factor of $v_i$ describes the impact from degree attributes of $v_i$ on seed selection. It is denoted as $\text{factor}_{\text{degree}}(i)$ and defined by Eq. (2):

$$\text{factor}_{\text{degree}}(i) = \frac{\text{degree}_{\text{out}}(i)}{\text{degree}_{\text{in}}(i) + \text{degree}_{\text{out}}(i)}. \quad (2)$$

**Definition 2** (Seed factor) $\forall\ v_i \in \boldsymbol{V}$, the seed factor of $v_i$ describes the suitability of $v_i$ as a seed. It is denoted as $\text{factor}_{\text{seed}}(i)$ and defined by Eq. (3):

$$\text{factor}_{\text{seed}}(i)=\text{factor}_{\text{degree}}(i)\cdot\text{rank}\Big(\text{factor}_{\text{degree}}(i)\Big), \quad (3)$$

where $\text{rank}\Big(\text{factor}_{\text{degree}}(i)\Big)$ represents the weight of $\text{factor}_{\text{degree}}(i)$ in all degree factors. Here, the PageRank algorithm (Brin and Page, 1998) is adopted to generate a weight for each node because of its high performance on very large size datasets. Any other rank algorithm can be adopted if it has the ability to process such a large size dataset with high performance. Based on the above definitions, seed nodes are selected through the following steps:

1. Calculate a degree factor for each node according to Eq. (2).

2. Apply the PageRank algorithm to all degree factors to generate a rank value for each node.

3. Calculate a seed factor for each node according to Eq. (3).

4. Sort all nodes in descending order of seed factors and take the top $k$ nodes as seeds.

### 5.2 Influence calculation

In the LT model, an inactive node changes to active status if the influence from its neighbors reaches or exceeds a predefined threshold. $\forall\ v_i \in \boldsymbol{V}$, the influence from the neighbors of $v_i$ is described as $\text{Inf}(i)$. For convenience of influence calculation, we define activeness and closeness.

**Definition 3** (Activeness) $\forall\ v_i \in \boldsymbol{V}$, the activeness of $v_i$ describes the enthusiasm of $v_i$ to participate in file-sharing activities. It is denoted as $\text{activeness}(i)$. $\text{activeness}(i)$ is obtained by calculating the file-sharing frequency of $v_i$.

**Definition 4** (Closeness) $\forall\ v_i \in \boldsymbol{V}$ and $\forall\ v_j \in \boldsymbol{V}$, the closeness between $v_i$ and $v_j$ describes the contact frequency between them. It is denoted as $\text{closeness}(i,j)$.

$$\text{Inf}(i) = \sum_{j\in\boldsymbol{N}_i}\Big(\text{activeness}(j)\cdot\text{closeness}(i,j)\Big). \quad (4)$$

Based on Definitions 3 and 4, $\text{Inf}(i)$ is calculated according to Eq. (4). If the influence on a node does not reach a predefined threshold, that node will not become active, which means it is impossible for the node to take part in file spread. To improve the efficiency of our solution, nodes are excluded if they remain inactive during file spread. However, this does not mean all the remaining nodes will become active. Some can remain inactive.

### 5.3 Spread scale prediction

According to the LT model, all active nodes are supposed to participate in file spread. Therefore, the file spread scale can be represented by the number of active nodes. Algorithm 1 shows the process to

**Algorithm 1** Spread scale prediction

**Input:** seeds $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$
**Output:** activeNodes

```
newNodes ← ∅
curNodes ← seeds
Add seeds into activeNodes
Calculate the influence for each node
Remove all the nodes of which the influence is smaller
    than θ from V
while (|newNodes|/|V| > ξ) or (newNodes = ∅) do
  newNodes ← ∅
  for each v_i ∈ curNodes do
    for each v_j belonging to the neighbor set of v_i
    do
      Inf(j) ← Calculate influence according to the
          active neighbors of v_j
      if Inf(j) ≥ θ then
        Change the status of v_j into active
        Insert v_j into newNodes
      end if
    end for
  end for
  curNodes ← newNodes
  Add newNodes to activeNodes
end while
Return activeNodes
```

predict active nodes. The prediction starts from seed nodes and promotes to inactive nodes iteratively. In the algorithm, $\theta$ represents the predefined threshold determining whether a node can change to active status and $\xi$ describes the predefined constant determining whether to stop iterations. In each iteration, each node in curNodes is traversed to decide whether their neighbors should be changed to active. All the neighbors that become active in the current iteration are added to newNodes. If the ratio of the number of nodes in newNodes to the total number of nodes is not larger than $\xi$, the prediction converges.

