Quota-Control Routing in Delay-Tolerant Networks

Shou-Chih Lo, Chun-Chieh Tsai, Yu-Han Lai
Department of Computer Science & Information Engineering
National Dong Hwa University, Hualien, 974, Taiwan, ROC
sclo@mail.ndhu.edu.tw

Abstract: Delay-tolerant networks (DTNs) are network environments that are subject to delays and disruptions. Traditional end-to-end routing protocols fail in such challenging network conditions because of intermittent connections and/or long delays. Research results have shown that per-hop forwarding of multiple copies of the same message to the destination can produce satisfactory routing performance in DTNs. Current methods rely on the fixed setting of a quota value to limit the number of message copies. This paper proposes a dynamic quota- control mechanism, allowing routing to operate effectively with different traffic loads. To remove useless message copies from the network, a low-cost probability-based method is also presented. The proposed routing framework is then extended to interest-based information dissemination, which is used to efficiently disseminate an event message to all interested users. A performance evaluation was conducted using a real social contact trace, and performance comparisons with other DTN routing protocols are provided.

Index Terms: Delay-Tolerant Network (DTN), Quota Control, Buffer Management, Information Dissemination

1. Introduction

The delay-tolerant network (DTN) was originally developed for interplanetary communications that are subject to delays and disruptions. The characteristic properties of DTNs include a long or variable delay, low data rates, intermittent connectivity, and high error rates [1]. These DTN properties violate assumptions required for current Internet communication, making the TCP/IP protocol unsuitable for data transmission in these challenged networks. A novel message switching method, other than the existing packet switching method, is required. The Delay-Tolerant Networking Research Group (DTNRG) combines research efforts to address DTN related design issues [2].

The DTN architecture [3] has been proposed by adding a new bundle layer [4] between the application and transport layers. The bundle layer provides store-and-forward messaging switching. Application data are conveyed using the protocol data unit, called a bundle, of the bundle protocol. A bundle is also named a message throughout the paper. A DTN node (or router) may have persistent storage used to store messages in its own buffer, and forwards these messages to other contact nodes. A DTN node that is willing to participate the store-and-forward switching is called a custodian. Data transmission between two end nodes may involve message forwarding by a sequence of custodians along a routing path, and this routing behavior is called custody transfer. Custodian-based node-to-node data retransmission will replace TCP-based end-to-end data retransmission [5].

Studies have examined many DTN application domains [6]. For example, the Interplanetary Internet project [7] studies outer-space communications. The ZebraNet project [8] uses sensors to study interactions between zebras. The DakNet project [9] enables communication in remote regions in India. The Haggle project [10] uses hand-held devices to enable people to exchange information. The CONDOR project [11], supported by the U.S. Marine Corps, studies communication on battlefields. The UMassDieselNet project [12] examines data transmissions between buses on the University of Massachusetts campus. These projects all use wireless transmission media that inherently have DTN properties.

Routing data from a source to a destination is a crucial task for all data networking systems. Most routing strategies involve discovering and maintaining routing paths in a network. For example, Internet routers exchange routing information (a distance vector or link state) with each other to maintain their data forwarding tables. A mobile node in a multi-hop wireless ad hoc network may flood a route request message into the whole network to discover a route path. These types of strategies are difficult or incur high costs in a DTN, because a DTN is prone to dynamically change with time, and intermittent connections make a stable end-to-end route path difficult to maintain.

Most DTN routing techniques use per-hop message forwarding; therefore, forward decisions are made on a per-hop and per-message basis. Two advantages to this are that routine maintenance is unnecessary and frequent routing decisions help the routing path to adapt to dynamic networks. To increase message delivery rates, several copies of the same message can be generated and individually forwarded toward the destination. However, these redundant message copies waste network resources, such as buffer space and transmission power. Quota-based DTN routing [13-15] solves this problem because it limits the number of message copies by associating each newly generated message with a quota. A message associated with a quota that has been reached can no longer be duplicated. Setting a proper initial quota for a message is difficult without knowing the conditions throughout the network.

In this paper, a mechanism is proposed to dynamically control the message quota (increasing or decreasing quota value) according to changing network congestion conditions. This mechanism reduces the sensitivity of the initial quota setting to the routing performance. The paper makes the following contributions: a new cost metric, called contact density, is introduced to evaluate the benefit of duplicating a message to a node; a dynamic quota-control mechanism is proposed based on observing local network congestion conditions; a low-cost method, which can be performed locally by each node based on a probability function, for removing redundant message copies is proposed; the unicast routing framework is applied to interest-based information dissemination, which can be used to efficiently distribute an event message to all interested users. This kind of information dissemination routing can be viewed as a DTN multicast protocol where those users having the interest to the same message form a multicast group.

The remainder of this paper is organized as follows: Section 2 provides a brief survey of DTN routing protocols; Section 3 describes the proposed routing approach; Section 4 presents a comparison of routing method performances; and Section 5 presents concluding remarks.

2. Related Work

Many studies have discussed DTN routing techniques. Existing strategies can be roughly classified into three categories according to the number of copies of the same message that are created: forwarding, quota-replication, and flooding strategies. In a forwarding scheme, a single-copy message is forwarded from the source through successive intermediate nodes to the destination. A quota-replication scheme involves creating a limited number of copies of a message as specified by a quota. The same number of nodes that hold these message copies exists in the network. Each of them is called a message holder. The destination, which is not a message holder, receives the message when contacting one of these message holders. A flooding scheme yields an extremely high number of message copies; therefore, all nodes in the network become message holders. A detailed comparison of these three routing schemes is provided in [16].

Flooding routing methods, such as Epidemic [17], MaxProp [18], and Prophet [19], perform well only when the buffer space in each node is sufficient. When the buffer space is limited, storing many irrelevant message copies causes buffer overflow. Epidemic replicates all messages in the buffer, which are not redundant, to every contact node. MaxProp uses the same routing method as Epidemic, but improves buffer space usage. Prophet acts as a gradient routing method and replicates a message to a contact node that has a higher delivery probability to the destination than the current node.

Forwarding routing methods, such as MEED [20] and SimBet [21], mainly save buffer space in each node, but they exhibit low delivery rates because a single route path is easily broken in the network. MEED involves a global exchange of link costs among nodes to derive the shortest path route. SimBet is also a gradient routing scheme, but forwards a message to a contact node with a high utility value, evaluated based on local topology and friend relationship information.

Quota-replication routing methods, such as Spray and Wait [13], EBR [14], and SARP [15], perform more effectively than the other two types of routing methods because of restricted message replication. The maximal number of copies of a message is fixed by specifying a quota that cannot be dynamically changed.

In quota-replication routing, a node can conditionally duplicate a message in the buffer and forward the duplicated message to another encountered node. The duplicated message is allocated part of the quota of the original message based on a quota allocation function. Therefore, the message quota gradually decreases after successive message duplication. A message with an associated quota of one can no longer be duplicated. A node holding a message with a quota of one waits for direct contact with the destination.

The design of the quota allocation function determines how message copies are spread throughout the network. Spray and Wait considers a binary allocation function, allowing a duplicated message to receive half of the quota from the original message. EBR maintains an encounter value for each node that equals the average number of encounters with other nodes during an observation period. The allocation function is based on the ratio of the encounter values of two contact nodes. SARP behaves similarly to EBR, but uses the encounter value with the message destination. SARP also uses a novel method to count the number of encounters between two nodes. Short and long contacts respectively contribute zero and more than one to the encounter time.

