JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 23, 1123-1139 (2007)

Integrated Buffer and Route Management in a DTN with Message Ferry

MOOI CHOO CHUAH AND WEN-BIN MA
Department of Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015, U.S.A.

Unlike normal wireless ad hoc networks, end-to-end connection may not exist in DTNs Thus, the Message Ferrying (MF) scheme has been proposed as a strategy for providing connectivity in disruption tolerant network (DTN)s, where a set of nodes called ferries are responsible for carrying messages for all nodes in the networks. In such store- and-forward networks, buffers at ferry and regular nodes become critical resources and need to be allocated fairly among different users. In this paper, we propose a max-min fairness model for a DTN with a message ferry. Based on this model, we propose a buffer allocation scheme that can achieve fairness among different sessions. We also design an integrated buffer and routing management scheme called buffer efficient routing scheme (BERS). Via simulations, we demonstrate that our fair buffer allocation scheme assigns buffers fairly to different sessions. Our simulation studies also show that BERS can achieve higher session throughput and lower packet delivery latency than the only-store-and-forward routing scheme that is typically used in a message ferry system.

Keywords: disruption tolerant network, buffer management, ferry route design, performance evaluation, fairness

1. INTRODUCTION

The advent of wireless communication technology opens up exciting new applications but also a number of challenges. One of the challenges is to maintain the communication links between participating entities as they move around. Wireless ad-hoc networks allow nodes to communicate with one another without any existing infrastructure. Many routing protocols have been proposed for ad-hoc networks [1, 2]. However, these algorithms assume that an end-to-end connectivity between any pair of nodes exists. Several real-life scenarios *e.g.* battlefield scenarios, node mobility, physical obstacles, and limited radio range, may prevent nodes from communicating with others and result in network partitions. Thus, it is interesting to explore how data can be delivered in a constantly disconnected network.

A number of schemes have been proposed recently to provide communications in highly partitioned networks. For example, in [3], the authors propose epidemic routing where a flooding-style mechanism is used by the nodes to forward messages to other nodes they meet. This scheme has the advantage of delivering all messages assuming no time and memory constraints. However, it incurs extra messages due to the blind flooding. In [4, 7] the authors proposed a message ferrying scheme, where a set of nodes

---

Received September 15, 2006; accepted February 6, 2007.
Communicated by Ten H. Lai, Chung-Ta King and Jehn-Ruey Jiang.

called “message ferries” are responsible for carrying messages between disconnected nodes. Similar proposal has been proposed for remote village communications [5] and tiered sensor network [6].

An interesting problem to investigate for a message ferry system is the issue of fairly allocating storage resources among the various active communication sessions since the storage available at a message ferry is limited. If we do not design proper buffer allocation scheme, some sessions may end up hogging the available buffers and leave other sessions starving for buffers. In this paper, we consider a remote village communication scenario where there are stationary nodes that accept messages and message delivery between the nodes can only be performed via a message ferry. We first define a new fairness model for DTN environments. Then, we describe a buffer allocation scheme that can achieve max-min fairness among different active sessions.

The rest of the paper is organized as follows: In section 2, we discuss some related work that defines fairness in wireless networks. Then in section 3, we describe the network model that we use and the different types of routes available in a DTN with a message ferry. We then define a path metric that is used to select a route among several available ones for a session. In section 4, we define the fairness model that we use for DTNs with message ferries. We also present a max-min fair buffer allocation scheme that we have designed and present some results using static session scenario. In section 5, we present some simulation results using dynamic session arrivals and departures to illustrate that the fair buffer allocation scheme we design can achieve its goal. We also show that using different types of routes can improve session throughputs. We provide some concluding remarks and future work that we intend to explore in section 6.

2. RELATED WORK

Much research work has been done on achieving throughput fairness in ad hoc networks [8-11]. Most of this work concentrates on how to fairly share the bandwidth available in wireless ad-hoc networks. [8, 9] concentrates on providing min-max fairness while [10, 11] address proportional fairness. TCP fairness in ad-hoc networks is also studied. For example, in [12], the authors studied TCP fairness among flows in an ad-hoc network that is connected to the Internet via gateway nodes. Simulations and testbed measurements are used to compare the throughputs of multihop flows to single-hop flows, wireless-to-wired flows, wired-to-wireless flows, and to study the fairness issues. The authors conclude that hidden terminals are the main reason for unfairness in TCP.