In our solution, influence is calculated multiple times. Some calculations are performed before iterations, while the others are implemented in iterations. The former calculations are executed with the influence from all neighbors, while the latter ones are performed with the influence only from active neighbors. The former calculations are exploited to remove the nodes that cannot participate in file spread, and the latter ones are employed to decide whether a node could become active. Finally, the file spread scale is determined by the total number of nodes predicted to be active.

### 5.4 Experimental results

In this subsection, we elaborate the evaluation of our approach. We extract a graph from the Xender dataset. The graph includes about 50 million vertices and about 100 million edges, where each vertex represents a user and each edge expresses the file-sharing activities between two users. Our solution is evaluated by comparison with two other solutions which select seeds in different ways but predict active nodes in the same way. These two solutions are denoted as MaxDegree+LT and PageRank+LT. MaxDegree+LT selects seeds according to max degree policy (Kempe et al., 2003; Rahimkhani et al., 2015), while PageRank+LT selects seeds according to PageRank values. Active nodes and execution time are used as the metrics to measure the solutions. The largest number of active nodes and the shortest execution time indicate the highest performance.

Before the experiment, we assign 0.0002 and 0.25 to $\xi$ and $\theta$, respectively, in Algorithm 1 as their default values. We run all the three solutions in Spark described in Table 1. The average numbers of active nodes are calculated for each solution when different numbers of seeds are selected. Fig. 8a shows the calculation results. According to Fig. 8a, our approach activates the largest number of nodes except for the scenarios in which prediction is started with 10 000 and 40 000 seeds, respectively. In the first scenario, the average number of nodes activated by our approach is 0.045% smaller than that by MaxDegree+LT. In the second scenario, the average number of nodes activated by our approach is 0.001% smaller than that by MaxDegree+LT. However, in all the other scenarios, our solution outperforms MaxDegree+LT.

To improve efficiency, our approach excludes all nodes that cannot become active. To evaluate the efficiency of our approach, we compare the execution times of the three solutions and plot the results in Fig. 8b. According to Fig. 8b, no matter how many seeds are chosen, our approach takes the least time to predict active nodes. On average, the time spent by our approach is about 86.7% of the time spent by PageRank+LT and 91.1% of that spent by MaxDegree+LT. In other words, our approach reduces the time spent on active node prediction by 13.3% and 8.9% when compared with PageRank+LT and MaxDegree+LT, respectively. Based

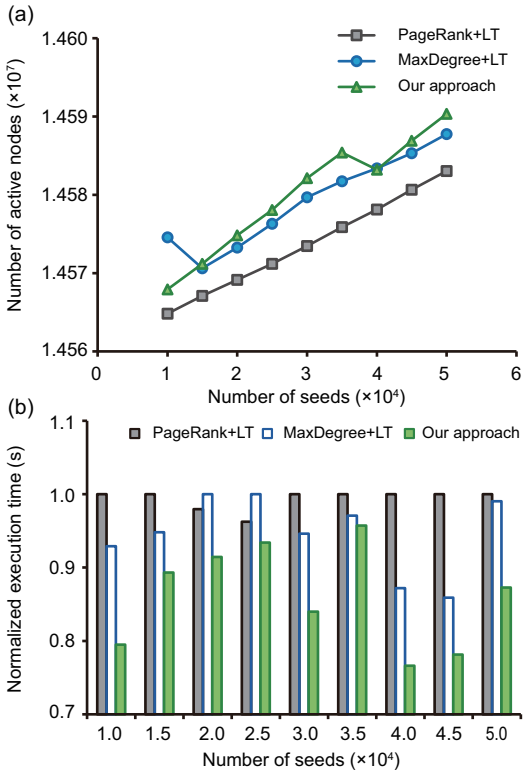


**Fig. 8 Comparison on the Xender dataset: (a) scale with different numbers of seeds; (b) execution time**

on the results in Fig. 8, our approach exhibits the highest performance.

The cost-effective lazy forward selection (CELF) algorithm (Leskovec et al., 2007b) maximizes spread scales by exploiting greedy policies. It is well known because it has the ability to obtain results approximating the optimum with high efficiency. To be more convincing, we try to compare our approach with an enhancement of CELF, that is, CELF++ (Goyal et al., 2011). However, CELF++ gets no invalid results on the Xender dataset since it is incapable of processing such a large-size dataset. Therefore, we implement the comparison on two datasets which are smaller than the Xender dataset. These two datasets are the European (EU) institution dataset (Leskovec et al., 2007a), which contains about 265 000 nodes and about 420 000 edges, and the Soc-sign-epinion dataset (Leskovec et al., 2010), which includes about 131 000 vertices and about 841 000 edges.