Selecting the message holders in quota-replication routing is critical. Spray and Wait randomly selects these holders; EBR selects holders that contact many other nodes; and SARP selects holders that frequently contact the destination. Evaluating the suitability of a node to be a message holder is important. The contact history or behavior between two nodes provides a valuable reference for this evaluation. Useful reference data include the average contact duration, average contact waiting time, and contact frequency [16]. In this paper, the concept of contact density is introduced to well evaluate the usefulness of a node to be a message holder. Store-and-forward switching requires buffer space at a node. Routing approaches that use extreme message replication quickly exhaust this buffer space. Dropping minor messages in a buffer is necessary when the buffer overflows. Moreover, two nodes may have insufficient time to exchange all the messages in their buffers during the contact time. Transmitting major messages first is essential. Determining the relative importance of each message in a buffer is a technical issue related to buffer management. Various information items such as the received time, remaining lifetime, and travelling hop count of a message, can be considered [16][22]. For example, a typical method is to first transmit and drop the message with the earliest received time in the buffer by following the basic first-in-first-out (FIFO) principle. MaxProp uses two items of information, such that the message with the lowest hop count is transmitted first and the message with the highest path delivery cost is dropped first.

The buffer exhaustion problem is serious at certain nodes when they experience traffic congestion. A congested node can transfer some messages in its buffer to surrounding nodes with available buffer space [23][24]. This method works but incurs extra message moving cost. In quota-replication routing, the message quota is dynamically reduced with a fixed fraction to alleviate the congestion problem [25]. The time to trigger the congestion control depends on the congestion level, which may be computed by using the ratio of the total number of dropped messages to the total number of message copies observed by a node [25]. [26] showed that using the buffer occupancy as the congestion level index is more favorable than using this ratio value. The same congestion index as [26] is used in [27], but the message quota is reduced by a parameterized function. This paper is extended from this previous work but proposes more improvements and ideas on the routing design.

In flooding or quota-replication routing, when a message copy reaches a destination, other message copies in the network are considered garbage. A common technique used to clean these garbage messages is to exchange the i-list data [28]. The i-list records the identifiers of messages that are known to have reached their destinations. When a destination successfully receives a message, the node adds a new record for the message into its i-list. Two contact nodes exchange and merge their i-list records. To prevent the formation of a long i-list, each record in the i-list is associated with a remaining time, allowing an expired record to be deleted. The remaining time is difficult to set in a real network environment. In contrast to the i-list, this paper proposes a probability-based method to remove garbage messages without exchanging any extra information.

The above routing techniques aim at DTN unicast applications. DTN multicast routing can be designed using the unicast-based, broadcast-based, or tree-based scheme [29]. In a unicast-based scheme, a sender sends each copy of a message to each receiver in the same group using the underlying unicast routing. In a broadcast-based scheme, a message is flooded throughout the whole network from a sender. In a tree-based scheme, a multicast tree is constructed from a sender to all receivers of the same group. The tree-based scheme is more cost-saving than others, but the maintenance of the tree structure is tough in DTNs. OS-Multicast [30], EBMR [31], and Delegation-Multicast [32] are all tree-based schemes. This paper considers another multicast application according to user interests. A user can claim its multiple interests on certain event messages such as sport and activity news. An information dissemination protocol is needed to send a message to all interested users. This kind of multicast group is more flexible, and a different multicast protocol against the traditional ones is required.

3. Quota-Control Routing

The motivation for using quota-replication routing and the disadvantage of using the i-list mechanism are first discussed based on a cost analysis. A novel unicast routing protocol is then presented. The core routing technique is then extended to disseminate event messages to all relevant nodes that users are interested in.

3.1. Routing Basis

A DTN can be modeled as a dynamic weighted graph, $G = (V, E, w_t(e))$, where V denotes the set of nodes and E denotes the set of links between nodes. $w_t(e)$ denotes the delivery cost of a message over a link e at time t. Each link is assumed to be symmetric. The delivery cost contains the following four components [33]:

- Connection waiting time: time spent waiting for the link to be connected. This value depends on contact behavior.
- Buffer waiting time: time spent waiting for the message to be served in the buffer. This value depends on the network traffic load.
- Transmission time: time required to deliver the message from the buffer to the link. Transmission bandwidth determines this value.
- Propagation time: time required for the message to traverse the link. The propagation speed determines this value.

To facilitate a discussion of routing behavior, a static weighted graph is considered by replacing $w_t(e)$ with $w(e)$ which denotes the average delivery cost of link e over time. A flooding routing scheme, such as Epidemic, spreads a message to all nodes in V. In Epidemic, to confirm that each node receives a single message copy, a meta-data list, called an m-list, is exchanged between two nodes before the message is transmitted. The m-list summarizes the content of one buffer by listing message identifiers within the buffer. By comparing two m-list records, a node can avoid transmitting a message to another node that has already held this message in the buffer.

The delivery paths of such a flooding routing scheme can be represented by the shortest-path tree (SPT) rooted at the message source node in G, as shown in Fig. 1a. The following cost metrics are used to evaluate the routing performance of a message from the source to the destination:

- Delivery probability: the probability that the destination successfully receives the message.
- Relay count: the number of message relays with values that increase by one as

any message copy is forwarded from one node to another node in the network.

- Relay cost: the ratio of the relay count to delivery probability.
- Latency: the end-to-end delay of a message from the source to the destination.

Assume that no message is dropped during the transferring and buffering time. When using flooding routing, the delivery probability is always one (except the case of buffer overflows, elaborated later) and the relay count is $|V| - 1$ (equal to the number of edges in the SPT). Therefore, the relay cost equals $|V| - 1$. The latency is the total delivery time along the path from the root to the destination in the SPT. The highest latency value is determined by the diameter (the longest of the shortest paths in a network) of G.

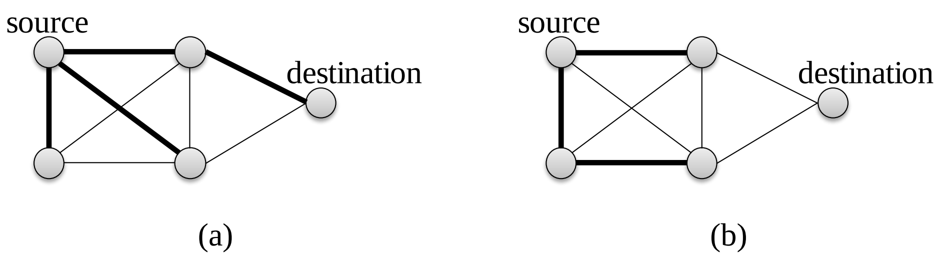


Fig. 1. Delivery path tree: (a) in flooding, and (b) in quota-replication.

When using quota-replication routing, the message is partially flooded into the network, and the delivery paths form a tree (not necessary an SPT) in G, as shown in Fig. 1b. F represents the set of nodes in the tree (including the root). If the destination belongs to F, then the destination receives the message. Therefore, the delivery probability of direct message replication is $(|F| - 1)/(|V| - 1)$. However, the destination can also indirectly receive the message by contacting a node in F. Let the probability that the destination has connection links to any nodes in F be p. The value of p is proportional to the size of F and E. Therefore, the total delivery probability is $(|F| - 1)/(|V| - 1) + p$. This delivery probability is always one if all nodes in the dominating set of G are message holders. The relay count is $|F| - 1$ and the upper bound of the relay cost is $|V| - 1$ when p is zero. The latency is either infinite or has a value greater than or equal to the flooding routing case.