The problem we are addressing in this paper is different. The above cited work defines two flows as contending if either the sender/receiver of one flow is within the transmission range of the sender/receiver of the other. However, in DTN environments, the nodes may not be able to communicate with one another so they will not compete for wireless bandwidth but the traffic flows may be contending for buffer space at the message ferry.

Another category of related work is the design of optimal ferry route [13, 14]. In such work, the researchers assume that the ferry is used only to transport packets and hence its route can be optimally chosen to meet certain system requirements *e.g.* a certain message delivery latency requirement. In our work, we assume that the ferry route is static and predetermined *e.g.* if transportation bus is used as the message ferry, then the bus route has already been predetermined.

3. NETWORK MODEL AND TYPES OF ROUTES

3.1 Network Model

Before describing our fairness model for a message ferry system, we first describe the network model we assume. We consider a message ferry system where the regular nodes are geographically distributed such that they cannot communicate with one another directly. Such a scenario is common in remote village communications [5]. One or more message ferry (ferries) periodically visit these nodes to collect/deliver messages between them. Each regular node in the message ferry system has a full-duplex radio that allows it to simultaneously transmit and receive information from the message ferry.

We assume that each message ferry (MF) follows a fixed regular route and makes certain number of stops in its regular route. The stops are positioned such that the ferry can have the longest contact time with the visited nodes. Each ferry maintains the following information: (1) the sequence in which it visits different nodes, (2) the source/destination of each session (flow), the amount of buffer that has been allocated, the session lifetime, (3) the expected contact time at each location where the ferry stops to collect/deliver messages.

The ferry is expected to construct and maintain a routing table based on the information that it gathers as it travels along its regular route. The ferry is also responsible for the route selection and buffer allocation for different admitted sessions where a session refers to a communication flow between two regular nodes. Whenever a ferry reaches a node, it will first deliver packets either destined to that node or using that node as an intermediate storage node. Then, it will pick up packets that need to be delivered to other nodes.

Each regular node maintains a local request list that records all sessions that have arrived and their session durations. When the message ferry visits a node, the node will inform the message ferry of any new session arrivals. The MF uses this information to update a global request list that it maintains. For each active session, the MF will check all possible paths for that session and select the best one according to a path metric which will be discussed in the next subsection.

3.2 Types of Routes

There are several types of routes for delivering messages from one node (or group) to another.

3.2.1 Type 1 route: direct connected route

In some scenarios, the message ferry can communicate with more than one node simultaneously. In this case, the messages between such nodes can be sent directly without being stored at the message ferry. Fig. 1 illustrates a simple scenario where the message ferry can simultaneously communicate with Node 1 and Node 2 but not to Node 3.

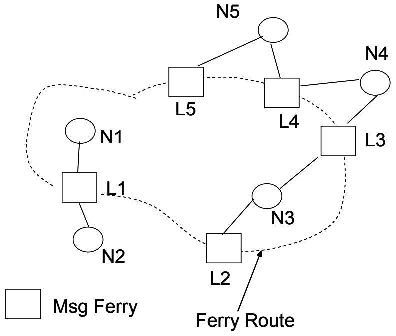


Fig. 1. A simple DTN with one message ferry.

Thus, any messages between Node 1 and Node 2 can be delivered directly. Normally, even for direct communication, some buffers need to be allocated (*e.g.* at the MAC layer to store the packets to be transmitted). In our work, we assume that such buffers are allocated separately, and are excluded from the regular buffers we considered for message ferry (denoted by BS (Kbits)).

3.2.2 Type 2 route: store-and-forward route

From Fig. 1, we observe that messages from Node 2 to Node 3 or messages from Node 1 to Node 3 need to be stored at the regular buffers of the message ferry until the ferry visits Node 3. Such a route is referred to as a Type 2 route.

