Data Size Aware Forwarding in Opportunistic Mobile Networks

Xiao Chen
Department of Computer Science, Texas State University, San Marcos, TX 78666
Email: xc10@txstate.edu

***Abstract*—With the proliferation of mobile devices, sending data by taking advantage of the opportunistic contact in an opportunistic mobile network (OMN) becomes a hot topic. Many previous data forwarding schemes assume that content can always be entirely transmitted over a single contact. However, with the increase of data sizes, such an assumption may no longer hold. In this paper, we propose a data size aware forwarding (DSAF) scheme for OMNs. Our method dynamically discovers the maximum transmission unit of an arbitrary path (PMTU) so that the source can adjust the amount of data to send in each transmission. Since the performance of DSAF is largely dependent on the latency to get the Message Too Big (MTB) feedback message, we first conduct theoretical analysis of the latencies for the receiver(s) to receive MTBs and then compare our theoretical results with the practical ones using a real trace. Simulation results show that our theoretical values closely match the practical ones and our DSAF scheme implemented by additive and multiplicative decrease flavors have low overhead compared with the benchmark algorithm where PMTU is pre-known.**

***Index Terms*—feedback, forwarding, maximum transmission unit, opportunistic mobile networks**

I. Introduction

With the proliferation of mobile devices and content delivery over these devices, opportunistic mobile network (OMN) has become a promising traffic offloading approach to relieving the burden of the overloaded cellular networks [2]. Due to high device mobility in OMNs, the path between the source and the destination is intermittent and hence the content forwarding uses multi-hop transmission [3].

Many previous data forwarding schemes [5], [12], [15] assume that content can always be entirely transmitted over a single opportunistic contact. However, with the rapid increase of content size in recent years, such an assumption may no longer hold. Currently, very few papers have addressed this issue. The only thrust we have found in this direction is in papers [13], [14]. The authors in these two papers propose fragmenting the original large data and sending the data chunks through multiple paths to the destination. One problem of this scheme, as they point out, is that the destination needs to collect all the data chunks from multiple paths to assemble the data. The loss or the delay of a fragment in any path can affect the collection and increase the delivery delay at the destination.

To avoid the fragmentation issue, in this paper, we take a different approach. Our method is let the source dynamically discover the maximum transmission unit of a path (PMTU) between itself and the destination so that it knows the amount of data to send to fit the opportunistic contact capacity of the network. The idea is enlightened by the Path MTU discovery (PMTU) protocol in the wired Internet [9]. The general procedure is shown in Fig. 1: source s tries to send the data and hopefully the data can pass the intermediate forwarders $f_1, f_2, \cdots, f_n$ to reach the destination d. But the entire packet may not go through if it is too large. So if some intermediate forwarder detects this, it will send a Message Too Big (MTB) message as the feedback to the source. Then the source will reduce its packet size and resend. The process goes on for several rounds until the source discovers the right size (PMTU) of the packet it can send to the destination through the network.

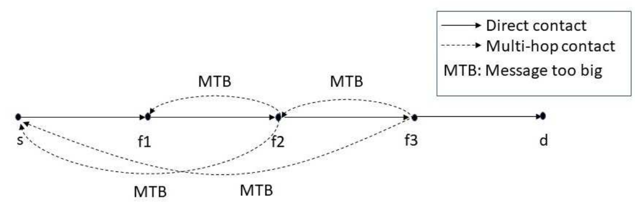


Fig. 1. Path Maximum Transmission Unit (PMTU) Discovery in OMNs

Because of the network differences, the PMTU protocol in the wired network cannot be directly used in the OMN. First, in the wired network, a host can test the bandwidth of the link and report to the source what size can fit. But in the OMN, a node cannot test the bandwidth of the link. Second, the PMTU discovery in the OMN is an on-going process. A PMTU discovered by the previous transfer may not be the one for the next transfer due to the change of the end-to-end path or nodes' contact duration. Third, there are relatively few PMTU values in use in the wired network because designers tend to choose PMTUs in similar ways. Thus we can only search the ones that are likely to appear. But the PMTU value in the OMN depends on the contact duration of two mobile nodes which is related to the mobility pattern of the OMN. Thus, the PMTU discovery process in the OMN needs to be redesigned.