Fig. 9 shows the execution times on both datasets. According to Fig. 9, our approach always executes faster than CELF++. It obtains about a 46% saving on the EU institution dataset and about a 60% saving on the Soc-sign-epinion dataset compared with CELF++. Fig. 10 shows the comparison of activated nodes. It is obvious that our approach matches better CELF++ on the EU institute dataset than on the Soc-sign-epinion dataset. Although our approach activates fewer nodes than CELF++, it promotes spread with higher performance, which is very important for spread scale maximization in large-scale social networks.

## 6 Conclusions

Understanding file spread characteristics in opportunity networks is critical for offloading the huge traffic on mobile networks. In this study, we aim to discover the spread characteristics by performing measurement and analysis on the dataset collected from Xender, which provides file-sharing services through user mobility and D2D communication. We take file spread procedures as graphs and implement scale, speed, and type analyses on the dataset. We also propose a solution to maximize file spread scales. This is very important for forecasting popular files

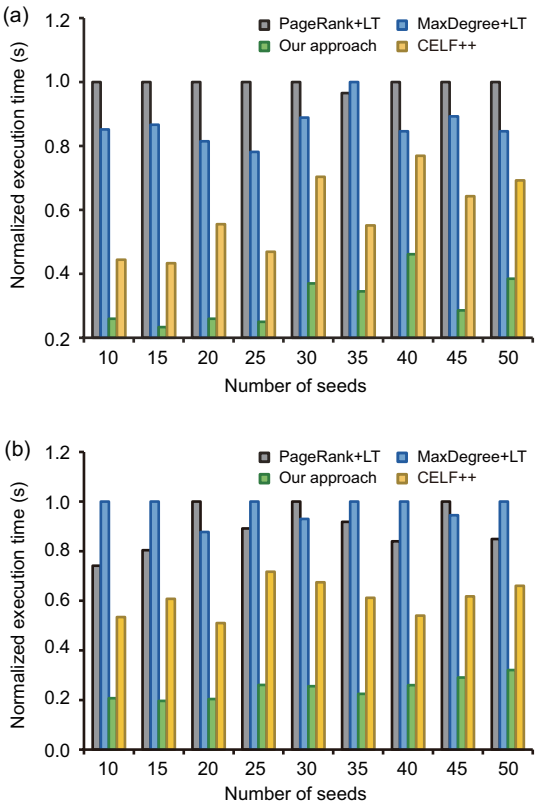


**Fig. 9 Comparison of execution time: (a) EU institute dataset; (b) Soc-sign-epinion dataset**

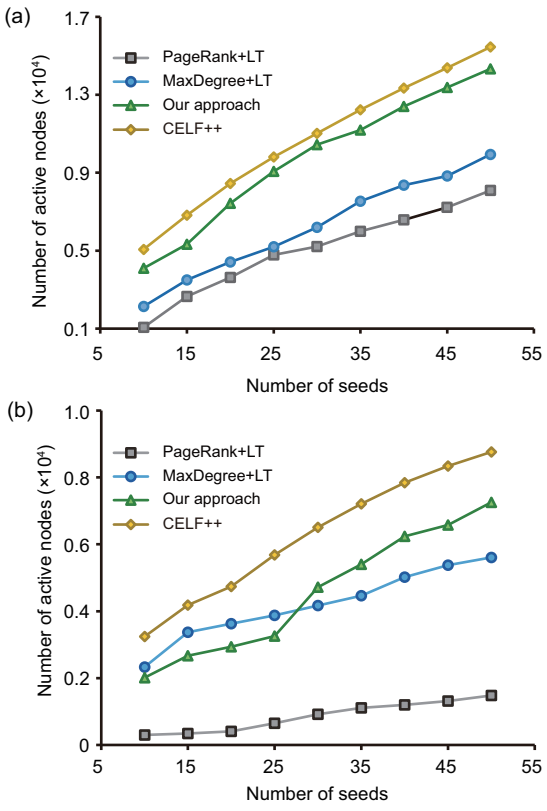


**Fig. 10 Comparison of spread influence: (a) EU institute dataset; (b) Soc-sign-epinion dataset**

and helping make traffic offloading strategies. In the future, we will focus on the insightful research on the topics about optimizing our approach, forecasting popular files, and analyzing user impacts.