This discussion shows that flooding routing provides a high delivery probability, a short latency, and a relay cost equal to the highest quota-replication routing case. However, without removing messages in the buffer, the buffer may eventually overflow, causing some messages in the buffer to be dropped. Flooding routing has a $|V|/|F|$ times higher risk of buffer overflow than quota-replication routing does. Dropping a message in the buffer of a node causes another message copy to the node in flooding routing. Therefore, the occurrence of more message dropping events causes more message relays, increasing the relay cost and the latency. The latency may even become infinitely long if message copies are always dropped before reaching their destinations; hence, the delivery probability is not always one. In quota-replication routing, no more message copies can be generated when the quota is reached. Therefore, a message dropping event may reduce the delivery probability, but does not increase the relay count. Thus, when buffer space is limited, flooding routing loses its advantage and quota-replication routing is preferable.

The i-list mechanism can remove garbage message copies, but it incurs high overheads. Because it is difficult to identify all nodes in the network that have been notified on the i-list, every encounter event between two nodes must involve the exchange of two i-list records. The total exchanging cost is exponential to $|E|$ in G, which is considerable. The longest time required to remove a garbage message is determined by the diameter of G. In this paper, a novel mechanism used to quickly remove possible garbage messages without exchanging information is proposed.

3.2 Routing Design

This paper proposes a quota-control routing (QCR) protocol with its operation shown in the diagram of Fig. 2. A source sends a message encapsulated with an initial message quota and other bundled information fields. A message with a quota greater than 1 is called a duplicable message. This means that this message can be duplicated to another node. When a node holding any duplicable messages encounters another node, these two nodes first exchange meta-data. A sorting policy is then applied to determine the transmission order of messages in the buffer. A *quota adjustment function* is performed to increase or reduce the quota of each transmitted message according to the observed network state. A *quota allocation function* is then used to allocate quota values between the message and its duplicate. The duplicate is subsequently forwarded to the contact node. Finally, the node performs buffer cleaning and deletes the duplicate message in the buffer if it is confirmed to be useless.

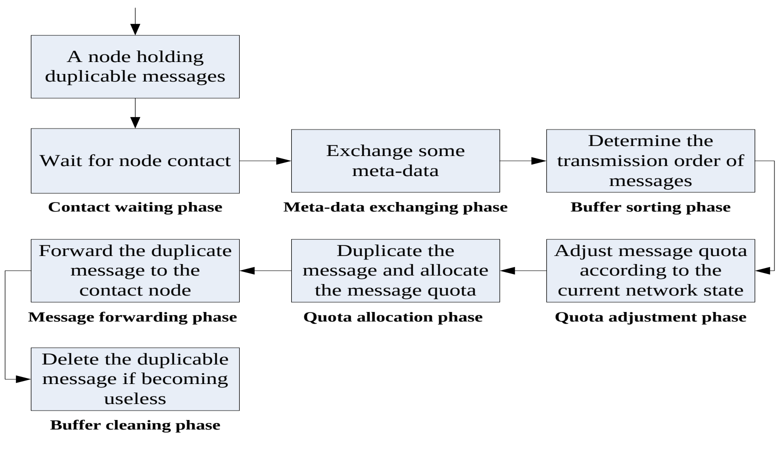


Fig. 2. Basic routing flow of QCR.

In QCR, each node maintains a neighbor table that records information related to each recent contact node within a sliding time window. This sliding window, with size W, is used to capture recent contact events with the node. The neighbor table contains the following data fields:

- node identifier: node identification number.
- buffer occupancy: the ratio of the buffer that is occupied in a node
- m-list: a list of message identifiers in the buffer of a node
- neighbor list: a list of node identifiers of the neighboring nodes surrounding a node.
- contact history: a list of time stamps showing the connection start and end times for a node.

A. Dynamic Quota Adjustment

Quota setting directly influences the number of message copies in a network. Values that are too low or too high reduce the delivery probability and increase the risk of buffer overflow, respectively. The network size and real traffic loads should be considered when determining a suitable setting. However, these reference data are difficult to acquire in a distributed environment without global information exchange among nodes.

Here, a local network congestion condition is considered by observing the buffer occupancies of neighboring nodes to adjust the message quota. A buffer occupancy between 0 (empty buffer) and 1 (full buffer) reflects the ratio of a buffer that is occupied in a node. High buffer occupancy indicates the condition of a congested network, where there are too many messages generated or replicated in the network. This case implies that reducing the message quota relieves the high buffer occupancy. Low buffer occupancy reflects a light traffic load; hence, the message quota can be increased to raise the delivery probability.

When a node encounters another node, they exchange and store their current buffer occupancies. The total average buffer occupancy (ABO) of all nodes in the neighbor table is used to indicate the local network conditions surrounding a node. Suppose that ABO_i is a local value calculated by node i, and $Q_{m,i}$ is the quota value associated with message m held by node i. A threshold-based method, shown in (1), is applied to adjust the message quota. When ABO_i is less than or equal to 0.2, the network is assumed to be lightly loaded; hence, the message quota is increased by one. When ABO_i is greater than or equal to 0.8, the network is assumed to be heavily loaded; hence, the message quota exponentially decreases. In other cases, the message quota remains unchanged.

$$Q_{m,i} = \begin{cases} Q_{m,i} + 1, if \;\; ABO_i \leq 0.2 \\ \lceil Q_{m,i} \times e^{-CM_{m,i}/CN_i} \rceil, if \;\; ABO_i \geq 0.8 \end{cases} \tag{1}$$

When reducing the message quota, the number of neighboring nodes and the messages held by these neighboring nodes are also considered. Let CN_i denote the number of nodes in the neighbor table of node *i*. The message quota is greatly decreased in (1) when CN_i is small, because this node has fewer contact nodes that help to spread the message. The term $CM_{m,i}$ represents the number of nodes in the node *i* neighbor table with message *m* in their buffers. This value can be calculated by referring to the m-list field in the neighbor table. The message quota is greatly decreased in (1) as well when $CM_{m,i}$ is large, because most surrounding nodes have already held the message.

B. Quota Allocation

A duplicated message obtains a quota value from the duplicated source. This is performed using a quota allocation function. If this duplicated message is forwarded to a more active node that can encounter more nodes than the current one can, then a high quota is allocated. A new metric called contact density (CD) is proposed to measure the activeness of a node in the network. The term CD_i is defined as the contact density of node *i* and is calculated using (2).

$$CD_i = \sum 1/WT_{i,j} \text{ , } \forall j \text{ that is a neighboring node of } i \tag{2}$$

The term $WT_{i,j}$ is the average contact waiting time between nodes *i* and *j*, as introduced in [20]. This value indicates the average time that node *i* waits for the next contact event with node *j*. Consider the fraction of contact history between nodes *i* and *j* in Fig. 3. When the first contact ends, the contact waiting time is *D* (the duration of one inter-contact period), and the contact waiting time then linearly decreases and becomes zero when the next contact starts. At any time point from when contact starts to when the second contact ends, all contact waiting times are zero. Therefore, the average contact waiting time between these two successive contact ends is calculated using (3), where *C* is the duration of one contact period.

$$WT_{i,j} = D^2/(2(D+C)) \tag{3}$$

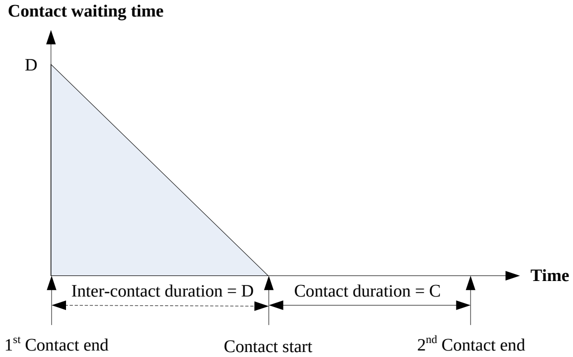


Fig. 3. Computation of average contact waiting time.