In this paper, we put forward a data size aware forwarding protocol (DSAF) to address the large data forwarding issue in OMNs. DSAF has two flavors: *additive decrease* and *multiplicative decrease*. The essence of DSAF is the PMTU discovery. We use epidemic routing [12] to quickly disseminate the MTB message to speed up the PMTU discovery and perform the search of PMTU in parallel with data sending to reduce the network overhead. We show analytically and experimentally using a real trace that DSAF can discover the right size fast with low overhead.

The key contributions of our work are as follows:

- We are among the few who address the large data forwarding issue in OMNs.
- We propose a data size aware forwarding protocol based on the idea of PMTU discovery.
- We demonstrate the efficiency of the protocol analytically and experimentally using a real trace.

The rest of the paper is organized as follows: Section II references the related work; Section III defines the problem; Section IV presents our solution; Section V gives mathematical analysis of our solution; Section VI shows the simulations; and Section VII is the conclusion.

II. Related Work

Most previous data forwarding schemes [5], [12], [15] assume that content can always be entirely transmitted over a single opportunistic contact. The authors in [13], [14] did not make such an assumption and proposed to fragment the original large data at the source and send data chunks through multiple paths to the destination. With data partition, the delay in any path would lead to a larger delivery latency. So they designed schemes to determine *which* data and *how* much data should be forwarded to *whom* through data partitioning to fully utilize the relatively short and abundant contact opportunities. In [13], they found the optimal data partitioning size in blind flooding and used network coding techniques to improve performance. And in [14], they further proposed to use nodes' social features to generate multiple edge-disjoint multi-hop forwarding paths to deliver the data chunks to the destination.

To avoid the data fragmentation issue, we take a different approach based on the PMTU discovery. In our solution, we use epidemic routing [12] to distribute the MTB feedback message. Epidemic routing spreads a message epidemically without data replication control. It is appropriate in our method because the source wants to know the PMTU value as soon as possible and other nodes are also interested in this value for future transmission. So our work is distinct from the related work.

III. Problem Definition

We consider an OMN with N mobile nodes contacting each other opportunistically through short range interfaces (e.g., Bluetooth or Wi-Fi). If a mobile node has new content, it can distribute it to other interested node either directly when they contact (move into each other's transmission range) or through multiple hops via intermediate forwarders. In each contact of two nodes, the message holder decides how much data to send. If the data transmission is finished before they depart, the data forwarding is successful. Otherwise, it fails. We assume that the contact between two nodes may not be long enough to finish forwarding the large data items. The data items will not be fragmented. Our goal is to design an efficient data size aware protocol to address the large data forwarding issue in OMNs.

IV. Our Solution

In this section, we propose the DSAF protocol to address the large data forwarding issue in OMNs.

A. Protocol Overview

In DSAF, the key point is for the source to find the MTU of the path to reach the destination so that it can adjust its data sending size. The procedure can be explained in detail using Fig. 1. Here source s tries to send the data to the destination d. This process may involve multiple intermediate forwarders $f_1, f_2, \cdots$. A node can send data to another one directly when they contact. In the figure, a solid line represents a direct contact. Initially source s sends out a large packet and we assume that the packet reaches forwarder f_1 completely. But when f_1 sends the packet to f_2, due to their short contact duration, the packet does not entirely reach f_2. So f_2 will drop the partially received packet and send an MTB message to s using epidemic routing. The MTB message may reach s in multiple hops through intermediate forwarders. In the figure, we use a dashed line to represent a multi-hop communication. Once s gets MTB, it will reduce the packet size and resend. Source s can reduce the packet size in two ways: additive decrease and multiplicative decrease. In additive decrease, s can reduce the size by a certain amount while in multiplicative decrease, s can cut the size by a percentage. This process continues until s receives very few or no more MTB messages. Then it is able to discover the MTU value of the path to the destination. The source will cache PMTU and update it after a new value is learned. All the other nodes that receive MTB via epidemic routing also get the idea of the right size for future data transfers.

