Efficient Quality-of-Service (QoS) Support in Mobile Opportunistic Networks

Yang Liu, *Student Member, IEEE*, Zhipeng Yang, *Student Member, IEEE*, Ting Ning, *Student Member, IEEE*, and Hongyi Wu, *Member, IEEE*

***Abstract*—This paper aims to support quality-of-service (QoS) provisioning, particularly the guarantee for end-to-end data delivery delay, in mobile opportunistic networks. The QoS-aware delivery probability (QDP) is introduced to reflect the capability of a node to deliver data to a destination within a given delay budget. Each node maintains a set of $QDPs$ to make autonomous decisions for QoS-aware data transmission. At the same time, a prioritized queue is employed by each mobile node. To support efficient prioritization and redundancy control, the priority is determined by a function of traffic class and data redundancy. The former is predetermined by the corresponding application, whereas the latter is dynamically estimated during data delivery. Two experiments are carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow sensors that are connected by air and ground mobile nodes with controlled mobility. The second experiment is under a mobile social network setting, where 23 Dell Streak Android tablets are carried by volunteers with arbitrary and diverse mobility patterns over a period of two weeks. Moreover, simulation results are obtained under DieselNet trace and power-law mobility model to study scalability and performance trends. Our experiments and simulations demonstrate that the proposed scheme achieves efficient resource allocation according to the desired delay budget and, thus, supports effective QoS provisioning.**

***Index Terms*—Mobile opportunistic networks, quality-of-service (QoS).**

I. Introduction

MOBILE opportunistic networks are characterized by intermittent and nondeterministic connectivity, often due to interruptible wireless links, sparse network deployment, and/or nodal mobility. Such opportunistic networking has been discussed in the context of delay/disruption-tolerant networks, sporadically connected sensor networks, vehicular networks, and peer-to-peer mobile social networks [1]–[7]. How to discover and utilize opportunistic communication resources for efficient data transmission has been one of the central research issues in such networks, as evidenced by extensive discussions in the literature [1]–[38]. However, limited prior work has addressed quality-of-service (QoS). While long data delivery delay is generally unavoidable given the unique intermittent connectivity, QoS, particularly the guarantee for end-to-end delivery delay, is highly desired in a variety of applications. For example, the dissemination of a data message (such as an advertisement or coupon [5], [6], [39], [40]) in a mobile social network must meet a delay budget no longer than its expiration date, and different data messages are often associated with different delay budgets. Separately, in wildlife tracking applications, interactive control and event report must be delivered within a short end-to-end delay bound, as opposed to routine transmissions of ambient environmental data that can tolerate long delay [2]. Data delivered beyond their delay budgets often lead to a reduced or completely forfeited value.

Manuscript received May 3, 2013; revised September 23, 2013 and January 21, 2014; accepted March 2, 2014. Date of publication March 12, 2014; date of current version November 6, 2014. The review of this paper was coordinated by Prof. Y. Qian.

The authors are with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70503 USA (e-mail: yxl0782@cacs.louisiana.edu; zhipengyang@hotmail.com; txn6704@cacs.louisiana.edu; wu@cacs.louisiana.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/TVT.2014.2311450

A. Challenges in QoS Provisioning in Opportunistic Networks

QoS has been extensively studied in wireless networks [41]. However, there are unique challenges to support QoS in an opportunistic communication setting. First of all, due to the nondeterministic connectivity, it is intrinsically infeasible to provide hard guarantee of end-to-end delivery delay. Thus, a probability-based delay budget is introduced in this research. More specifically, let $Q_m(\delta, \gamma)$ denote the desired QoS of message m, which must be delivered to its destination within δ time units with a probability no less than γ.

Second, since end-to-end paths often do not exist in the network, a routing decision must be made based on predicted future connections. To this end, temporal and/or spatial information in nodal contacts is exploited by mobile nodes to estimate their probabilities to deliver data to corresponding destinations [3], [5], [14]. Such delivery probability serves as a routing metric to guide data transmission, where a data message is always forwarded to nodes with higher delivery probabilities. However, most prior studies do not consider delay budget. Therefore, the delivery probabilities may become misleading for QoS support. For example, a node with a high delivery probability to a destination may in fact experience long average delay, thus deceptively attracting many data messages by following the routing scheme previously described but frequently failing to meet the desired QoS requirement.

Third, the QoS priority associated with a data message is static, i.e., does not change during its transmission, in conventional networks. However, redundancy is commonly employed in opportunistic networks for dealing with a high data loss probability and achieving a desired delivery rate. Consequently, the importance of a message varies during its transmission, depending on the amount of redundancy created. For example, a newly generated message is the sole copy and should be processed with high priority and protected from being lost. When multiple copies of the message are produced during its transmission, deferring or losing a copy would not significantly degrade the delivery probability of the message. In general, data messages that are in the same traffic class may have diverse redundancy (even if they were created by the same node at the same time) and, accordingly, should be associated with different priority levels. Therefore, data messages must be prioritized not only by QoS requirement but according to their dynamically changing redundancy as well.

B. Contribution of This Work

This work proposes an effective solution to the challenges previously discussed. The QoS-aware delivery probability (QDP) is introduced to reflect the capability of a node to deliver data to a destination within a given delay budget. Each node maintains a set of $QDPs$ to make autonomous decisions for QoS-aware data delivery. At the same time, a prioritized queue is employed by each mobile node. To support efficient prioritization and redundancy control, the priority is determined by a function of traffic class and data redundancy. The former is predetermined by the corresponding application, whereas the latter is dynamically estimated during data delivery.

Two experiments are carried out to demonstrate and evaluate the proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow Micaz sensors that are connected by three mobile nodes carried respectively by a flying vehicle (with high mobility) and two people (with low mobility). All mobile nodes move according to predefined routes and speeds. The second experiment is under a mobile social network setting, where the experimental program is implemented on Dell Streak Android tablets carried by 23 volunteers with arbitrary and diverse mobility patterns for a period of two weeks. Moreover, simulation results are obtained under DieselNet trace and power-law mobility model to study the scalability and performance trend with the increase in network size, traffic load, and nodal mobility. The experimental and simulation results show that the contact opportunity among nodes is decisive for the overall network performance, verifying the observations reported in earlier work [1]–[3], [5], [14]. The proposed scheme achieves efficient resource allocation according to the desired delay budget and, thus, supports effective QoS provisioning. When the network becomes heavily loaded, it gracefully degrades QoS performance, in contrast to Best Effort routing that often results in dramatical performance downgrade.

II. Related Work

QoS is a critical issue in mobile networks, with a diversity of approaches recently proposed [42]–[47]. However, none of them are developed for QoS support in mobile opportunistic networks. Given the unique characteristics of mobile opportunistic networks (particularly the intermittent nondeterministic network connectivity), solutions for QoS support in conventional networks are not applicable here. In general, QoS provisioning in mobile opportunistic networks is a less-studied area with limited existing solutions.

In opportunistic networks, the delivery of a single-copy data message is often subject to a high loss rate or extremely long delay. Therefore, redundancy (achieved by duplication or coding [3]) is commonly employed for desired communication performance. However, redundancy increases overhead, and worse yet, excessive redundancy may degrade overall network performance due to congested channels and frequent buffer overflow [27], [48]. To this end, a series of approaches [8]–[14] has been developed to limit redundancy for efficient resource utilization. Among them, [14] is most relevant to this work, where a node sorts its packets in the queue in decreasing order of the ages and makes routing decisions according to a marginal utility function to improve the probability of delivering packets within their deadlines. While this scheme maximizes the overall deadline-satisfied delivery rate, it does not differentiate traffic and is not equivalent to the support of QoS. For instance, a data flow with high QoS priority can be submerged by a large volume of background low-priority data that have stayed in the network long enough and, thus, occupied the head of queue. As a result, the transmission of high-priority data is delayed, resulting in poor QoS provisioning. Moreover, a “control channel” is required in [14] to timely share global information among nodes, but such a channel is not always available in a practical mobile opportunistic network.