The average contact duration and the average inter-contact duration between two nodes are computed by referring to the contact history filed in the neighbor table. The reciprocal of $WT_{i,j}$ reflects the contact frequency between nodes i and j. The contact frequencies of the contact nodes in the node i neighbor table are summed to measure the activeness of node i.

The basic method used to allocate the message quota is based on the ratio of the contact densities of two contact nodes. A node having a higher contact density receives a higher quota. Actually, it is more expected that a node can contact more new neighboring nodes. The neighboring nodes of two contact nodes should be compared too. The term N_i is the set of nodes in the node i neighbor table, and d_{ij} is the number of elements in the relative complement of N_j in N_i, as shown in (4). This value is computed by referring to the neighbor list field in the neighbor table. If node i has message m with quota $Q_{m,i}$ and would like to forward a message copy with quota $Q_{m,j}$ to node j, then the quota allocation is calculated using (5). The contact density is weighted according to the difference between neighboring sets. If node i can encounter all neighboring nodes of node j (i.e., d_{ji} = 0), then this node does not forward a message copy to node j.

$$d_{ij} = |N_i - N_j| = |\{x \in N_i | x \notin N_j\}| \tag{4}$$

$$Q_{m,j} = \left\lfloor Q_{m,i} \times \frac{CD_j \times d_{ji}}{CD_i \times d_{ij} + CD_j \times d_{ji}} \right\rfloor \tag{5}$$

As shown in Fig. 4, the source, Node S, holds a message with a quota of 12. After contacting Node A, Node S duplicates a message with a quota of 2 to Node A and leaves a quota of 10 to its held message. After contacting Node C, Node S duplicates a message with a quota of 2 to Node C. Finally, Node C forwards the message to the destination, Node E.

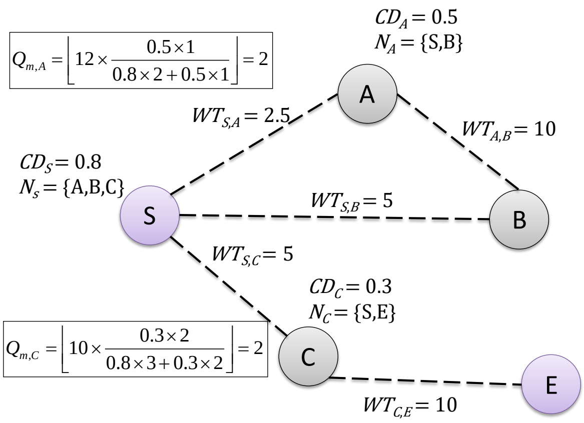


Fig. 4. Example of quota allocation using CD.

C. Buffer Management

A suitable buffer management system is required to determine the transmission order of buffered messages when they encounter opportunities to be forwarded or copied to other nodes. The message drop order should be determined when the buffer space is full. Although many methods can be used to determine these orders, an efficient method is proposed here. Each message in the buffer is associated with two items of information: the received time (the time when a buffer is entered) and hop count (the number of hops taken from the message source). The transmission order of messages is determined according to the product value of these two items of information. Low transmission priority is given to a message with a high product value, because this message has travelled the network for a long distance and has recently entered the buffer. Those messages whose hop counts are zero have high transmission priorities and the order is determined by the ascending order of received time (i.e., FIFO policy). The drop order is determined first by the hop count and the message with the highest hop count is dropped first. The reason is that this message has the high probability of owning many message copies in the network and removing one of them has no great impact on message delivery. If the hop count is zero (i.e., the corresponding message remains in the source node), the message with the shortest received time (i.e., an old message in the buffer) is dropped then.

This study proposes a cost-free mechanism, other than the i-list mechanism, to clear garbage messages in the buffer when their destinations receive these messages from other message copies. A message held by a node becomes useless if the recently contacted nodes have already held this message. Based on this assumption, a probability-based method is introduced to remove a message from the buffer. The probability of removing message *m* in the buffer of node *i* is calculated using the quadratic function in (6). This probability is increased because $CM_{m,i}$ is high and equals 1 because $CM_{m,i} = CN_i$. Compared with the i-list, no extra messages are exchanged and the latency required to remove a message is short. For example, the message is definitely deleted after a node has contacted all neighboring nodes. This method is more suited to conditions where each node has a stable set of neighboring nodes, which are used as social networking cases.

$$P_{m,i} = \left(\frac{CM_{m,i}}{CN_i}\right)^2 \tag{6}$$

D. Routing Algorithm

The pseudocodes of QCR routing are given and the entire routing process is triggered when two nodes encounter each other. First, meta-data are exchanged between them and some local data items are updated. The messages in the buffer are then sorted in transmission order and are individually examined. A redundant message and a message destined for the contact node are skipped and forwarded, respectively. Otherwise, the message quota is adjusted and allocated accordingly and a duplicated message is forwarded to the contact node. The probability of removing the message is then calculated. If a randomly generated number between 0 and 1 is less than this probability value, then the message is removed from the buffer.

Pseudocodes of QCR:

Triggered when node i contacts with node j

Begin

1. Store the exchanged meta-data (buffer occupancy, contact density, m-list, and neighbor-list).
2. Compute the new values of CD_i, CN_i, $CM_{m,i}$, and ABO_i.
3. Order the messages in the buffer according to the sorting policy.
4. **For** each message m with quota $Q_{m,i}$ in the buffer,
5.   **If** message m is in the m-list of node j,
6.   Skip this message.
7.   **Else if** the destination of message m is node j,
8.   Forward message m from node i to node j.
9.   **Else,**
10.   Update $Q_{m,i} = \begin{cases} Q_{m,i}+1, if \ \ ABO_i \le 0.2 \\ \lceil Q_{m,i} \times e^{-CM_{m,i}/CN_i} \rceil, if \ \ ABO_i \ge 0.8 \end{cases}$.
11.   **If** $Q_{m,i} > 1$,
12.   Compute $Q_{m,j} = \left\lfloor Q_{m,i} \times \frac{CD_j \times d_{ji}}{CD_i \times d_{ij} + CD_j \times d_{ji}} \right\rfloor$.
13.   **If** $Q_{m,j} > 0$,
14.   Copy message m with quota $Q_{m,j}$ to node j and update $Q_{m,i} = Q_{m,i} - Q_{m,j}$.
15.   Increase $CM_{m,i}$ by one.
16.   Drop message m with probability $P_{m,i} = (\frac{CM_{m,i}}{CN_i})^2$.

End.