3.2.3 Type 3 route: concatenation of two directly connected hops

One can also deliver messages via a concatenation of two directly connected hops. For example, messages from Node 3 to Node 5 can be sent first to Node 4, stored at Node 4 and delivered only when the ferry is at location L4 where it can communicate directly with both Node 4 and Node 5. We refer to this type of route as a Type 3 route. A Type 3 route allows us to use buffers available at regular nodes so that the buffers at the message ferry can be used by messages from sessions that can only be delivered via Type 2 routes. The message delivery time for a Type 3 route may be higher than a Type 2 route.

3.2.4 Type 4 route: concatenation of Type 1 & Type 2 routes

Another delivery method is via a concatenation of Type 2 and Type 1 routes. For example, messages from N1 to N4 can be delivered by a Type 2 store-and-forward route to Node 3, stored at Node 3, and delivered only when the ferry is at location L3. Similar to Type 3 routes, Type 4 routes also allow us to use buffers available at the regular nodes so that the buffers at the message ferry can be utilized by sessions whose messages can only be delivered via Type 2 routes.

3.2.5 Path metric

Sessions that use direct connected routes will be admitted or rejected based on the data transfer volume availability (which is a function of contact duration and wireless bandwidth). For example, if we set this data transfer volume availability to be the same as the ferry buffer size (denoted as BS), then a ferry will allow two nodes to transfer this much traffic when the ferry is at a location that allows it to connect these two nodes. Sessions that use stored-and-forward routes will be admitted or rejected depending on the ferry buffer availability. Thus, after the path selection process, the buffer allocation process will be triggered. A session path may consist of a store-and-forward route segment followed by a direct connected route segment. This means that if packets are delivered to an intermediate node when the next route segment is not ready yet, then the packets will be stored at this intermediate node until that route segment becomes available.

To choose between different routes, we need to define a path metric (PM). Traditional ad hoc routing protocol often use hop count as the path metric. To find a suitable metric, we need to understand the characteristics of the DTN scenarios that we are studying. A direct connected route segment (Type 1 route) connects two nodes via the message ferry. Type 1 route may only be available for a certain period of time. Thus, packets from a particular session may need to be buffered at regular nodes if a Type 1 route selected for this session is not available. On the other hand, a store-and-forward route contends for a different resource, namely the ferry buffer space. If the ferry buffer space is a limited resource, then one may use a direct connected route (if one exists) before using the store-and-forward route. Among the different available store-and-forward route segments, the one with the shortest distance should be selected since that means the ferry buffer space can be released the soonest. In addition, we want to pick a session path that allows the packets from a session to be delivered the earliest *i.e.* the packets enjoy small packet delivery delay.

The above discussions motivate us to define the path metric for this work as follows: the path metric consists of two components: (a) path delay (PD), (b) ferry transportation cost (FTC).

$$PM = w_1 PD + w_2 FTC \tag{1}$$

The path delay is the sum of the delay in traversing each route segment of a path including the waiting time for the route segment to be available. The path delay is normalized with the ferry round-trip time. The ferry transportation cost (FTC) is a product between buffer share allocated to a session and the ferry occupation time. To allow for easy comparison of these two components, we decide to use the normalized version of the path delay and the ferry transportation cost. Thus, the FTC used is defined as follows:

$$FTC = \frac{FBS}{BS} \times \frac{FBOT}{SSFOT} \tag{2}$$

where FBS is the ferry buffer share allocated to that session, BS is the ferry buffer size, FBOT is the ferry buffer occupation time if that route (only for Type 2 route) is chosen and SSFOT is the ferry buffer occupation time if a store-and-forward (SF) route is chosen.[1] The weights, w_1 and w_2, are provided to allow the system designer to put different

---

[1] SSFOT assumes that ferry carries the message from source to destination using a SF route while a path may only use a Type 2 route as a delivery segment and hence FBOT is smaller than SSFOT.

emphasis on the two components. If w_1 is set to 0, it means a system designer cares more for the resources at the ferry. Similarly, if w_2 is set to 0, it means a system designer cares for the packet delivery delay. The routing scheme that uses this path metric with $w_1 = w_2 = 0.5$ will be referred to as the buffer efficient routing scheme (BERS) and we will compare its performance with a routing scheme that uses only stored and forward paths (OSFRS) in section 5.