Separately, there are a handful of studies dedicated to QoS support in delay-tolerant networks [49]–[51]. However, they are either based on simplified settings or operate at the level of individual links only. The bundle protocol developed for delay-tolerant networks supports class-of-service [2], which is reliability-centric, aiming to ensure correct data transmission but does not support delay constraints. In addition, delay budget is considered in [26], which mainly concerns incentive provisioning based on pairwise tit-for-tat (TFT). More detailed comparison with related work will be presented in Sections IV and V.

III. Proposed Quality-of-Service-Aware Delivery Scheme

To address the unique challenges in QoS provisioning in opportunistic networks with intermittent and nondeterministic network connectivity, we propose a QoS data communication scheme based on QoS-aware delivery probability and adaptive queue prioritization, as outlined below. The former serves as the QoS routing metric, which guides a data message through the best routing path that meets the desired QoS requirement with high probability. The latter supports efficient resource utilization by proper redundancy control.

A. QDP

As discussed in Section I-A, we adopt a probability-based delay budget, denoted by $Q_m(\delta, \gamma)$, as the QoS metric, demanding message m to be delivered to its destination within δ time units with a probability no less than γ. A node often has a number of messages in its data queue. It may transmit a message directly to the destination or to an intermediate node, which subsequently continues to forward the message directly or indirectly to the destination. When a node meets another node, the former needs to decide whether to transmit a message to the latter. Such a routing decision must be made based on a QoS-aware routing metric, which indicates if the latter has a higher probability to deliver the message to its destination within the delay budget. To this end, we introduce a new routing metric for QoS provisioning in DTN, dubbed QDP.

Since a data message may be associated with any arbitrary delay budget and to any destination, it is imperative for a node to maintain a set of $QDPs$ to make autonomous decisions for QoS provisioning. Let $P_i = \{p_i^k(t) | 0 \leq t \leq \infty, k \in \Phi\}$ denote the $QDPs$ of node i, where $p_i^k(t)$ is the probability that a message can be delivered from node i to node k within t time units, and Φ is the set of DTN nodes. For a given k, $p_i^k(t)$, $0 \leq t \leq \infty$, is intrinsically the cumulative distribution function of delivery delay, which is ideal to support QoS data delivery but impractical to maintain in continuous time. Thus, a finite set of discrete delays, denoted by Υ, is employed, arriving at $P_i = \{p_i^k(t) | t \in \Upsilon, k \in \Phi\}$.

While QDP is previously defined, it is obviously challenging to be obtained in a distributed manner, since a node is connected to other nodes only occasionally. With no end-to-end connections, it is extremely difficult, if not impossible, to gain up-to-date global knowledge to compute accurate $QDPs$. However, at the same time, we notice that the accurate $QDPs$ are, although desired, not imperative. As a matter of fact, the QDP can be over- or underestimated across the network, due to the approximation in QDP update. The approximate $QDPs$ can effectively support QoS routing as long as they are proportional to the real $QDPs$, i.e., a node with a truly higher (or lower) QDP maintains a higher (or lower) approximate QDP. Despite such $QDPs$ being inaccurate, they efficiently guide data messages through the best routing paths for QoS provisioning. This observation is verified by our simulations.

To this end, we propose a lightweight distributed algorithm to learn approximate $QDPs$. The overall idea is to let individual nodes maintain their approximate $QDPs$, which are updated based on locally learnt information upon meeting events. Initially, a node only knows the QDP with itself as the destination, which is obviously one. It learns the $QDPs$ to other destinations via recursive information exchange, during which the QoS delivery probabilities are updated in a ripple manner propagated from the corresponding destinations. More specifically, node i initializes $p_i^k(t)$ as follows:

$$p_i^k(t) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases} \tag{1}$$

and updates them autonomously according to its transmission history, in both direct and cascaded deliveries, as outlined below. Each node divides time into windows. The size of a window can be chosen to be the maximum delay budget interested by the node. The windows of different nodes do not have to be synchronized. This is because each node updates its QDP autonomously. It can choose any arbitrary window size and start its window at an arbitrary time instance. The process does not require synchronization among different nodes.

$QDPs$ are updated based on time windows. Let us consider node i. It maintains a parameter $\xi_i^k(t)$, which is used to calculate the QDP of node i to node k in each window, for each $k \in \Phi$ and $t \in \Upsilon$. It intrinsically indicates the probability that the message fails to be delivered to the destination within the delay budget in a time window. In each time window, $\xi_i^k(t)$ is initialized to 1. When node i meets node j, it compares $p_i^k(t)$ and $p_j^k(t)$ for every $k \in \Phi$ and $t \in \Upsilon$. If $p_i^k(t) < p_j^k(t)$, the former transmits the corresponding message with destination k and delay budget t to the latter and, at the same time, updates $\xi_i^k(t)$ as follows:

$$\xi_i^k(t) \leftarrow \xi_i^k(t) \left(1 - p_j^k(t)\right) \tag{2}$$

where $1 - p_j^k(t)$ is the probability that node j cannot deliver the message to destination k within the required delay budget of t. If $j = k$, it is a direct delivery. According to (1), $p_j^k(t) = 1$, and thus, $1 - p_j^k(t) = 0$. Otherwise, it is an indirect delivery where node j may or may not successfully transmit the message to its destination. Consequently, node i cannot receive a confirmation immediately from node j. Therefore, it estimates the probability that node j delivers the message by $p_j^k(t)$.

By the end of the window, node i calculates its window-based QDP as

$$\hat{p}_i^k(t) = 1 - \xi_i^k(t) \tag{3}$$

which essentially equals $1 - \Pi(1 - p_j^k(t))$, i.e., the probability that at least one of such transmissions delivers the message to its destination by its delay budget. Clearly, once node i delivers the message directly to its destination (i.e., node k), $\hat{p}_i^k(t)$ becomes one, and there is no need to further transmit the message.

$\hat{p}_i^k(t)$ is a window-based QDP. Its value often varies from window to window, exhibiting undesired instability. It is highly preferable to keep $QDPs$ stable, since they are employed to guide data transmission. In this research, we adopt the exponentially weighted moving average (EWMA) to maintain and update $QDPs$. More specifically, we have

$$p_i^k(t) \leftarrow (1 - \mu) p_i^k(t) + \mu \hat{p}_i^k(t) \tag{4}$$

where $0 \leq \mu \leq 1$ is a constant weight to keep partial memory of historic status. It has been shown in [52] that the given EWMA-based average converges to a constant under statically distributed mobility. $p_i^k(t)$ indicates the probability that node i delivers a message to node k within a delay budget of t. Node i performs similar calculation for all k and t to yield the QDP matrix $P_i = \{p_i^k(t) | t \in \Upsilon, k \in \Phi\}$.

Every node follows the given algorithm to learn its $QDPs$. We observe in our simulations that, since node i is only aware of $p_i^i(t)$ (i.e., with itself as the destination) during initialization, $QDPs$ are updated in a ripple manner starting from the corresponding destinations. While the simulation details are deferred to Section V, Fig. 1 shows that the converged QDP indeed reflects the true QoS-aware delivery probability after the warm-up period of simulation, thus serving as an efficient routing metric for QoS provisioning.