3.3 Interest-based Data Dissemination

QCR is easily extended from unicast to multicast. Multicast involves several destinations, and these destinations can retrieve one message copy from different contact nodes that are message holders. The only modification is to remove the statements of Numbers 7 and 8 in the pseudocodes of QCR. In other words, a duplicable message is continuously copied to a contact node regardless of whether it is a destination. This means that the source node and other intermediate nodes do not need to know where the destinations are in advance. Therefore, a node can freely join and leave a multicast group without sending notification messages to the source node.

The flexibility of the proposed routing scheme allows an interest-based data dissemination framework to be developed. The problem is modeled as follows. Each node is associated with an interest vector (IV) specified by the corresponding user. An IV is represented by a multi-dimensional vector $\{w_1, w_2, w_3, \ldots, w_k\}$, where each w_i equals one or zero. Each dimension corresponds to a separate keyword (e.g., movie, sports, etc.) that describes an interest or interest group. w_i is set to one if the user has the corresponding interest. A user can have multiple interests or join multiple interest groups at the same time. Any node can send an event message associated with an IV that describes the interests belonging to the message. Users with interests that match one of the interests in this IV are interested in this event message. The goal is to efficiently route an event message to all interested users.

The multicast version of QCR provides a straightforward solution to this problem and it is modified to further improve its performance. QCR is prone to allocate a high message quota to an active node that easily contacts all other nodes. Here, the rule is changed such that a high quota is allocated to an active node that easily contacts nodes interested in the event message. The contact density is first changed to a vector form (called the CDV), shown in (7), that separates the contact densities of individual interest groups. IV_j is the interest vector of node j. The quota allocation function is then changed to (8), where IV_m is the interest vector specified in message m. The inner product of $CDV_i \cdot IV_m$ (or $CDV_j \cdot IV_m$) shows the accumulated contact density of node i (or j) to the interest groups specified in IV_m. A node with a high accumulated contact density satisfies the new definition of an active node.

According to this quota allocation function, contact node j receives a zero quota if $CDV_j \cdot IV_m = 0$. In other words, a message is not replicated to a node that does not encounter nodes that are interested in message m. During the early message spreading stage, all interested nodes may be far from the source node and its neighboring nodes. Hence, the event message remains stuck at the source. To solve this problem and increase the chance of spreading the event message in certain special cases, the accumulated contact density of all interest groups in the network is considered. That is, IV_m is changed to a constant vector **1**, of which all values are ones. (9) shows the condition in which IV_m should be modified. This modification is performed within the quota allocation function and does not affect the IV_m value associated with the event message.

$$CDV_i = \sum \frac{1}{WT_{i,j}} \times IV_j \text{ , } \forall j \text{ that is a neighboring node of } i \tag{7}$$

$$Q_{m,j} = \left\lfloor Q_{m,i} \times \frac{(CDV_j \cdot IV_m) \times d_{ji}}{(CDV_i \cdot IV_m) \times d_{ij} + (CDV_j \cdot IV_m) \times d_{ji}} \right\rfloor \tag{8}$$

$$IV_m = \begin{cases} \mathbf{1}, if \;\; CDV_j \cdot IV_m = 0 \\ \;\; IV_m, \text{otherwise} \end{cases} \tag{9}$$

As shown in Fig. 5, the source, Node S, holds a message with $IV_m = \{1,0,0,0\}$ and a quota of 12. After contacting Node A, Node S finds that Node A has no neighboring nodes that are interested in this message (i.e., $CDV_A \cdot IV_m = 0$). All values in CDV_A and CDV_S are separately summed using the quota allocation function. Therefore, Node S duplicates a message with a quota of 2 to Node A. If all neighboring nodes have only one interest, then $CDV_j \cdot \mathbf{1} = CD_j$. After contacting Node C, Node S finds that Node C has one neighboring node that is interested in this message (i.e., $CDV_C \cdot IV_m \neq 0$). Only the first elements in CDV_C and CDV_S are included in the quota allocation function. Finally, Node S duplicates a message with a quota of 10 to

Node C.

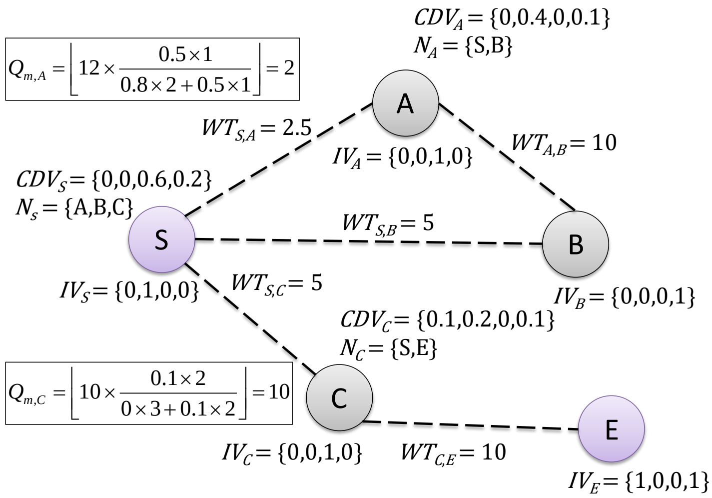


Fig. 5. Example of quota allocation using CDV.

4. Performance Evaluation

The performance evaluation was conducted using the Opportunistic Network Environment (ONE) simulator [34]. ONE is Java-based open-source software that has provided the implementation codes for certain DTN routing protocols. Three cost metrics were used in the evaluation:

- Delivery ratio: the ratio of all the messages received by destinations to all the messages intended to be received by destinations.
- Average relay cost: the ratio of the total message relay count to all the messages received by destinations.
- Effective latency: the ratio of the average end-to-end delay for a message from the source to the destination to the delivery ratio.

Effective latency is different from a conventional end-to-end delay, and exhibits a compound effect to reveal a protocol with a short end-to-end delay but a low delivery ratio.

A. Simulation Model

Two real trace files, which recorded the contact events of attendants during the 2005 and 2006 INFOCOM conferences respectively, were downloaded from the CRAWDAD webpage [35]. To capture purely social contact, all non-person-to-person contact events were removed. In total, 41 nodes in INFOCOM05 and 50 nodes in INFOCOM06 remained. These contact events spanned time intervals of 276000 s and 340000 s, respectively. The constructed contact graph shows a high node degree for these trace files. The contact and inter-contact duration distributions are highly skewed and follow the power law with a heavy tail. This means that people frequently contacted other people for a considerably short time (e.g., walked past) and long contact relationships seldom occurred.

Table 1 lists the experimental parameter settings. Fixed-size messages are constantly generated per time interval during a generation period. In unicast routing, the source and destination of a generated message are two distinct nodes that are randomly selected from all nodes. In interest-based routing, each node selects from ten interests and the number of selected interests is followed by a normal distribution (mean = 5 and standard deviation = 1). Interest setting also follows the power law that 80% of nodes select from 20% of interests. The source node of a generated event message is randomly selected from all nodes, and the destination group is selected from the ten interest groups. In other words, the event message only specifies one interest. Nodes with interests that match the specified interest belong to the destination group and are the destinations of the generated event message. Table 2 shows a comparison and provides descriptions of five DTN routing protocols. In the experiments, the average value of five tests is computed.

Table 1. Parameter Settings.