Since we desire to support multiple sessions in a DTN with message ferries such that each session enjoys similar session throughput, we need a fair buffer allocation strategy. Thus, we discuss this issue in the next section.

4. MAX-MIN FAIRNESS AND BUFFER ALLOCATION SCHEME

The buffers available in a message ferry are limited. Thus, in a DTN scenario where the nodes can communicate only via a message ferry, we define contending flows as follows: two flows are contending if their messages need to be stored simultaneously at the message ferry before the messages arrive at their respective destinations.

A formal definition of max-min fairness for a DTN with a message ferry is described below.

Assume the wireless link bandwidth and contact time is such that all data transfer can be completed. Let *N* be the set of nodes that the ferry visits, and let *L*(*h*) be the set of all sessions on the route segment h of the message ferry. Assuming that all sessions are treated equally, then an allocation of buffer U_s for session *s* is Max-Min fair if it is feasible (*i.e.* $\sum_{s \in L(h)} U_s \leq BS,\ \forall h \in H,$ where *L*(*h*) is the number of sessions sharing the route segment *h*), and for each session *s*, U_s cannot be increased (while maintaining feasibility) without decreasing $U_{s'}$ for some session *s*′ for which $U_{s'} \leq U_s$. If the sessions are assigned different weights (to reflect different QoS classes), then a buffer allocation U_s for session *s* is Max-Min fair if it is feasible and for each session *s*, U_s cannot be increased (while maintaining feasibility) without decreasing $U_{s'}$ for some session *s*′ for which $U_{s'}/ws' \leq U_s/ws$, where *ws*′ and *ws* are the session weights of session *s* and *s*′ respectively.

4.1 Fair Buffer Allocation Scheme

Next, we describe a fair buffer allocation scheme.

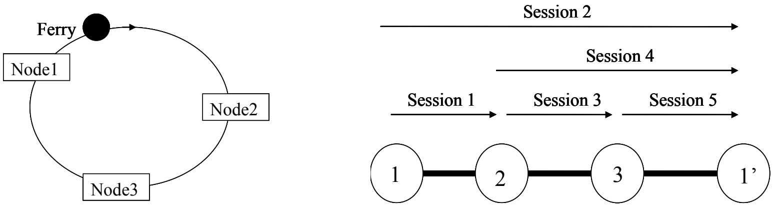
Let *Di* be the buffer allocated to session *i* and let *S* be the set of all sessions whose buffer allocations have not been fixed. Let *L* be the set of all route segments that constitute the regular route of a message ferry. Let *BS* be the buffer space available at the message ferry and let B_l be the current buffer space utilized at route segment *l*.

The algorithm step is described as follows:

1. Initially, $Ds = 0$ for any session *s*, and $B_l = 0$ for any route segment.
2. At each step, *S* denotes the set of sessions not passing through any buffer saturated link; *L* denotes the set of route segments where ferry buffer is not yet saturated. $\sum w_l$ denotes the sum of session probabilities of sessions using link *l*. Finally, Δr indicates the unitary rate increment and w_s is the session probability of session *s*.

Table 1. Session weights for network 2.

Session (s)	Session Weights (w_s)
1	1/2
2	1/2
3	1/2
4	1/2
5	1



(a) Network 2.

(b) Contention group for network 2 (1' indicates node 1 in the next round).

Fig. 2. A simple example.

3. At each step, the algorithm computes for each route segment the sum of the weights of all sessions in S that use that particular route segment. The unitary rate increment is determined as $\Delta r = \min_{l \in L}\left(\frac{BS - B_l}{\sum w_l}\right)$ and buffer demand of sessions not passing through any buffer saturated segments are increased proportionally to their weights: $D_s = D_s + \Delta r \times w_s$.
4. After each step, the sets L and S are updated accordingly. The route segment that has reached buffer saturation will be removed from the set L and the sessions whose buffer allocation are fixed (*i.e.* they go through that buffer saturated segment) are also removed from the set S.
5. Repeat steps 3 & 4 until set S is empty.