B. Adaptive Data Queue Prioritization

For the sake of low complexity in queue management, most QoS-aware systems (e.g., IEEE 802.11e) employ multiple "first-in first-out" queues: one for each traffic class. In opportunistic networks, however, it is often inevitable and, at the

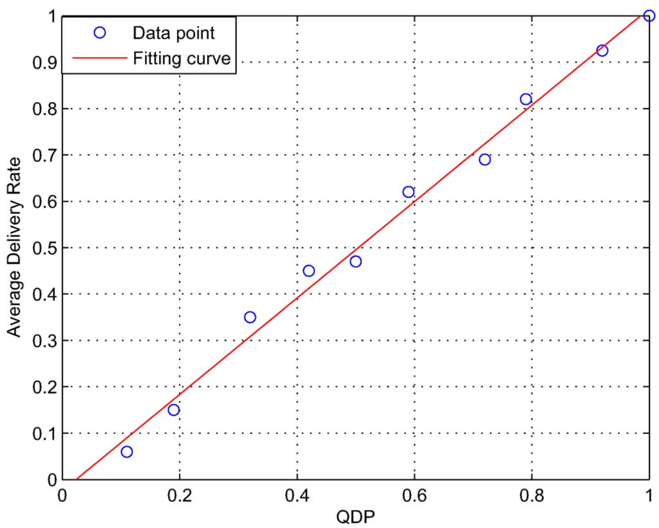


Fig. 1. QDP versus delivery rate (average of $t \in \Upsilon$).

same time, affordable to deal with more complicated queuing strategies. In contrast to conventional store-and-forward networks where only a single copy of a data message is actively transmitted at a time, redundancy (by either simple duplication or coding) is commonly employed for desired communication performance, particularly for QoS support, in opportunistic networks. However, the creation and distribution of redundancy depend on nondeterministic nodal meeting events, thus exhibiting high dynamics. At a given time, data messages in the same traffic class may have very different redundancy. For example, multiple copies may have been created for a message that has been circulated in the network for some time, and thus, its preferred delivery probability can be maintained even if a copy is delayed or dropped. On the other hand, a newly created message may be the sole copy that should be absolutely protected from being lost. Consequently, data messages in the same traffic class must be prioritized according to their redundancy level, naturally leading to prioritized queues with a logarithm time complexity for message insertion and removal. Although such computing time is undesired and may become performance bottleneck in conventional networks, it is affordable in opportunistic networks, where a node has plenty of time to manipulate its queue before the next transmission opportunity becomes available.

Based on the given observations, a single prioritized queue is employed by each node. The queue is sorted according to the priority of data messages, which is a function of traffic class and redundancy. The former is predetermined by the corresponding application and remains unchanged during the transmission of the messages, whereas the latter is dynamically estimated, which is to be discussed next. More specifically, let q_i^m denote the priority of message m in the message queue of node i. A smaller q_i^m indicates a higher priority. q_i^m is calculated as follows:

$$q_i^m = (1-\lambda)C_i^m + \lambda F_i^m \tag{5}$$

where C_i^m denotes the traffic class of message m, F_i^m is the redundancy level of message m estimated by node i, and $0 \leq \lambda \leq 1$ is a constant to balance the weight of traffic class and redundancy for queue prioritization. λ is a tunable parameter that can be determined according to specific application needs. We simply set it to 1/2 in our implementation.

As discussed earlier, while the traffic class [i.e., C_i^m in (5)] is known, the redundancy of a message (i.e., F_i^m) needs to be dynamically estimated. In a typical store-and-forward network, messages are deleted from a node's buffer after they are transmitted to the next hop successfully. In opportunistic networks, however, multiple copies of the data message are often created and stored by different nodes in the network, to maintain necessary redundancy for achieving the desired QoS. In general, the higher the redundancy, the higher the message delivery probability when network capacity is not a concern. In this paper, we define the redundancy of a message as the estimated probability that at least one copy of message m is delivered to its destination within t time units, where t is updated during transmission to reflect the up-to-date remaining delay budget.

The redundancy of a message (F_i^m) is initialized to zero and updated during its transmissions. Consider a general scenario where node i has an opportunity to communicate with node j. It fetches message m that is destined to node k. First, the delay budget of message m is updated as $t = t - \tau$, where τ is the time for which the message has stayed in the queue. Node i simply transmits the message to node j and removes it from its queue if $j = k$. Otherwise, if $F_i^m \geq \beta$, where β is a predefined desired delivery probability, no action will be taken, because there is already sufficient redundancy. Node i will hold the message until it encounters the destination directly or the delay budget expires. If the delay budget expires, node i simply discards the message from its queue. Generally, the more the redundancy, the higher the probability. $F_i^m \geq \beta$ means that current redundancy (e.g., number of copies of message m) is large enough to ensure a delivery probability no less than β. Note that, even with $F_i^m \geq \beta$, there is no guarantee that message m will be delivered to the destination within the delay budget. However, if we look at a large number of such data messages, the protocol delivers them with an overall probability of β, thus achieving our goal. If $F_i^m < \beta$ and $p_i^k(t) < p_j^k(t)$, the message is transmitted to node j. This transmission creates two copies of message m, each sharing partial responsibility to deliver the data to its destination. Appropriate redundancy needs to be assigned to them, i.e.,

$$F_j^m \leftarrow 1 - (1 - [F_i^m]) \left(1 - p_i^k(t)\right) \tag{6}$$
$$F_i^m \leftarrow 1 - (1 - [F_i^m]) \left(1 - p_j^k(t)\right). \tag{7}$$

In both formulas, $(1 - [F_i^m])$ gives the probability that none of the other nodes (except nodes i and j) can deliver the message, where $[F_i^m]$ is the value before it is updated due to this transmission. Therefore, the updated F_j^m (or F_i^m) indicates the probability that at least one copy of message m can be delivered by other nodes except node j (or node i). In general, the more times a data message is forwarded, the more redundancy is created. For example, Fig. 2 shows that average message redundancy grows with the message's life span (where the simulation details are to be discussed in Section V). At the same time, a node often holds a set of data messages in its queue. Fig. 3 shows that their redundancies largely fall into a normal distribution.

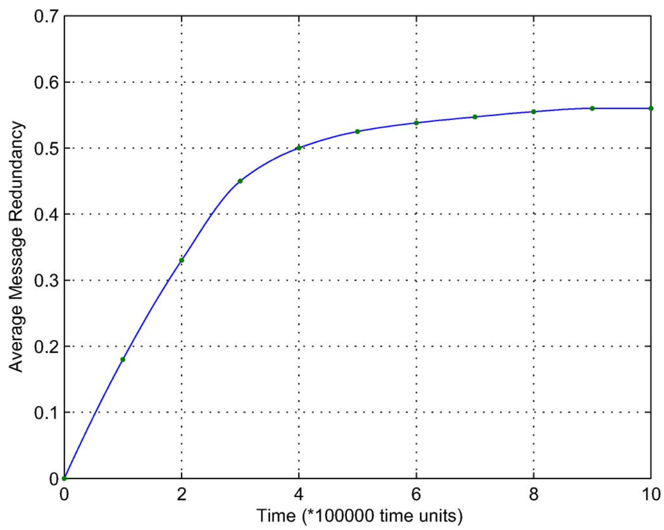


Fig. 2. Evolvement of redundancy over time.

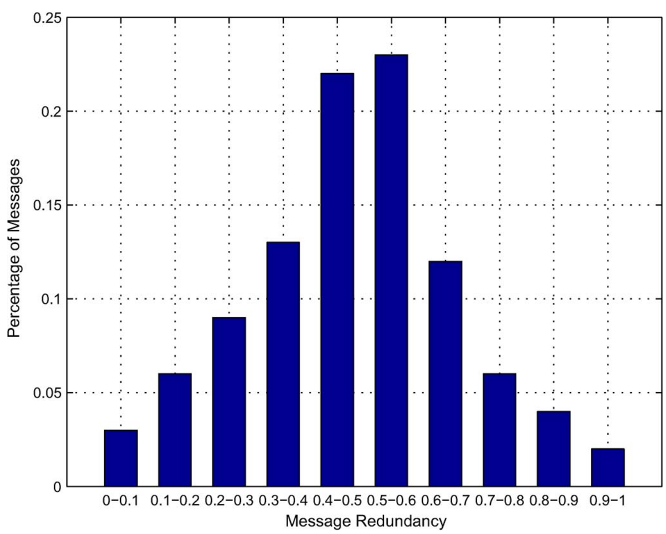


Fig. 3. Distribution of message redundancy.

The given scheme is efficient in redundancy control and queue management, limiting redundancy to be just enough to achieve the desired delivery probability, i.e., β. How to choose the optimal β still remains an open issue. It is affected by such parameters as nodal contact probabilities, maximum queuing capacity, and traffic load and patterns. In a specific scenario, we can run simulations to identify the approximate optimal β.