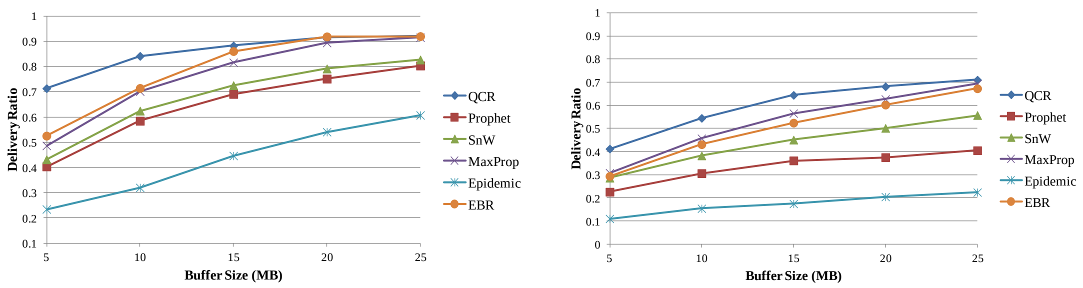
Parameter	Value
Buffer size	5 MB, 10 MB (default), 15 MB, 20 MB, 25MB
Transmission rate	250 kBps
Message size	500 kB
Message interval	100 s (INFOCOM05) 50 s (INFOCOM06)
Message generation period	20000 s to 130000 s (INFOCOM05) 10000 s to 200000 s (INFOCOM06)
Message TTL	infinite
Initial quota	12 (INFOCOM05) 30 (INFOCOM06)
Time window	3000 s
Number of interest groups	10
Destination group size	2, 4, 8, 16 (default), 32

Table 2. Evaluated Routing Protocols.

Protocol	Routing Strategy	Garbage Message Cleaning	Message Sending Policy	Message Dropping Policy
Epidemic	Flooding	No	Old message	Old message
MaxProp	Flooding	i-list	Message with small hop count	Message with high path delivery cost
Prophet	Flooding	No	Old message	Old message
Spray&Wait (SnW)	Quota-Replication	No	Old message	Old message
EBR	Quota-Replication	i-list	Old message	Old message
QCR	Quota-Replication	Probability-based	Message with old age and small hop count	Message with large hop count

B. Simulation Results

In the first set of experiments, performance comparisons of various unicast routing protocols were performed by varying the buffer size. QCR produced the highest delivery ratio of all the routing protocols, particularly when the buffer size is small (Fig. 6). This is because that QCR relieves buffer overflows by reducing message quotas and does forward the limited message copies to those valuable nodes. When the buffer space is sufficient (greater than 15 MB for INFOCOM05 and 25 MB for INFOCOM06), EBR and MaxProp perform near to QCR. QCR provides no more significant improvement because of the nature limit of contact behaviors in these trace files. This indicates that QCR is more resource efficient. The other protocols performed poorly when the buffer space was limited, because the excessive number of message copies caused numerous messages to be dropped. If the buffer size is extremely large, there is no doubt that flooding routing which acts like a brute-force way to find where the destination is in the network will become excellent. In this case, QCR should increase the message quota exponentially. However, we show the resource efficiency of the proposed approach even with small buffer space.

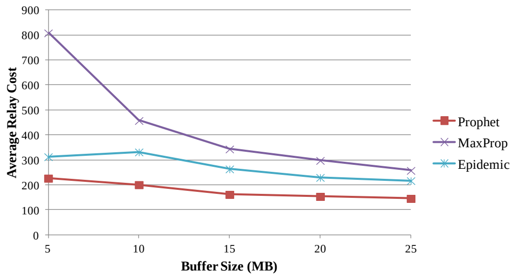


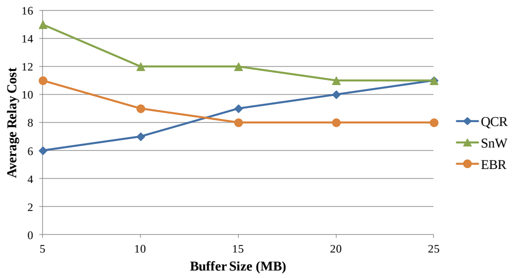
(a) INFOCOM05 (b) INFOCOM06

Fig. 6. Unicast routing comparison: delivery ratio vs. buffer size.

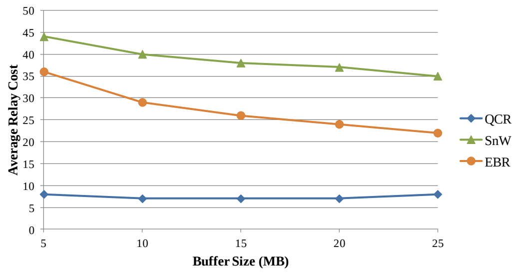
Protocols without garbage message cleaning functions, such as SnW, Prophet, and Epidemic, exhibited low delivery ratios. MaxProp and EBR use the i-list mechanism, and could compete with QCR when the buffer size was sufficiently large. This indicates that cleaning garbage message copies is worthwhile. After an inner test using the default parameter settings, the i-list mechanism incurred a high data exchange (more than 20 MB in the network). The proposed probability-based method was used to prevent this problem.

Flooding protocols (Epidemic, MaxProp, and Prophet) exhibited higher relay costs than did the quota-replication protocols (QCR, EBR, and SnW), because more message copies were generated during the message routing stage (Fig. 7). Of the flooding protocols, MaxProp produced the highest relay cost, because the i-list mechanism saved substantial amounts of buffer space, allowing a message to reach its destination after many relays. QCR has the lowest relay cost when the buffer space is less than 15 MB in INFOCOM05, where the buffer space is not sufficient. The relay cost is increasing when the buffer space becomes large, since the message quota is increased. QCR produced a moderately effective latency performance (Fig. 8). QCR experienced long end-to-end delays, because messages were not easily dropped before reaching their destinations. Indeed, there is a little bit tradeoff between the increasing of delivery ratio and the deceasing of latency. However, a long delay is expected in DTNs and a high delivery ratio is the major concern.



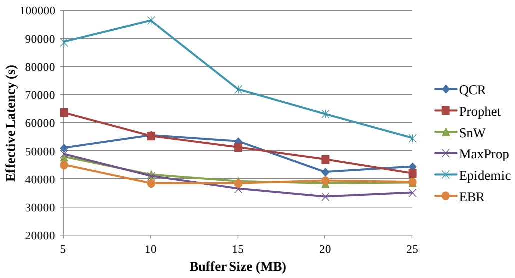


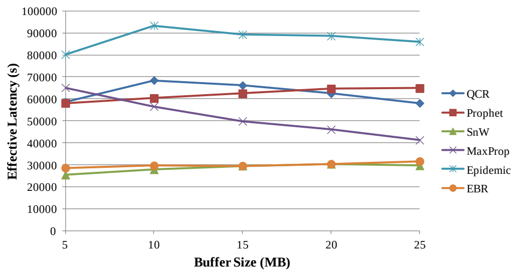
(a) INFOCOM05 (b) INFOCOM05



(c) INFOCOM06

Fig. 7. Unicast routing comparison: relay cost vs. buffer size.