B. Epidemic Routing of MTB Message

In the PMTU discovery process, the MTB message is sent back using epidemic distribution. In the epidemic model, nodes continuously replicate and transmit messages to newly discovered contacts that do not already possess a copy of the message. It is usually treated as an expensive process [8], [14] in OMNs. But it is appropriate here for the following reasons. First, the nature of the wireless communication is broadcast. Second, the size of the MTB message is small, which will not use too many network resources and can be successfully transmitted in one opportunistic contact. Third, all the nodes want to know this value for their future transfers.

C. Update of PMTU

In OMNs, the PMTU discovered for a given destination may be wrong if a new route comes into use due to node mobility. Thus, the PMTU cached by a host can become stale. If the stale PMTU is too large, it will be discovered immediately once an MTB message reaches the source. However, no such mechanism exists for realizing that the stale PMTU is too small. Thus we need to "age" the cached value: when PMTU has not been decreased for a while, the source node can increase its PMTU, for example, additively, for the next data transfer.

D. Overhead of the Protocol

The overhead of the protocol lies in the first few MTB messages used to find the MTU along the network path. But the epidemic distribution of MTB allows a source and other

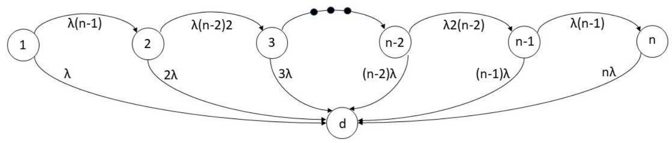


Fig. 2. Markov chain where each state X_i represents i nodes having received MTB and the destination d receives MTB after i nodes have received it

involved nodes to discover PMTU very quickly. After a while, they will not receive many MTBs. As long as the path is stable, the PMTU value is stable. But if the path changes, PMTU will change. So the search is an ongoing process. To reduce the overhead, we can perform the search in parallel with data sending. The performance of the protocol largely relies on the latency to receive an MTB. In the following, we conduct a theoretical analysis and later use simulations to show the low overhead of the protocol.

V. Analysis of Latency

In this section, we evaluate the efficiency of the DSAF protocol by theoretically estimating the latencies of the MTB message sent from a sender (source of the MTB message) to a receiver (destination of the MTB message) and from the sender to all the nodes in the network. In the Lemma and Theorems below, we assume that an OMN has n mobile nodes and the inter-contact time between two nodes obeys an exponential distribution with parameter λ following the results in [7].

A. Latency of MTB Received by a Receiver

Lemma 1. *We denote $E(Y_i)$ as the expected latency for a receiver to receive MTB spread epidemically from a sender after i nodes have received it. Then $E(Y_i) = \frac{i}{\lambda'}$, where $\lambda' = \lambda i(n+1-i)$.*

Proof.

The whole MTB epidemic distribution process can be described by a Markov chain in Fig. 2. The Markov chain is in state i if i nodes have received the MTB message. The transition rate from state i when there are i copies of MTB in the network to state $i+1$ when there are $i+1$ copies is $\lambda i(n-i)$ and one of these i copies reaches the destination d (transition from state i to d) at a rate λi. The sojourn time denoted by T_i in state i is exponentially distributed with parameter $\lambda' = \lambda i(n+1-i)$ (obtained as the sum of the transition rates going out of state i). Then the latency for a destination to receive MTB after i nodes have received MTB is a random variable $Y_i = \sum_{j=1}^{i} T_j$. To calculate the expected value of this variable, we first find its PDF. Normally the PDF of Y_i is the convolution of all the PDFs of $T_j s (1 \leq j \leq i)$. But we can use the following short-cut to obtain its PDF.