We use a simple example described in Fig. 2 to show how the algorithm works. In this example, we assume that we have 5 sessions and three route segments. The node shown here is a logical node. Thus, each node may potentially be a cluster of real physical nodes. We further assume that there are two physical nodes within Node 1 that cannot communicate with one another and rely on the Ferry to deliver messages between themselves. This particular flow is shown as section 2 in Fig. 2 (a).

Initially,

$L = \{1, 2, 3\}$; $S = \{1, 2, 3, 4, 5\}$, (three route segments, and five sessions)

$BS = 1$, (assume there is only one ferry and its buffer size is one unit)

$D1 = 0$; $D2 = 0$; $D3 = 0$; $D4 = 0$; $D5 = 0$, (buffer demand for each session is 0)

$B1 = 0$; $B2 = 0$; $B3 = 0$. (buffer used by all sessions at each route segment is 0)

Let us start with route segment 3 (between node 3 to node 1'). Since $(BS - B3) = 1.0$ and the sum of session weights $\sum_3 = (1/2 + 1/2 + 1) = 2$ (sessions 2, 4 and 5 go through route segment 3). Thus, $\Delta r = 1/2$. The buffer allocation for these 3 sessions is then increased using $D_s = D_s + \Delta r \times w_s$, and we get the following values:

$D1 = 1/4$; $D2 = 1/4$; $D3 = 1/4$; $D4 = 1/4$; $D5 = 1/2$.

Now route segment 3 is buffer saturated ($B3 = 1$) so the buffer allocations for sessions 2, 4 and 5 which goes through this segment are fixed. After updating L and S, we have

$L = \{1, 2\}$; $S = \{1, 3\}$,
$D1 = 1/4$; $D3 = 1/4$,
$B1 = 1/2$; $B2 = 3/4$; $B3 = 1$.

Since $\frac{BS - B_1}{\sum_1} = \frac{1 - 1/2}{1/2} = 1$ and $\frac{BS - B_2}{\sum_2} = \frac{1 - 3/4}{1/2} = 1/2,$ we pick segment 2 in our second step as the segment to be buffer saturated, and $\Delta r = 1/2$. The buffer allocations for sessions 1 and 3 are thus increased by an amount proportional to their weights, and we get the following results:

$D1 = 1/2$; $D2 = 1/4$; $D3 = 1/2$; $D4 = 1/4$; $D5 = 1/2$.

Now segment 2 is buffer saturated and hence the buffer allocation for session 3 is fixed. Then, we have

$L = \{1\}$; $S = \{1\}$,
$D1 = 1/2$,
$B1 = 3/4$; $B2 = 1$; $B3 = 1$.

Next, we consider the segment 1, which is the only segment left that has not yet become buffer saturated. We get $\Delta r = 1/2$, thus the final results are as follow:

$D1 = 3/4$; $D2 = 1/4$; $D3 = 1/2$; $D4 = 1/4$; $D5 = 1/2$.

4.2 A More Complex Scenario

In Fig. 3, we show a more complex DTN with a message ferry. We assume that the message ferry has a fixed route. We also assume that the wireless bandwidth and the contact time is large enough to allow all data transfer in every ferry cycle (denoted as T). There are six nodes in Fig. 3. In this example, node 1 (G1) is visited three times in each cycle. The various locations marked with a square denote the various stops that the message ferry will make. Let us assume that the ferry starts from location L0. At location L1, the ferry can communicate with both nodes G1 and G2. In location L2, the message ferry can only communicate with G2 and so on. The ferry route repeats after the ferry returns to location L0 again after visiting location L8.