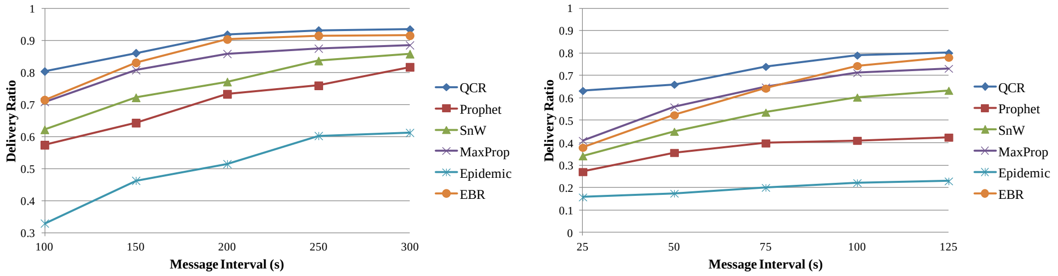




(a) INFOCOM05 (b) INFOCOM06

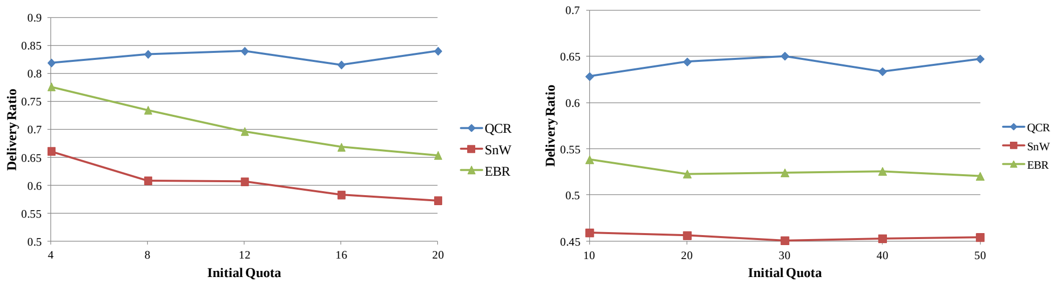
Fig. 8. Unicast routing comparison: effective latency vs. buffer size.

In the second set of experiments, performances were compared by varying the message generation time intervals. The delivery ratio was re-examined to verify the favorable performance of QCR (Fig. 9). This emphasizes the ability of the proposed routing scheme to adapt to different network traffic loads. Because quota-replication protocols rely on setting a suitable initial message quota, their performances may be sensitive to this quota value. However, QCR is less sensitive than the other two protocols particularly in INFOCOM05 (Fig. 10).



(a) INFOCOM05 (b) INFOCOM06

Fig. 9. Unicast routing comparison: delivery ratio vs. message interval.



(a) INFOCOM05 (b) INFOCOM06

Fig. 10. Sensitivity test of initial quota values.

Some internal tests are performed to verify the setting of QCR. For example, the pair of threshold values (0.2, 0.8) in quota adjustment is compared with other pairs such as (0.3, 0.7) and (0.4, 0.6). The result shows that the original setting has little improvement. Moreover, the combined buffering policy of received time and hop count is compared with each. The result shows the order of performance: combined > hop count > received time.

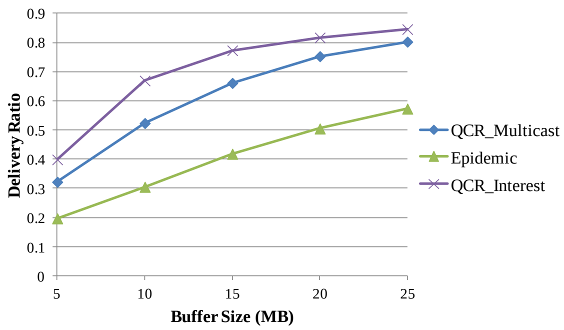


Fig. 11. Interest routing comparison: delivery ratio vs. buffer size.

The performance of interest-based routing was evaluated using the INFOCOM05 trace file. The native multicast version of QCR without CDV (QCR_Multicast) was compared with the interest-based version of QCR (QCR_Interest). Epidemic, which naturally supports both unicast and multicast, was also compared. QCR_Interest produced a substantially higher delivery ratio (Fig. 11) and incurred fewer relay costs (Fig. 12) than did QCR_Multicast. The effective latencies of the three routing schemes were similar (Fig. 13). This indicates that message relays through nodes that contact any nodes in the destination group help to increase the delivery ratio and decrease the relay count.

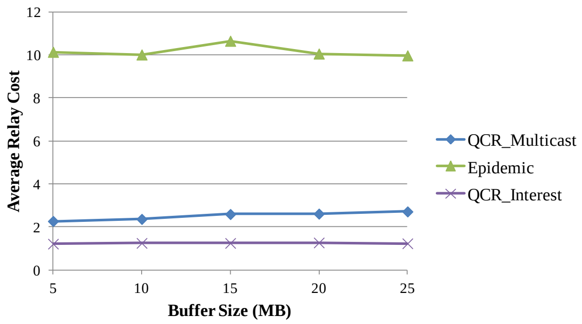


Fig. 12. Interest routing comparison: relay cost vs. buffer size.

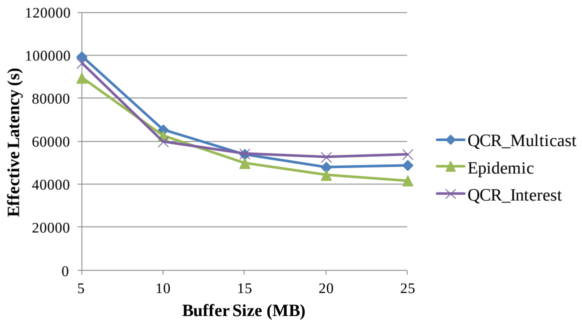


Fig. 13. Interest routing comparison: effective latency vs. buffer size.

QCR_Interest considers the contact information of the destination group or the contact information of all nodes. In the subsequent set of experiments the number of nodes in the destination group was controlled. Regardless of the destination group size (at least two), QCR_Interest always outperformed QCR_Multicast (Figs. 14 to 16). As the size increased, the probability that more relay nodes could contact nodes in the destination group increased; hence, more CDV data were used. Therefore, the delivery ratio of QCR_Interest increased slightly (Fig. 14).

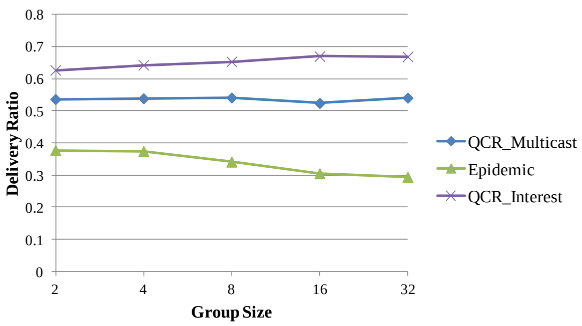


Fig. 14. Interest routing comparison: delivery ratio vs. group size.

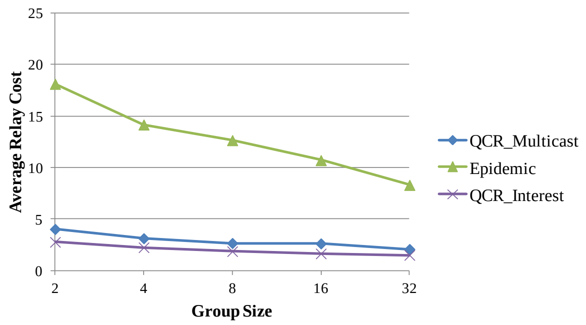


Fig. 15. Interest routing comparison: relay cost vs. group size.