In order to see the MTB delivery after i nodes have got their MTBs during some time period, without loss of generality, we assume that $i-1$ nodes get their MTBs during interval $[0,t]$ and the i-th node gets its MTB in the small interval $[t, t+\delta]$. Variable δ is a value chosen small enough to contain the occurrence of just one receipt of MTB. Since the PDF of a continuous random variable is the probability per unit length, for a small interval δ, we have

$$f_{Y_i}(t)\delta = P(t \leq Y_i \leq t+\delta), t \geq 0$$

Since the times for i nodes to receive MTB can be seen as a Poisson process, due to the memorylessness of the process, the events that ith node receiving MTB and the previous $i-1$ nodes receiving MTB are independent. So we have

$$\begin{aligned} f_{Y_i}(t)\delta &= P(t \leq Y_i \leq t+\delta) \\ &= P(i-1 \text{ nodes get MTB in [0,t]}) \cdot \lambda'\delta \end{aligned}$$

According to the PDF of the Poisson process,

$$P(i-1 \text{ nodes get MTB in [0,t]}) = \frac{(\lambda' t)^{i-1}}{(i-1)!} \cdot e^{-\lambda' t}$$

So,

$$f_{Y_i}(t)\delta = \frac{(\lambda' t)^{i-1}}{(i-1)!} \cdot e^{-\lambda' t} \cdot \lambda'\delta$$

Canceling δ on both sides, we get Y_i's PDF

$$f_{Y_i}(t) = \frac{(\lambda')^i t^{i-1} e^{-\lambda' t}}{(i-1)!}, t \geq 0 \quad (1)$$

Now Y_i's expected value is:

$$\begin{aligned} E(Y_i) &= \int_0^\infty t f_{Y_i}(t)dt = \int_0^\infty t \frac{(\lambda')^i t^{i-1} e^{-\lambda' t}}{(i-1)!} dt \\ &= \frac{(\lambda')^i}{(i-1)!} \frac{1}{(\lambda')^i} \int_0^\infty i! e^{-\lambda' t} t^{i-i} dt = \frac{i}{\lambda'} \quad (2) \end{aligned}$$

□

Theorem 1. *We denote $E(T_d)$ as the expected latency for a receiver to receive MTB spread epidemically from a sender. Then $E(T_d) = \frac{1}{\lambda n} \sum_{i=1}^{n} \frac{1}{i}$.*

Proof.

The latency for a destination to receive MTB can be the latency after one node has received MTB, or two nodes have received MTB, or i nodes have received MTB, which are mutually independent. According to the law of total expectation [16], we have

$$E(T_d) = \sum_{i=1}^{n} E(T_d | X_i = i) P(X_i = i) \quad (3)$$

$E(T_d | X_i = i)$ is the expected latency to receive MTB after i nodes have received MTB in Equation (2) in Lemma 1. And for $P(X_i = i)$, we know that the Markov chain jumps from state i to state $i+1$ with probability $\frac{n-i}{n+1-i}$ and jumps from state i to state d with probability $\frac{1}{n+1-i}$. So

$$P(X_i = i) = \frac{1}{n+1-i} \prod_{j=1}^{i-1} \frac{n-j}{n+1-j} = \frac{1}{n} \quad (4)$$

Putting Equations (2), (4) and $\lambda' = \lambda i(n+1-i)$ into Equation (3), we get

$$E(T_d) = \frac{1}{n} \sum_{i=1}^{n} E(T_d | X_i = i) = \frac{1}{n} \sum_{i=1}^{n} \frac{i}{\lambda'} = \frac{1}{\lambda n} \sum_{i=1}^{n} \frac{1}{i} \quad (5)$$