C. QoS-Aware Data Delivery

Since communication opportunity is low, transmission is often between two nodes only. If more than two nodes are within communication range, we assume an underlying medium access control protocol (e.g., IEEE 802.11) that randomly selects one node as the sender and another as the receiver. Therefore, we focus on the scenario where node i transmits a data message to node j in the following discussions. Node i first learns the $QDPs$ of node j (i.e., P_j) via two-way handshaking. Then, it fetches the first message in its queue, which is denoted by message m to destination k and with a remaining delay budget

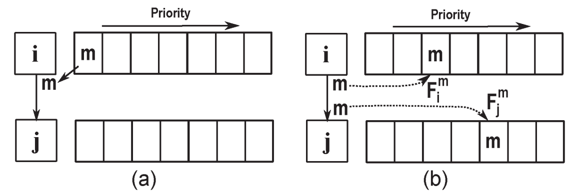


Fig. 4. Example of transmission between nodes i and j. (a) Before transmission. (b) After transmission.

TABLE I
EXPERIMENT PARAMETERS

	Experiment I	Experiment II
Number of nodes	39	23
Buffer size	50	1000
Duration	3.5 mins	2 weeks
Message size	10 KB	1 KB-1 MB
Message generation rate	1 per 5 seconds	20 per day
Delay budget	3-90 seconds	4 hours-3.5 days

of t. If $t = 0$, node i simply drops the message. Otherwise, if node j is the destination (i.e., $j = k$), node i transmits the message to node j and removes it from its queue. If node j is not the destination, node i transmits message m to node j if and only if $F_i^m < \beta$ and $p_i^k(t) < p_j^k(t)$.

Upon a message transmission, two copies of the message are created, with their redundancies calculated according to (6) and (7) and their priorities updated by (5), respectively. Then, both nodes insert their copies into their data queues according to the updated priorities (see Fig. 4). If a queue is full, the message at the end of the queue (i.e., the message with the lowest priority) is dropped.

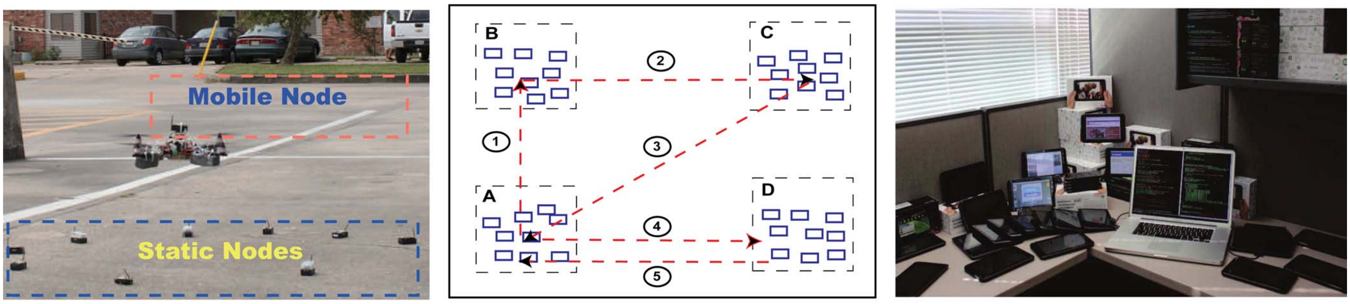
The given process repeats with a randomly chosen node as the sender, until the communication link is broken (e.g., due to nodal mobility), or no messages are available for transmission.

D. Complexity Analysis

In general, the computational complexity at individual nodes is linear to the network size. More specifically, the communication between two nodes has a computational complexity of $O(k)$, where k is the buffer size of the message queue of each node. Update of $QDPs$ for each node at the end of each time window has a complexity of $O(nl)$, where n is the total number of nodes in the network, and l is the number of delay budget levels. Therefore, the overall time complexity of the proposed QoS-aware delivery algorithm is $O(k + nl)$. In addition, the overall space complexity of the proposed QoS-aware delivery algorithm is $O(nl)$. Note that k and l are often constant values. Therefore, the time complexity is essentially $O(n)$, and the space complexity is $O(n)$ as well.

IV. PROTOTYPING AND EXPERIMENTS

To demonstrate the feasibility of our proposed QoS-aware data delivery scheme and gain useful empirical insights, we have carried out two sets of experiments using the off-the-shelf Crossbow Micaz motes and Dell Streak Android tablets, respectively. The first experiment involves multiple clusters of static sensors that are connected by a small set of mobile



(a) A cluster of ground sensors in the experimental field and the flying quadrotor that carries a sensor.

(b) Topology of the sensor network testbed and the moving pattern of the quadrotor.

(c) The Android tablets used in the mobile social network experiment.

Fig. 5. Testbed setup. (a) and (b) Experiment I. (c) Android tablets used in Experiment II. (a) Cluster of ground sensors in the experimental field and the flying quadrotor that carries a sensor. (b) Topology of the sensor network testbed and the moving pattern of the quadrotor. (c) Android tablets used in the mobile social network experiment.

nodes with regular movement patterns, whereas the second experiment is under the setting of a mobile social network where the nodes have diverse and uncontrolled mobility. The reason we carry out the two experiments is that they are representative examples in mobile opportunistic networks that need QoS support. For both of the experiments and the simulation presented in Section V, the first fraction of system time is used to warm up for nodes to accumulate $QDPs$. Only source address, destination address, and delay budget are generated during this period; data are generated and forwarded during the remaining part of time. Table I summarizes the experimental settings and configuration parameters. Section IV is organized as follows. We present two experiments in Sections IV-A and B, respectively. In each subsection, we first introduce the testbed setup and configuration and then discuss experimental results and related observations.

A. Experiment I: Opportunistic Sensor Network

1) Testbed Setup: Our sensor network testbed consists of 36 static Crossbow Micaz sensors and three mobile nodes. The static sensors are randomly deployed at four corners of a parking lot, forming four isolated clusters [see clusters A, B, C, and D in Fig. 5(b)]. The sensors in a cluster are well connected. We created such an experiment setting because it is common under practical applications. For example, when biologists study animals in a field, it is obviously too costly to cover the entire field with sensors. However, they are often interested in several target spots and, thus, can deploy a cluster of sensors at each spot. The sensors in a cluster are closely located, within radio communication range. Therefore, they are well connected. At the same time, the distance between any two clusters is farther than the maximum radio transmission range of Micaz, and hence, the clusters are isolated, calling for mobile nodes to carry data between them.

The mobile sensors act as Datamule [15] for data transmission among the sensor clusters. They are carried by a quadrotor and two students, with high and low mobility, respectively. The quadrotor is built upon the Mikrokopter platform. It can fly up to several hundred meters high and at a speed between 0 and 40 km/h and, thus, is well suitable for remote sensor fields. Fig. 5(a) shows the ground Micaz motes and the mobile node on a quadrotor. In our experiment, the quadrotor is controlled by an independent remote controller and commutes among the four sensor clusters. It flies according to a predetermined route, as shown in Fig. 5(b). More specifically, it begins its journey from cluster A; then sequentially visits clusters B, C, A, and D; and finally returns to cluster A. It repeats the above routine flight during the experiment. The average flying height is about 2.2 m based on the barometer readings on the flying platform. We say the quadrotor visits a cluster if its onboard sensor can communicate with any sensor in the cluster. The average time between it visits two adjacent clusters is 12 s. The average waiting times (i.e., the average interval for a cluster to meet the quadrotor) at clusters A, B, C, and D are 10, 20, 10, and 15 s, respectively. The other two mobile nodes are carried by two students. One student moves in clockwise direction and the other counterclockwise. It takes about 3.5 min to complete a round of visits to the four clusters. Note that, although the routes are predetermined, the communication is opportunistic due to dynamic mobility (e.g., unavoidable dynamics in moving speed and height) and varying channel conditions.

Each ground sensor generates a data message of 10 KB every 5 s to a randomly selected destination. A message is associated with a delay budget between 3 and 90 s.