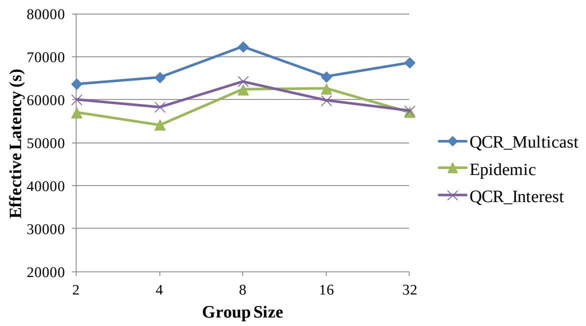


Fig. 16. Interest routing comparison: effective latency vs. group size.

5. Conclusions

Quota-replication routing has the excellent feature of limiting the number of message copies, reducing the network traffic loads of DTNs. However, existing routing protocols set fixed message quotas, which is not suitable for certain network conditions. In this paper, a quota control mechanism for dynamically adjusting message quotas is proposed in which the messages stored in the buffers of neighboring nodes are observed. To clean useless message copies that remain in the network, a low-cost probability-based method is introduced to replace the traditional i-list mechanism. To properly identify a suitable message holder, a new metric—contact density—is also introduced.

The proposed routing mechanism involves three components: quota allocation, quota adjustment, and buffer management. The benefit of the new quota allocation is due to the introduction of contact density. The advantage of quota adjustment is more significant when the network is much congested. The benefit of the new buffer management is also significant when the buffer space is small.

A multicast application of interest-based information dissemination is proposed to increase the flexibility of the proposed routing framework. Through the message forwarding of nodes with the same interests, an event message can be efficiently disseminated to all interested users. The experimental results show that the proposed routing protocol outperforms other well-known protocols according to different cost metrics. Future studies should consider an advanced routing framework by adding various features, such as service differentiation, quality-of-service guarantee, and load balancing.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful suggestions. This research was partially supported by National Science Council of the Republic of China under Contract No. NSC 100-2221-E-259-020.

References

[1] A. McMahon and S. Farrell, "Delay- and Disruption-Tolerant Networking," *IEEE Internet Computing*, vol. 13, no. 6, pp. 82-87, Nov. 2009.

[2] Delay-Tolerant Networking Research Group (DTNRG), http://www.dtnrg.org.

[3] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *in Proc. ACM SIGCOMM Conf.*, pp. 27-34, Aug. 2003.

[4] K. Scott and S. Burleigh, "Bundle Protocol Specification," *IETF RFC* 5050, Nov. 2007.

[5] C. Caini, R. Firrincieli, and M. Livini, "DTN Bundle Layer over TCP: Retransmission Algorithms in the Presence of Channel Disruptions," *Journal of Communications*, vol. 5, no. 2, pp. 106-116, Feb. 2010.

[6] A. Voyiatzis, "A Survey of Delay- and Disruption-Tolerant Networking Applications," *Journal of Internet Engineering*, vol. 5, no. 1, Jun. 2012.

[7] IPN Special Interest Group (IPNSIG), http://www.ipnsig.org/.

[8] The ZebraNet Wildlife Tracker, http://www.princeton.edu/~mrm/zebranet.html.

[9] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *Computer*, vol. 37, no. 1, pp. 78-83, Jan. 2004.

[10] Haggle, http://haggleproject.org/.

[11] D. Mohney, "Soaring Condor Relieves Headaches," *Mobile Radio Technology*, vol. 22, no. 6, pp. 16-23, Jun. 2004.

[12] UMass DieselNet, http://prisms.cs.umass.edu/dome/umassdieselnet/.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *in Proc. ACM SIGCOMM Workshop*, pp. 252-259, Aug. 2005.

[14] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-Based Routing in DTNs," *in Proc. IEEE INFOCOM Conf.*, pp. 846-854, Apr. 2009.

[15] A. Elwhishi and P. H. Ho, "SARP - A Novel Multi-Copy Routing Protocol for Intermittently Connected Mobile Networks," *in Proc. IEEE GLOBECOM Conf.*, pp. 1-7, Nov. 2009.

[16] S. C. Lo, M. H. Chiang, J. H. Liou, and J. S. Gao, "Routing and Buffering Strategies in Delay-Tolerant Networks: Survey and Evaluation," *in Proc. IEEE ICPP Workshop*, Sept. 2011.

[17] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Duke University, *Technical Report* CS-200006, Apr. 2000.

[18] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," *in Proc. IEEE INFOCOM Conf.*, pp. 1-11, Apr. 2006.

[19] A. Lindgren, A. Doria, and O. Scheln. "Probabilistic Routing in Intermittently Connected Networks," *LNCS*, Springer, vol. 3126, pp. 239-254, 2004.

[20] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical Routing in Delay-Tolerant Networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 8, pp. 943-959, Aug. 2007.

[21] E. M. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs," *in Proc. ACM MobiHoc Conf.*, pp. 32-40, Sept. 2007.

[22] A. Lindgren and K. S. Phanse, "Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks," *in Proc. Intl. Conf. Communication System Software and Middleware*, pp. 1-10, 2006.

[23] M. Seligman, et al, "Storage Routing for DTN Congestion Control," *Wireless Communications and Mobile Computing*, vol. 7, no. 1, pp. 1183-1196, 2007.

[24] Yue Cao, Cruickshank,H. ; Zhili sun , “Active Congestion Control Based Routing for Opportunistic Delay Tolerant Networks” *in Proc. IEEE Vehicular Technology Conf*. pp. 1-5, May 2011.

[25] N. Thompson, S. C. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets, "Retiring Replicants: Congestion Control for Intermittently-Connected Networks," *in Proc. IEEE INFOCOM Conf.*, pp. 1-9, Mar. 2010.

[26] S. C. Lo and C. L. Lu, "A Dynamic Congestion Control Based Routing for Delay-Tolerant Networks," *in Proc. Intel. Conf. on Fuzzy Systems and Knowledge Discovery*, pp. 2047-2051, May 2012.

[27] S. C. Lo and W. R. Liou, "Dynamic Quota-Based Routing in Delay-Tolerant Networks," *in Proc. IEEE VTC Conference*, May 2012.

[28] P. Mundur, M. Seligman, and G. Lee, "Epidemic Routing with Immunity in Delay Tolerant Networks," *in Proc. IEEE MILCOM Conf.*, pp. 1-7, Nov. 2008.

[29] W. Zhao, M. Ammar and E. Zegura, "Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms," *in Proc. ACM Workshop on Delay-Tolerant Networking*, pp.268-275, Aug. 2005.

[30] Q. Ye; L. Cheng, M. C. Chuah and B. D. Davison, "OS-multicast: On-demand Situation-aware Multicasting in Disruption Tolerant Networks," *in Proc. IEEE Vehicular Technology Conf*., pp. 96-100, May 2006.

[31] Y. Xi and M. C. Chuah, "An Encounter-Based Multicast Scheme for Disruption Tolerant Networks," *Computer Communications*, vol. 32, no. 16, pp.1742-1756, Oct. 2009.

[32] Y. Wang, X. Li, and J. Wu, "Delegation Forwarding in Delay Tolerant Networks Multicasting," *Journal of Communications*, vol. 6, no. 5, pp. 384-392, Aug. 2011.

[33] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *in Proc. ACM SIGCOMM Conf.*, pp. 145-158, Aug. 2004.

[34] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," *in Proc. Intl. Conf. Simulation Tools and Techniques*, Mar. 2009.

[35] The Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD), http://uk.crawdad.org/index.php.