□

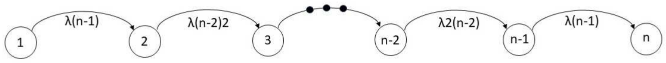


Fig. 3. Markov chain where each state X_i represents i nodes having received MTB

B. Latency of MTB Received by All

Theorem 2. *We denote $E(T)$ as the expected latency for all the nodes to receive MTB spread epidemically from a sender. Then $E(T) = \frac{2}{n\lambda}\sum_{i=1}^{n-1}\frac{1}{i}$.*

Proof. We model the epidemic distribution process by an n-state ($i = 1, \cdots, n$) Markov chain shown in Fig. 3. Same as the above, the Markov chain is in state i if i nodes have received MTB. As the inter-meeting time follows an exponential distribution with parameter λ, the transition rate from state i to state $i+1$ is $\lambda(n-i)i$. The sojourn time T_i in state i is exponentially distributed with parameter $\lambda(n-i)i$ (obtained as the transition rate going out of state i). Its PDF is:

$$f(t_i) = (n-i)i\lambda e^{-(n-i)i\lambda}, t_i \geq 0. \tag{6}$$

This Markov chain transition can be treated as a percolation process [10], where the whole delay T is a random variable which is equal to the sum of all the sojourn times in $n-1$ states.

$$T = \sum_{i=1}^{n-1} T_i = \sum_{i=1}^{n-1} \lambda(n-i)i \tag{7}$$

To estimate the latency for all the nodes to receive MTB, we calculate the expected value $E(T)$ of T. We denote random variable T's PDF as $f(t)$. Since T is the sum of a set of random variables following an exponential distribution, to facilitate the calculation, we can first derive its moment generating function (MGF) $M(s)$ and then obtain its expected value $E(T)$ by calculating $M'(0)$ [6].

The MGF of T is:

$$M(s) = E(e^{st}) = E(e^{\sum_{i=1}^{n-1} st_i}) \tag{8}$$

Since $T_1, T_2, \cdots, T_{n-1}$ are independent random variables, we have

$$M(s) = E(e^{\sum_{i=1}^{n-1} st_i}) = \prod_{i=1}^{n-1} E(e^{st_i}) \tag{9}$$

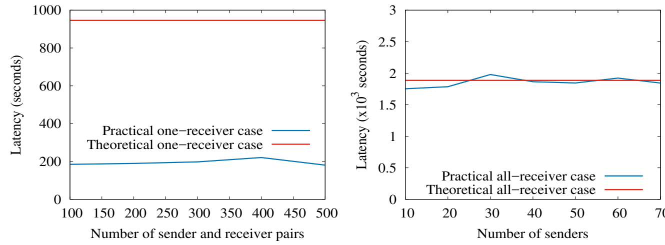
$E(e^{st_i})$ is the MGF of random variable T_i which follows an exponential distribution with parameter $(n-i)i\lambda$. Inserting Equation (6) into Equation (9), we get

$$\begin{aligned} E(e^{st_i}) &= \int_0^\infty e^{st_i} \cdot (n-i)i\lambda e^{-(n-i)i\lambda t_i} dt_i \\ &= (n-i)i\lambda \int_0^\infty e^{[s-(n-i)i\lambda]t_i} dt_i \\ &= \frac{(n-i)i\lambda}{(n-i)i\lambda - s} \end{aligned} \tag{10}$$

Plugging Equation (10) into Equation (9), we obtain

$$M(s) = \prod_{i=1}^{n-1} \frac{(n-i)i\lambda}{(n-i)i\lambda - s} \tag{11}$$

Now, we can calculate $E(T)$ as follows:



(a) Latency to get MTB for one receiver (b) Latency to get MTB for all receivers

Fig. 4. Theoretical and practical MTBs using Infocom trace

$$\begin{aligned} E(T) &= M'(0) = \frac{dM(s)}{ds}\Big|_{s=0} = \frac{d\prod_{i=1}^{n-1}\frac{(n-i)i\lambda}{(n-i)i\lambda - s}}{ds}\Big|_{s=0} \\ &= \sum_{i=1}^{n-1}\Big(\frac{d\frac{(n-i)i\lambda}{(n-i)i\lambda-s}}{ds}\prod_{j\neq i}\frac{(n-j)j\lambda}{(n-j)j\lambda - s}\Big)\Big|_{s=0} \\ &= \frac{1}{n\lambda}\sum_{i=1}^{n-1}\Big(\frac{1}{n-i}+\frac{1}{i}\Big) = \frac{2}{n\lambda}\sum_{i=1}^{n-1}\frac{1}{i} \end{aligned} \tag{12}$$