2) Experimental Results: For performance comparison, we have implemented four schemes, dubbed QoS-aware, Best Effort, DDG, and Incentive, respectively. QoS-aware is the proposed QoS-aware delivery scheme; Best Effort is a delivery protocol without QoS support [3], which makes the decision on when and where to transmit data messages only according to the delivery probability. It is not a surprise to find the Best Effort approach results in a low delivery rate because it does not differentiate traffic at all. DDG is the Delay-Differentiated Gossiping approach [49], which considers multiple traffic classes and dynamically assigns the packets in each class a transmission probability and a time-to-live, which, together, govern the total overhead for data transmission. Although DDG supports QoS provisioning, its data transmission is randomized. Therefore, a packet is often delivered via a long path and, consequently, subject to high dropping probability due to the expiration of its delay budget. Incentive is the pairwise TFT approach [26], which adopts an incentive-aware routing protocol that allows selfish nodes to maximize their own performance while conforming to

TABLE II
RESULTS OF EXPERIMENT I (OPPORTUNISTIC SENSOR NETWORK)

	QoS-aware	Best Effort	DDG	Incentive
Avg. Del. Rate (overall)	72%	38%	42%	58%
Avg. Del. Rate (3 Sec)	2%	2%	2%	2%
Avg. Del. Rate (5 Sec)	8%	3%	5%	6%
Avg. Del. Rate (10 Sec)	32%	5%	16%	26%
Avg. Del. Rate (60 Sec)	98%	96%	97%	97%
Avg. Del. Rate (90 Sec)	100%	100%	100%	100%

TFT constraints. The Incentive scheme exhibits an unsatisfied delivery rate because of its "candidate path generation" process, which results in high overhead under our experiment setting. At the same time, it does not consider different delay budgets for different messages when formulating the linear programming model. For fair comparison, trace data are collected to run comparable schemes.

We are primarily interested in the data delivery rate. A message is delivered if it reaches its destination within its delay budget. The data delivery rate is defined as the ratio of the total number of delivered messages to the number of generated messages. Table II shows the overall average delivery rate and the delivery rates for messages with delay budgets of 3, 5, 10, 60, and 90 s, respectively. As can be seen, the proposed QoS-aware scheme achieves an overall delivery rate of 72%, which is significantly higher than other approaches. It is not a surprise to find the Best Effort approach results in a low delivery rate because it does not differentiate traffic at all. DDG considers multiple traffic classes and dynamically assigns the messages in each class a transmission probability and a time-to-live, which, together, govern the total overhead for data transmission. However, as a gossiping approach, its data transmission is randomized. Therefore, a message is often delivered via a long path and, consequently, subject to high dropping probability due to the expiration of its delay budget. The Incentive scheme exhibits an unsatisfied delivery rate because of its "candidate path generation" process, which results in high overhead under our experiment setting. At the same time, it does not consider different delay budgets for different messages when formulating the linear programming model.

We also observe that the proposed QoS-aware data delivery scheme achieves the best performance compared with other schemes under moderate delay constraints. When the delay budget is extremely low, the QoS-aware scheme does not improve the performance because there is simply no way to deliver the data message within its time budget. As the delay budget increases, the QoS-aware scheme shows superior performance since it prioritizes messages and allocates resource to deliver more urgent and possibly deliverable data messages. It is obvious that, if the delay budget is very high, most data messages can always be delivered, no matter which scheme is employed.

B. Experiment II: Mobile Social Network

1) Testbed Setup: The second experiment is carried out under a mobile social network setting that involves 23 volunteers including faculty members, senior Ph.D. students (who do not have classes), and graduate students at the M.S. level (who go to classrooms regularly). A mobile social network is often created for a local community where the participants have frequent interactions, e.g., people living in a neighborhood, students studying in a college, or tourists visiting an archaeological site. It exploits Bluetooth and WiFi connections to form a sparse ad hoc network to support social networking. This is in a sharp contrast to web-based online social networks that rely on the Internet infrastructure (including cellular systems) for communication.

Unlike the first experiment where the mobile nodes move under regular patterns, the volunteers in this experiment have arbitrary and diverse mobility patterns. Every volunteer carries a Dell Streak 5 or Streak 7 tablet [see Fig. 5(c) for a photo of the tablets used in the experiment], which operates on Android 2.2. The mobile nodes are paired and ready to communicate with each other via Bluetooth. To save power, a service is created, which runs on background to adaptively adjust the scanning frequency of the Bluetooth interface. The default scanning interval is set to 10 min during nighttime and 1 min during daytime. A node creates 20 data files everyday with the file size varying from 1 KB to 1 MB and time budget ranging from 4 h to 3.5 days. The destination for a data file is randomly selected. The experiment lasts two weeks.

2) Experimental Result: Similar to Experiment I, the proposed QoS-aware scheme outperforms other comparable schemes, i.e., Best Effort, DDG, and Incentive, in this mobile social network experiment. The results are omitted here for conciseness. Instead, we focus on investigating the impact of human activity on QoS-based data delivery. Fig. 6(a) shows the average delivery rate for data files with different delay budgets. Clearly, the larger the delay budget, the higher the delivery rate. When the delay budget reaches three days, an average delivery rate of more than 80% can be achieved.

Fig. 6(b) presents a detailed look into daily experimental data. More specifically, it shows the delivery rate of data files generated on different days during a week. As can be seen, data files generated during weekend always have a lower delivery rate than those on weekdays. This is due to the low interactive activities of students and faculty on a Saturday and Sunday. As a result, many data files cannot be delivered in a timely manner and, eventually, must be dropped due to their limited delay budgets. The data delivery rate on a Friday is lower than on other weekdays because no classes are scheduled on Friday afternoon, and many offices are closed after 1:00 P.M.

Fig. 6(c) further zooms in to show the delay of data files generated from the first to the 24th hour of a day. The second Tuesday of our experiment is chosen as an example, while similar results are observed on other days as well. The delivery rate is high during daytime and low at night, which again shows that the QoS-aware delivery scheme heavily depends on nodal mobility.

C. Further Discussion

The buffer size indicates the maximum messages each of the nodes in both of the experiments can have in its message queue. The buffer size of Experiment I is much smaller than that of Experiment II because Crossbow Micaz sensors are used in

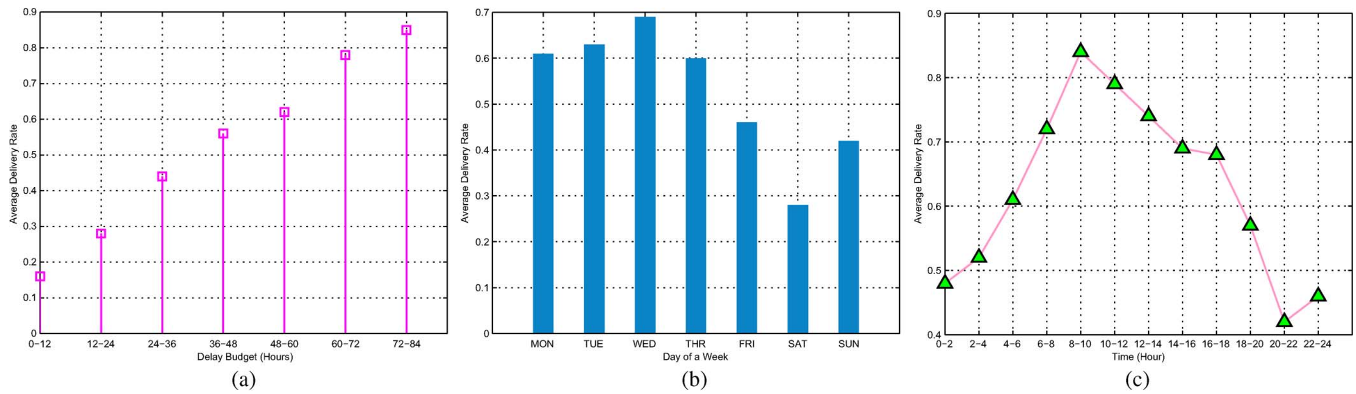


Fig. 6. Results of Experiment II (mobile social network). (a) Delivery rate under different delay budgets. (b) Delivery rate distribution during a week. (c) Hourly delivery rate on the second Tuesday.