## Compliance with ethics guidelines

Xiao-hong ZHANG, Kai QIAN, Jian-ji REN, Zong-pu JIA, Tian-peng JIANG, and Quan ZHANG declare that they have no conflict of interest.

## References


Andreev S, Pyattaev A, Johnsson K, et al., 2014. Cellular traffic offloading onto network-assisted device-to-device connections. *IEEE Commun Mag*, 52(4):20-31. https://doi.org/10.1109/MCOM.2014.6807943

Bao XY, Zhou XJ, Zhang Y, et al., 2016. Cellular traffic offloading utilizing set-cover based caching in mobile social networks. *J China Univ Posts Telecommun*, 23(2): 46-55. https://doi.org/10.1016/S1005-8885(16)60020-1

Brin S, Page L, 1998. The anatomy of a large-scale hypertextual Web search engine. *Comput Netw ISDN Syst*, 30(1-7):107-117. https://doi.org/10.1016/S0169-7552(98)00110-X

Cha M, Kwak H, Rodriguez P, et al., 2007. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. Proc 7th ACM SIGCOMM Conf on Internet Measurement, p.1-14. https://doi.org/10.1145/1298306.1298309

Che HL, Cao Y, 2014. Examining WeChat users' motivations, trust, attitudes, and positive word-of-mouth: evidence from China. *Comput Human Behav*, 41:104-111. https://doi.org/10.1016/j.chb.2014.08.013

Cheng RG, Chen NS, Chou YF, et al., 2015. Offloading multiple mobile data contents through opportunistic device-to-device communications. *Wirel Pers Commun*, 84(3):1963-1979. https://doi.org/10.1007/s11277-015-2492-1

Chuang YJ, Lin KCJ, 2012. Cellular traffic offloading through community-based opportunistic dissemination. IEEE Wireless Communications and Networking Conf, p.3188-3193. https://doi.org/10.1109/WCNC.2012.6214356

Cisco, 2017. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html [Accessed on Feb. 27, 2019].

Gao G, Xiao M, Wu J, et al., 2016. Deadline-sensitive mobile data offloading via opportunistic communications. 13th Annual IEEE Int Conf on Sensing, Communication, and Networking (SECON), p.1-9. https://doi.org/10.1109/SAHCN.2016.7732980

Goyal A, Lu W, Lakshmanan L, 2011. CELF++: optimizing the greedy algorithm for influence maximization in social networks. 20th Int Conf Companion on World Wide Web, p.47-48. https://doi.org/10.1145/1963192.1963217

Guan W, Gao H, Yang M, et al., 2013. Analyzing user behavior of the micro-blogging website Sina Weibo during hot social events. *Phys A Stat Mech Appl*, 395:340-351. https://doi.org/10.1016/j.physa.2013.09.059

Ioannidis S, Chaintreau A, Massoulie L, 2009. Optimal and scalable distribution of content updates over a mobile social network. IEEE INFOCOM, p.1422-1430. https://doi.org/10.1109/INFCOM.2009.5062058

Jiang J, Zhang S, Li B, et al., 2016. Maximized cellular traffic offloading via device-to-device content sharing. *IEEE J Select Areas Commun*, 34(1):82-91. https://doi.org/10.1109/JSAC.2015.2452493

Jiang N, Guo L, Li J, et al., 2016. Data dissemination protocols based on opportunistic sharing for data offloading in mobile social networks. 22nd Int Conf on Parallel and Distributed Systems, p.705-712. https://doi.org/10.1109/ICPADS.2016.0097

Kempe D, Kleinberg J, Tardos E, 2003. Maximizing the spread of influence through a social network. Proc 9th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.137-146. https://doi.org/10.1145/956750.956769

Leskovec J, Kleinberg J, Faloutsos C, 2007a. Graph evolution: densification and shrinking diameters. *ACM Trans Knowl Discov Data*, 1(1):1-40. https://doi.org/10.1145/1217299.1217301

Leskovec J, Krause A, Guestrin C, et al., 2007b. Cost-effective outbreak detection in networks. Proc 13th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.420-429. https://doi.org/10.1145/1281192.1281239

Leskovec J, Huttenlocher D, Kleinberg J, 2010. Signed networks in social media. 10th SIGCHI Conf on Human Factors in Computing Systems, p.1361-1370. https://doi.org/10.1145/1753326.1753532