□

VI. Simulations

In this section, we evaluate the performance of our proposed DSAF scheme using a customized simulator in Matlab.

A. Real Trace Used

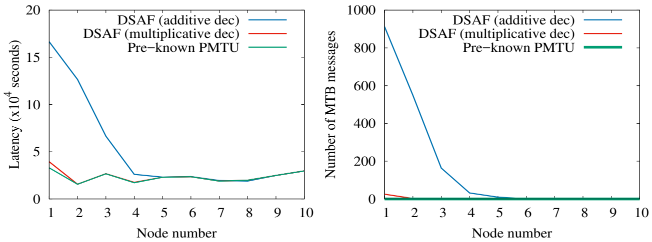
To obtain the opportunistic meetings of users, we used the Infocom trace [11] on the Crawdad website [1]. The Infocom trace has been widely used to test routing algorithms in mobile social networks [4], [17]. It recorded 78 attenders' encounter history using Bluetooth small devices (iMotes) at INFOCOM 2006 for 4 days.

B. Theoretical and Practical Latencies

In the first experiment, we compared the theoretical and practical latencies for a node and all nodes to receive MTB from a sender using the trace.

For the one-receiver case, we wanted to find out the latency starting from the time the sender generates an MTB until the time the receiver receives the message after epidemic distribution. We observed the whole Infocom trace with $n = 78$ nodes. To obtain the theoretical latency, we considered all the inter-contact times of all the pairs in the trace and found the average inter-contact time rate λ to be 6.6965×10^{-5}. Using Equation (5), we obtained the theoretical latency of the one-receiver case. It is a constant after λ and n are fixed. To get the practical latency, we randomly generated 100 to 500 sender and receiver pairs in the trace and the practical latencies were obtained through simulation. The theoretical and practical latencies are shown in Fig. 4(a). The difference between the theoretical and the practical values is reasonable which shows the effectiveness of our model.

For the all-receiver case, we counted the latency from the time the sender generates an MTB until the time all the nodes



(a) Latency

(b) # of MTB messages (zero for Pre-known PMTU)

Fig. 5. Latency and MTBs using Infocom trace

receive the message after epidemic distribution. We calculated the theoretical value using Equation (12). Again it is a constant after λ and n are fixed. To obtain the actual latency, we randomly generated 10 to 70 senders and the practical latencies were obtained through simulation. The results are shown in Figs. 4(b). We can see that the practical and the theoretical latencies match so well that their curves are almost overlapped.

In summary, our derived theoretical latency values closely match the practical ones in the trace.

C. Latencies using Different Algorithms

In the second experiment, we compared the two flavors (additive decrease and multiplicative decrease) of the DSAF scheme with the benchmark algorithm having a known PMTU value. We used two metrics: latency and the number of MTBs generated during the process of a large-size message sent entirely from a source to a destination.

1) *DSAF (additive dec)*: This is one flavor of DSAF. Each time a source receives MTB, it will decrease the message size additively by a certain amount.
2) *DSAF (multiplicative dec)*: This is another flavor of DSAF. Each time a source receives MTB, it will decrease the message size multiplicatively by a certain percentage.
3) *Pre-known PMTU*: This is the benchmark algorithm where PMTU is pre-known.

In the trace, we set the message size to $20MB$, the bandwidth to $100kbps$, the decrement amount to $100k$ bits, and the multiplicative percentage to 25%. We looked at the first 10 sources in the network. We calculated the latency for them to finish sending a message to the destination and the number of MTB messages generated in the process. We tried 500 samples and Fig. 5 shows the averaged results.