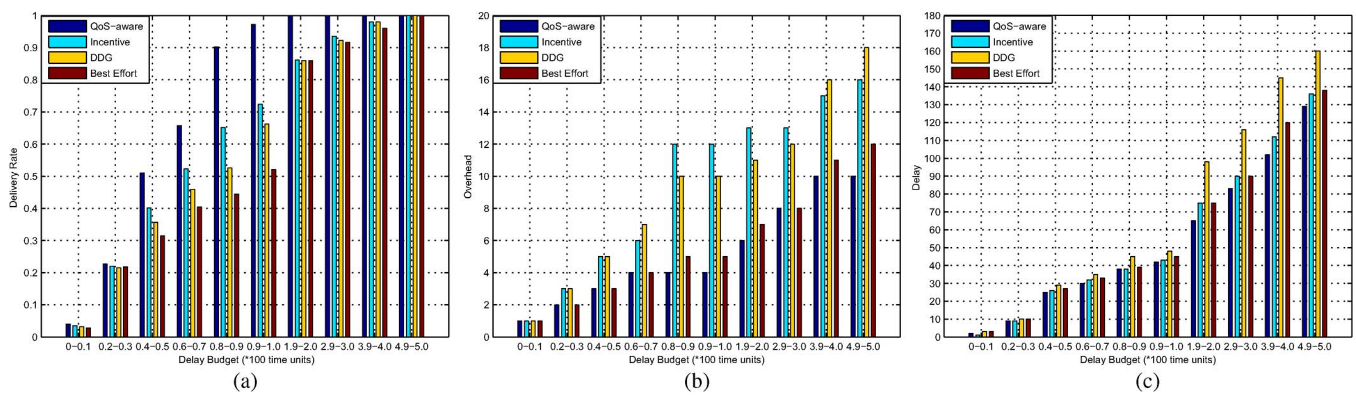


Fig. 7. Performance comparison under the DieselNet trace. (a) Delivery rate distribution. (b) Overhead distribution. (c) Delay distribution.

Experiment I, which is a tiny wireless measurement system with a low-power microcontroller with very limited storage, whereas Experiment II used Dell Streak Android tablets with very large internal and external storage. The second experiment is carried out under a mobile social network setting that involves 23 volunteers including faculty members, senior Ph.D. students (who do not have classes), and graduate students at the M.S. level (who go to classrooms regularly). Based on the mobile node's social roles (e.g., professions), interests, and available resources and messages' utilities, each node creates messages with the size varying from 1 KB to 1 MB. For example, a professor may deliver homework to students, which needs larger message size, whereas a student may send sport news to his friends, which requires smaller message size. The messages are randomly generated; hence, there is no average message size. The second experiment is under a mobile social network setting, where 23 Dell Streak Android tablets are carried by volunteers with arbitrary and diverse mobility patterns during a period of two weeks. Unlike the first experiment where the mobile nodes move under regular patterns, the volunteers in this experiment have arbitrary and diverse mobility patterns. Therefore, it will not take as long for the messages generated in Experiment II to be delivered to the destinations. All the parameters we set for the two experiments are all based on the characteristics of the two experiments.

Overall, our experimental results demonstrate that the nodal mobility and, accordingly, the contact opportunity among nodes is decisive for the overall network performance, as already revealed in earlier work [1]–[3], [5], [14]. The proposed scheme efficiently allocates resources according to delay budgets of data messages and, thus, supports effective QoS provisioning, achieving a significantly higher delivery rate in comparison with other schemes.

V. Simulation Results

In addition to the experiments previously presented, we carry out two separate simulations. The first is based on DieselNet trace to study QoS provisioning in vehicular networks. The second simulation is performed under a power-law mobility model for evaluating the scalability of the proposed scheme with the increase in network size, traffic load, and nodal mobility.

A. Simulation Under DieselNet Trace

The DieselNet testbed comprises 33 buses, serving an area of approximately 150 mi^2. Each bus carries a node with WiFi. Our simulation is based on the trace data obtained in 2008 [53]. As shown in Fig. 7(a), the proposed QoS-aware scheme achieves the highest data delivery rate under all delay budgets. For example, it delivers more than 90% of messages with delay budget between 90 and 100 time units (minutes), in comparison with 52% in the Best Effort approach, 66% in the DDG approach, and 72% in the Incentive approach. At the

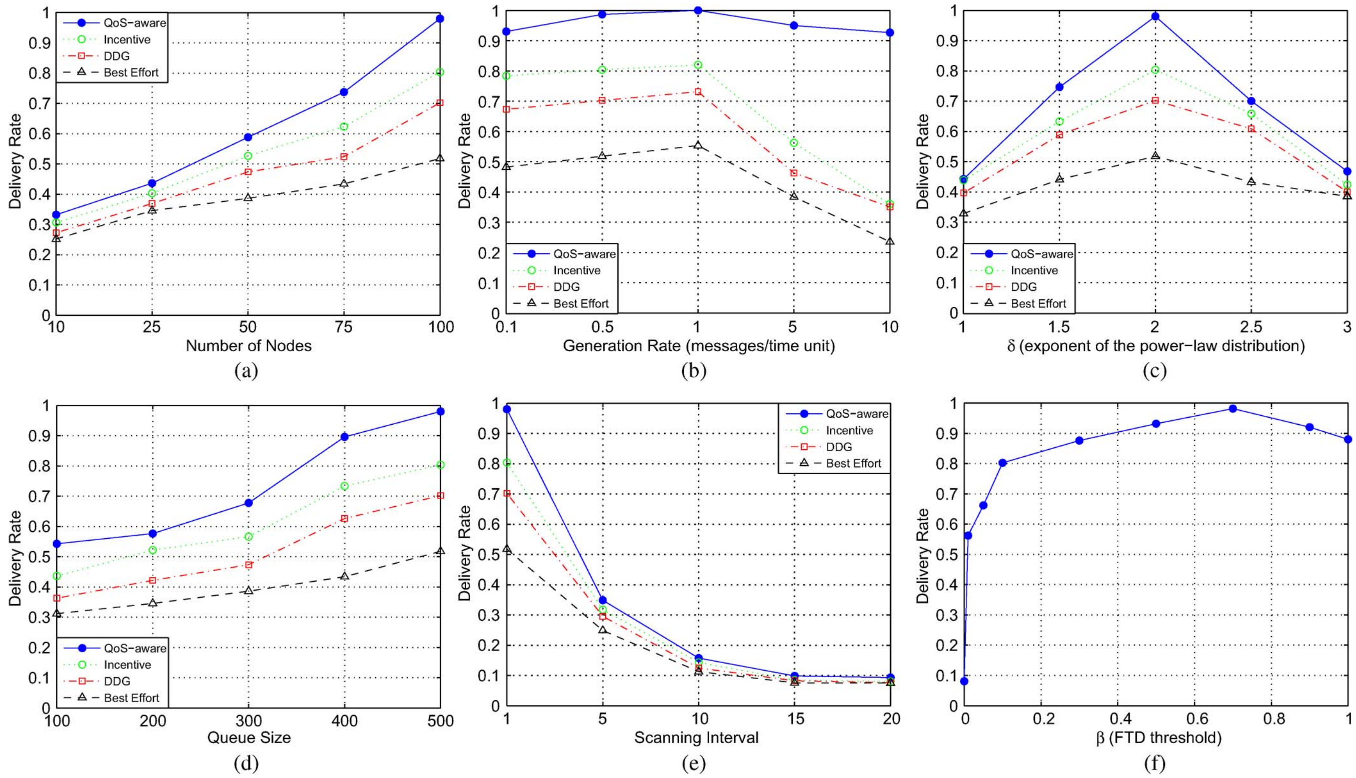


Fig. 8. Performance trend under power-law mobility model. The results are obtained for data messages with delay budget of 90–100 time units. (a) Network size. (b) Traffic load. (c) Nodal mobility. (d) Queue size. (e) Scanning frequency. (f) FTD threshold.

same time, it controls transmission overhead well, as shown in Fig. 7(b). To deliver a data message, an amount of redundancy (i.e., a number of copies) of the message is generated, hoping that at least one of them can reach the destination. The overhead is the average number of such copies per message. The high efficiency of the QoS-aware scheme is attributed to the fact that the estimated QDP enables efficient use of communication resource (i.e., the capacity of nodes and their meeting opportunities) and that the adaptive prioritization scheme supports effective queue management and redundancy control. With the increase in delay budget, the delivery rate increases accordingly under all approaches, because data messages have more time and better chance to reach their destinations. In addition, we observe similar average delay under all schemes [see Fig. 7(c)]. However, note that the average delay is calculated for delivered messages only. Due to a low delivery rate in the Best Effort, DDG, and Incentive approaches, many messages that in fact experience long delays are not included in the calculation.