Lin KCJ, Chen CW, Chou CF, 2012. Preference-aware content dissemination in opportunistic mobile social networks. IEEE INFOCOM, p.1960-1968. https://doi.org/10.1109/INFCOM.2012.6195573

Lu Z, Wen Y, Cao G, 2014. Information diffusion in mobile social networks: the speed perspective. IEEE Conf on Computer Communications, p.1932-1940. https://doi.org/10.1109/INFOCOM.2014.6848133

Mashhadi AJ, Mokhtar SB, Capra L, 2012. Fair content dissemination in participatory DTNs. *Ad Hoc Netw*, 10(8):1633-1645. https://doi.org/10.1016/j.adhoc.2011.05.010

Obar JA, Wildman S, 2015. Social media definition and the governance challenge: an introduction to the special issue. *Telecommun Pol*, 39(9):745-750. https://doi.org/10.1016/j.telpol.2015.07.014

Pietiläinen AK, Diot C, 2012. Dissemination in opportunistic social networks: the role of temporal communities. 13th ACM Int Symp on Mobile Ad Hoc Networking and Computing, p.165-174. https://doi.org/10.1145/2248371.2248396

Rahimkhani K, Aleahmad A, Rahgozar M, et al., 2015. A fast algorithm for finding most influential people based on the linear threshold model. *Expert Syst Appl*, 42(3): 1353-1361. https://doi.org/10.1016/j.eswa.2014.09.037

Rebecchi F, de Amorim MD, Conan V, 2016. Should I seed or should I not: on the remuneration of seeders in D2D offloading. 17th Int Symp on a World of Wireless, Mobile and Multimedia Networks, p.1-9. https://doi.org/10.1109/WoWMoM.2016.7523496

Tang R, 2017. Performance tradeoff between energy conservation and user fairness for D2D communication underlaying cellular networks. *Chin J Electron*, 26:600-607. https://doi.org/10.1049/cje.2016.11.011

Thilakarathna K, Viana AC, Seneviratne A, et al., 2012. The Power of Hood Friendship for Opportunistic Content Dissemination in Mobile Social Networks. Research Report No. 8042, Teams HIPERCOM, Université Paris-Saclay, France.

Thilakarathna K, Seneviratne A, Viana AC, et al., 2014. User generated content dissemination in mobile social networks through infrastructure supported content replication. *Perv Mob Comput*, 11(2):132-147. https://doi.org/10.1016/j.pmcj.2014.01.005

Tian F, Liu B, Xiong J, et al., 2016. Movement-based incentive for cellular traffic offloading through D2D communications. IEEE Int Symp on Broadband Multimedia Systems and Broadcasting, p.1-5. https://doi.org/10.1109/BMSB.2016.7521954

Wang G, Liu PZ, Yang Z, et al., 2018. Joint college admissions game and auction theory for data offloading in heterogeneous networks. *Chin J Electron*, 27(1):168-174. https://doi.org/10.1049/cje.2017.09.001

Wang H, Wang S, Zhang Y, et al., 2017a. Measurement and analytics on social groups of device-to-device sharing in mobile social networks. Int Conf on Communications, p.1-6. https://doi.org/10.1109/ICC.2017.7997038

Wang H, Wang X, Li K, et al., 2017b. A measurement study of device-to-device sharing in mobile social networks based on Spark. *Concurr Comput Pract Exp*, 29(16):e4021. https://doi.org/10.1002/cpe.4021

Wang X, Chen M, Han Z, et al., 2014. TOSS: traffic offloading by social network service-based opportunistic sharing in mobile social networks. IEEE Conf on Computer Communications, p.2346-2354. https://doi.org/10.1109/INFOCOM.2014.6848179

Wang Z, Sun L, Zhang M, et al., 2016. Social- and mobility-aware device-to-device content delivery. http://arxiv.org/abs/1606.04195

Zhang S, Wu J, Qian Z, et al., 2015. Mobicache: cellular traffic offloading leveraging cooperative caching in mobile social networks. *Comput Netw*, 83:184-198. https://doi.org/10.1016/j.comnet.2015.03.011

Zhang Y, Pan E, Song L, et al., 2015. Social network aware device-to-device communication in wireless networks. *IEEE Trans Wirel Commun*, 14(1):177-190. https://doi.org/10.1109/TWC.2014.2334661

Zhao Y, Song W, 2016. Social-aware energy-efficient data dissemination with D2D communications. IEEE 83rd Vehicular Technology Conf, p.1-5. https://doi.org/10.1109/VTCSpring.2016.7504479