The latency results of the algorithms are in Fig. 5(a). In the trace, the first few sources in additive and multiplicative dec flavors spent some time searching for their PMTUs and thus have latencies larger than those of the pre-known PMTU algorithm where exploration is not necessary. But only after a few sources (5 in our experiment), the later sources got the idea of their PMTUs and their latencies are almost the same as those of the pre-known case. Also we found that the multiplicative dec algorithm learned the PMTU value much faster than the additive dec one.

The results of the number of MTB messages generated are shown in Fig. 5(b). Since the pre-known PMTU algorithm knows the PMTU value, its number of MTB messages is zero. For the other two, the first few sources received some MTB messages and the later ones received much fewer MTB messages as they got the idea of the right data size. Again, the multiplicative dec algorithm generates much fewer MTB messages than the additive dec algorithm, which means that it can learn the PMTU value much faster.

In summary, the DSAF protocol can discover the PMTU value quickly. Please note that if node meeting duration increases, as indicated by the number of MTB messages falling below a threshold, the sources can additively increase their message sizes. And when they start to receive more MTBs again, they can reduce message sizes and go back to the situation in our experiment.

VII. Conclusion

In this paper, we have proposed a data size aware forwarding (DSAF) scheme based on PMTU discovery using MTB messages as the feedback for OMNs. Theoretical analysis and experimental results have proved its effectiveness. In the future, we will explore other ways to address the large data forwarding problem in OMNs.

References

[1] Crawdad - A Community Resource for Archiving Wireless Data At Dartmouth. http://www.crawdad.org/.

[2] R. Alkurd, R. Shubair, and I. Abualhaol. Survey on device-to-device communications: Challenges and design issues. In *IEEE International New Circuits and Systems Conference*, 2014.

[3] A. Asadi, Q. Wang, and V. Mancuso. A survey on device-to-device communication in cellular networks. *IEEE Communication Surveys & Tutorials*, 16(4), 2014.

[4] X. Deng, L. Chang, J. Tao, J. Pan, and J. Wang. Social profile-based multicast routing scheme for delay-tolerant networks. In *IEEE ICC*, pages 1857–1861, 2013.

[5] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *ACM MobiHoc*, 2008.

[6] G. Grimmett. *Probability-An Introduction*. Oxford University Press, 1986.

[7] T. Karagiannis, J. Boudec, and M. Vojnović. Power law and exponential decay of inter contact times between mobile devices. *IEEE Transactions on Mobile Computing*, 9(10), 2010.

[8] W. Li, X. Chen, and S. Lu. Content Synchronization using Device-To-Device Communication in SMART Cities. *Computer Networks*, 2017.

[9] J. Mogul and S. Deering. Path MTU Discovery. *Internet RFC 1191*, 1990.

[10] R. Parshani, S. Carmi, and S. Havlin. Epidemic threshold for the susceptible-infectious-susceptible model on random networks. *Physical Review Letters*, 104(25), 2010.

[11] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD trace cambridge/haggle/imote/infocom2006(v.2009-05-29). http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006, 2009.

[12] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Department of Computer Science, Duke University, 2000.

[13] N. Wang and J. Wu. Optimal data partitioning and forwarding in opportunistic mobile networks. In *IEEE Wireless Communications and Networking Conference*, 2018.

[14] N. Wang and J. Wu. Optimalcellular traffic offloading through opportunistic mobile networks by data partitioning. In *IEEE ICC*, 2018.

[15] Y. Wang, W.-S. Yang, and J. Wu. Analysis of a hypercubebased social feature multipath routing in delay tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(9):1706–1716, 2013.

[16] N. Weiss. *A Course in Probability*. Boston: AddisonWesley, 2005.

[17] Y. Xu and X. Chen. Social-similarity-based multicast algorithm in impromptu mobile social networks. In *IEEE Globecom*, 2014.