B. Performance Under Power-Law Mobility Model

Our experiments and trace-based simulation have provided a comprehensive evaluation of the proposed scheme in several practical settings including sensor networks, vehicular networks, and mobile social networks. We next present simulation results based on a power-law mobility model, which offer a valuable performance trend by scaling several network parameters.

The simulated area is divided into a grid of 10 × 10 cells. Each node has a home cell where it initially locates and moves according to power-law distribution, which is deemed a realistic model for human mobility. Two nodes only communicate if they are in the same cell. Let $P_i(x)$ denote the probability for node i to be at cell x. $P_i(x) = k_i(1/d_i(x))^{\delta}$, where k_i is a constant, δ is the exponent of the power-law distribution, and $d_i(x)$ denotes the distance between cell x and node i's home cell. Under this model, δ is the key parameter governing node behavior. When δ is large, nodes tend to move among a very small subset of cells. With the decrease of δ, the moving range becomes wider. By default, we set the maximum queue size to be 500. The message generation of each node follows a random process with an average interval of 30 time units out of 100 time units. The fault-tolerant degree (FTD) threshold is set to be $\beta = 0.7$. The results are shown in Fig. 8 for data messages with a delay budget of 90–100 time units. Results under other delay budgets exhibit a similar trend and, thus, are omitted here.

Fig. 8(a) shows the performance of different schemes by varying the number of nodes in the network. With the increase in network size, nodes have more opportunities to meet each other and to reach the destination. Thus, the messages have better chances to be delivered within their delay budgets. This explains why the delivery rate of all schemes increases.

With the increase in the message generation rate, the proposed QoS-aware delivery scheme exhibits graceful degradation in the data delivery rate [see Fig. 8(b)], because it differentiates traffic and makes efficient use of communication and storage resources to meet the QoS needs. For example, when more messages with low delay budgets are generated, the protocol postpones the transmission of some messages with long delay budgets, such that more messages are delivered within their delay budgets in total. On the other hand, Best Effort, DDG, and Incentive approaches do not effectively support traffic differentiation, thus suffering dramatic decrease in the delivery rate.

The power-law factor δ determines the mobility pattern of nodes. As shown in Fig. 8(c), if δ is small, all nodes tend to have similar wide mobility and, thus, almost identical QDP, which consequently results in ineffective data transmission and low delivery rate. With a large δ, on the other hand, a node stays close to its home cell, i.e., can hardly reach any remote cells. Lower mobility leads to lower network capacity and, thus, a lower delivery rate.

Fig. 8(d) shows the impact of queue size. With an increase in queue size, all schemes enjoy a higher delivery rate because more messages can reside in the queue without being dropped.

We have also studied nodal scanning frequency. A node has a duty cycle. It wakes up to explore possible communication opportunities by scanning nearby nodes. The lower the scanning frequency, the less the meeting events, which leads to lower communication capacity. As a result, a lower delivery probability is observed in Fig. 8(e). Clearly, a higher scanning frequency is at the cost of higher energy consumption.

The FTD threshold β is introduced for redundancy control. In Fig. 8(f), we first observe a higher delivery rate with the increase of β, because a larger β permits more redundancy and accordingly increases data delivery probability. However, when β is greater than 0.7, the delivery rate starts to decrease. Due to given constraints on communication bandwidth and nodal queue size, the excessive redundancy created under high β often does not contribute to improving the delivery rate. Worse yet, it leads to inefficient use of communication opportunities and storage space, resulting in degraded overall performance. It still remains an open problem to find optimal β. As a rule of thumb, the highest delivery rate is achieved when β is around 0.7–0.8 in our simulations and experiments.

VI. Conclusion

In this paper, we have proposed a QoS-aware data delivery scheme for mobile opportunistic networks. It employs QDP to reflect the capability of a node to deliver data to a destination within a given delay budget and maintains a prioritized queue, where the priority is determined by a function of traffic class and dynamic redundancy to support efficient prioritization and redundancy control. Two experiments have been carried out to demonstrate and evaluate our proposed QoS-aware data delivery scheme. The first experiment involves multiple clusters of static Crossbow sensors that are connected by air and ground mobile nodes with controlled mobility. The second experiment is under a mobile social network setting during a period of two weeks, where the prototype is implemented on Dell Streak Android tablets carried by 23 volunteers with arbitrary and diverse mobility patterns. Moreover, simulation results have been obtained under DieselNet trace and power-law mobility model to study scalability and performance trend. Our experiments and simulations have shown that the proposed scheme achieves efficient resource allocation according to the desired delay budget, thus supporting effective QoS provisioning.

References

[1] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking—An approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[2] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary Internet: State-of-the-art and research challenges," *Comput. Netw.*, vol. 43, no. 2, pp. 75–112, Oct. 2003.

[3] Y. Wang and H. Wu, "Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A new paradigm for pervasive information gathering," *IEEE Trans. Mobile Comput.*, vol. 6, no. 9, pp. 1021–1034, Sep. 2007.

[4] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "The BikeNet mobile sensing system for cyclist experience mapping," in *Proc. SenSys*, 2007, pp. 87–101.

[5] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *Proc. IEEE INFOCOM*, 2011, pp. 3119–3127.

[6] B. Zhang, J. Teng, X. Bai, Z. Yang, and D. Xuan, "P3-coupon: A probabilistic system for prompt and privacy-preserving electronic coupon distribution," in *Proc. IEEE Percom*, 2011, pp. 93–101.

[7] Z. Yang and H. Wu, "FINDERS: A featherlight information network with delay-endurable RFID support," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 961–974, Aug. 2011.

[8] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *Proc. SAPIR*, 2004, pp. 239–254.

[9] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proc. SIGCOMM*, 2005, pp. 109–120.

[10] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proc. INFOCOM*, 2005, pp. 398–408.

[11] T. Small and Z. J. Haas, "Resource and performance tradeoffs in delay-tolerant wireless networks," in *Proc. WDTN*, 2005, pp. 260–267.

[12] J. Widmer and J.-Y. L. Boudec, "Network coding for efficient communication in extreme networks," in *Proc. WDTN*, 2005, pp. 284–291.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[14] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proc. SIGCOMM*, 2007, pp. 373–384.

[15] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," in *Proc. IEEE Int. Workshop Sensor Netw. Protocols Appl.*, 2003, pp. 30–41.

[16] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Proc. ACM ASPLOS*, 2002, pp. 96–107.

[17] D. J. Goodman, J. Borras, N. Mandayam, and R. Yates, "INFOSTATIONS: A new system for data and messaging services," in *Proc. IEEE VTC*, 1997, pp. 969–973.

[18] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proc. WDTN*, 2005, pp. 244–251.

[19] J. LeBrun, C.-N. Chuah, and D. Ghosal, "Knowledge based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proc. IEEE VTC-Spring*, 2005, pp. 1–5.

[20] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proc. MoBiHoc*, 2004, pp. 187–198.

[21] P. Klasnja, B. L. Harrison, L. LeGrand, A. LaMarca, J. Froehlich, and S. E. Hudson, "Using wearable sensors and real time inference to understand human recall of routine activities," in *Proc. Ubicomp*, 2008, pp. 154–163.

[22] E. Miluzzo, N. D. Lane, K. Fodor, R. A. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application," in *Proc. SenSys*, 2008, pp. 337–350.

[23] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. SenSys*, 2008, pp. 323–336.

[24] D. Henriksson, T. F. Abdelzaher, and R. K. Ganti, "A caching-based approach to routing in delay-tolerant networks," in *Proc. ICCCN*, 2007, pp. 69–74.

[25] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proc. MoBiHoc*, 2003, pp. 19–20.

[26] U. Shevade, H. Song, L. Qiu, and Y. Zhang, "Incentive-aware routing in DTNs," in *Proc. IEEE ICNP*, 2008, pp. 238–247.

[27] W. Mitchener and A. Vadhat, "Epidemic Routing for Partially Connected Ad Hoc Networks," Duke Univ., Durham, NC, USA, Tech. Rep. CS-2000-06, 2000.

[28] F. Xu, M. Liu, J. Cao, G. Chen, H. Gong, and J. Zhu, “A motion tendency-based adaptive data delivery scheme for delay tolerant mobile sensor networks,” in *Proc. IEEE GLOBECOM*, 2009, pp. 1–6.
[29] I. Leontiadis, P. Costa, and C. Mascolo, “Extending access point connectivity through opportunistic routing in vehicular networks,” in *Proc. IEEE INFOCOM*, 2010, pp. 486–490.
[30] Y. Huang, Y. Gao, K. Nahrstedt, and W. He, “Optimizing file retrieval in delay-tolerant content distribution community,” in *Proc. IEEE ICDCS*, 2010, pp. 308–316.
[31] A. Petz and C. Julien, “An adaptive middleware to support delay tolerant networking,” in *Proc. ARM*, 2008, pp. 17–22.
[32] A. Passarella, M. Kumar, M. Conti, and E. Borgia, “Minimum-delay service provisioning in opportunistic networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 8, pp. 1267–1275, Aug. 2011.
[33] M. Musolesi and C. Mascolo, “CAR: Context-aware adaptive routing for delay-tolerant mobile networks,” *IEEE Trans. Mobile Comput.*, vol. 8, no. 2, pp. 246–260, Feb. 2009.
[34] J. Ott, E. Hyytiä, P. E. Lassila, T. Vaegs, and J. Kangasharju, “Floating content: Information sharing in urban areas,” in *Proc. IEEE PerCom*, 2011, pp. 136–146.
[35] D. Fischer, K. Herrmann, and K. Rothermel, “GeSoMo—A general social mobility model for delay tolerant networks,” in *Proc. IEEE MASS*, 2010, pp. 99–108.
[36] H. Dang and H. Wu, “Clustering and cluster-based routing protocol for delay-tolerant mobile networks,” *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 1874–1881, Jun. 2010.
[37] Y. Wang, H. Wu, F. Lin, and N.-F. Tzeng, “Cross-layer protocol design and optimization for delay/fault-tolerant mobile sensor networks,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 809–819, Jun. 2008.
[38] H. Wu, Y. Wang, H. Dang, and F. Lin, “Analytic, simulation, and empirical evaluation of delay/fault-tolerant mobile sensor networks,” *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, pp. 3287–3296, Sep. 2007.
[39] D. Quercia, G. D. Lorenzo, F. Calabrese, and C. Ratti, “Mobile phones and outdoor advertising: Measurable advertising,” *IEEE Pervasive Comput.*, vol. 10, no. 2, pp. 28–36, Apr.–Jun. 2011.
[40] T. Ning, Z. Yang, X. Xie, and H. Wu, “Incentive-aware data dissemination in delay-tolerant mobile networks,” in *Proc. IEEE SECON*, 2011, pp. 539–547.
[41] L. Cheng, J. Cao, C. Chen, J. Ma, and S. K. Das, “Exploiting geographic opportunistic routing for soft QoS provisioning in wireless sensor networks,” in *Proc. IEEE MASS*, 2010, pp. 292–301.
[42] Q. Du and X. Zhang, “Base-station selections for QoS provisioning over distributed multi-user MIMO links in wireless networks,” in *Proc. IEEE INFOCOM*, 2011, pp. 3038–3046.
[43] Y. Li, P. Wang, D. Niyato, and W. Zhuang, “A dynamic relay selection scheme for mobile users in wireless relay networks,” in *Proc. IEEE INFOCOM*, 2011, pp. 256–260.
[44] M. Geva and A. Herzberg, “QoSoDoS: If you can’t beat them, join them!” in *Proc. IEEE INFOCOM*, 2011, pp. 1278–1286.
[45] R. Southwell, X. Chen, and J. Huang, “QoS satisfaction games for spectrum sharing,” in *Proc. IEEE INFOCOM*, 2013, pp. 570–574.
[46] J. Yallouz and A. Orda, “Tunable QoS-aware network survivability,” in *Proc. IEEE INFOCOM*, 2013, pp. 944–952.
[47] N. Abedini, S. Sampath, R. Bhattacharyya, S. Paul, and S. Shakkottai, “Realtime streaming with guaranteed QoS over wireless D2D networks,” in *Proc. MobiHoc*, 2013, pp. 197–206.
[48] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Performance modeling of epidemic routing,” *Comput. Netw.*, vol. 51, no. 10, pp. 2867–2891, Jul. 2007.
[49] P. Ramanathan and A. Singh, “Delay-differentiated gossiping in delay tolerant networks,” in *Proc. IEEE Int. Conf. Commun.*, 2008, pp. 3291–3295.
[50] H. Liu, A. Srinivasan, Y. Chen, K. Whitehouse, and J. A. Stankovic, “Melange: Supporting heterogeneous QoS requirements in delay tolerant sensor networks,” in *Proc. INSS*, 2010, pp. 93–96.
[51] P. Tsao, S.-Y. C. Wang, and J. L. Gao, “Space QoS framework over a delay/disruption tolerant network,” in *Proc. IEEE Aerosp. Conf.*, 2010, pp. 1–5.
[52] Z. Yang, H. Dang, and H. Wu, “Online estimation of effective delivery capability in delay-tolerant mobile sensor networks,” in *Proc. ICDCS Workshops*, 2008, pp. 228–233.
[53] J. Burgess, B. N. Levine, R. Mahajan, J. Zahorjan, A. Balasubramanian, A. Venkataramani, Y. Zhou, B. Croft, N. Banerjee, M. Corner, and D. Towsley, CRAWDAD data set umass/diesel (v. 2008-09-14) 2008.

Yang Liu (S’11) received the B.E. degree in electrical engineering and its automation and the M.E. degree in control theory and control engineering from Harbin Engineering University, Harbin, China, in 2008 and 2010, respectively. He is currently working toward the Ph.D. degree in computer engineering with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA, USA.

His current research interests include delay-tolerant networks, radio frequency identification systems, and wireless sensor networks.

Zhipeng Yang (S’07) received the B.S. and M.S. degrees in computer science from Tianjin University, Tianjin, China, in 2001 and 2004, respectively. He is currently working toward the Ph.D. degree in computer science with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA, USA.

From 2004 to 2006, he was a Software Engineer with Nortel and Lucent China. His current research interests include delay-tolerant networks, radio frequency identification systems, and wireless sensor networks.

Ting Ning (S’10) received the B.S. and M.S. degrees in computer science from Lanzhou University, Lanzhou, China, in 2005 and 2008, respectively. She is currently working toward the Ph.D. degree in computer science with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA, USA.

Her current research interests include delay-tolerant networks, the application of game theory, and radio frequency identification systems.

Hongyi Wu (M’02) received the B.S. degree in scientific instruments from Zhejiang University, Hangzhou, China, in 1996 and the M.S. degree in electrical engineering and the Ph.D. degree in computer science from the University at Buffalo, The State University of New York at Buffalo, NY, USA, in 2000 and 2002, respectively.

Since then, he has been with the Center for Advanced Computer Studies, University of Louisiana at Lafayette (UL Lafayette), Lafayette, LA, USA, where he is currently a Professor and holds the Alfred and Helen Lamson Endowed Professorship in Computer Science. His research interests include delay-tolerant networks, radio frequency identification systems, wireless sensor networks, and integrated heterogeneous wireless systems.

Dr. Wu received the U.S. National Science Foundation CAREER Award in 2004 and the UL Lafayette Distinguished Professor Award in 2011.