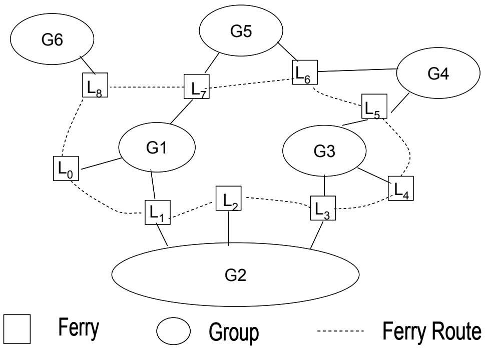


Fig. 3. A more complex DTN scenario.

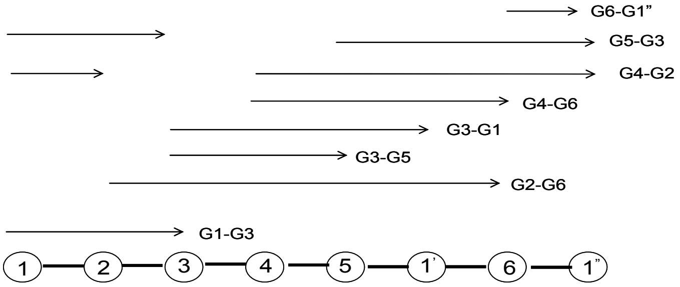


Fig. 4. Resource contention graph.

Let us assume that there are 10 sessions: (a) from G1 to G3, (b) from G1 to G5, (c) from G2 to G3, (d) from G2 to G6, (e) from G3 to G5, (f) from G3 to G1, (g) from G4 to G6, (h) from G4 to G2, (i) from G5 to G3, (j) from G6 to G1.

4.2.1 Using only Type 1 & Type 2 routes

Assume that we only allow message delivery via Type 1 and Type 2 routes, then the resource contention graph for eight relevant sessions that use Type 2 routes is shown in Fig. 4. Since the ferry can communicate with G1 at three locations (denoted as L_1 , L_7 and L_9 in Fig. 3) in each ferry tour, we use three nodes marked as 1, 1' and 1'' to represent this fact in Fig. 4. Messages from session G2-G3 and session G1-G5 will be delivered using Type 1 routes. Thus, these two sessions are not considered in the resource contention graph.

From Fig. 4, we can see that the route segments 4-5 and 5-1' are the bottleneck segments. Each of them has 5 active sessions. Thus, each of the sessions G2-G6, G3-G5, G3-G1, G4-G6 and G4-G2 is allocated $BS/5$ when we consider the segment 4-5 and session G5-G3 will also be allocated $BS/5$ when we consider the segment 5-1'. Sessions from these bottleneck segments are then removed from the set S. Thus, after the first two steps, we see that the next bottleneck segment is segment 6-1' and session G6-G1 can be allocated $3BS/5$. Then, by considering segment 1-2, we see that session G1-G3 can be allocated $3BS/5$ as well. Using such buffer allocations, the message ferry can deliver $12BS/5$ worth of data every ferry cycle for the 8 DTN sessions that use Type 2 routes.

4.2.2 Using Type 1, Type 2 & Type 3 routes

If we include Type 3 routes, then we can remove sessions G1-G3, G3-G1, G3-G5, G4-G2 and G5-G3 from the resource contention graph since messages from these sessions can be delivered via Type 3 routes. So, only 3 sessions (shown in Fig. 5) that utilize Type 2 routes remain. Sessions G2-G6, G4-G6 will be allocated BS/2 of the message ferry buffer and session G6-G1' will be allocated the whole message buffer. As a result, the message ferry will transfer 2BS worth of data every ferry cycle for these 3 sessions that use Type 2 routes.

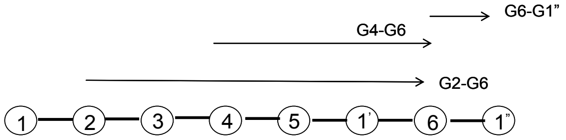


Fig. 5. Revised resource contention graph.

Of course, one has to make sure that the wireless bandwidth and the contact time the message ferry make with the nodes at various visiting locations is large enough such that those messages that are using Type 1 and Type 3 routes can be delivered. In addition, we also assume that the buffers at the regular nodes are big enough to store those messages that are delivered via Type 3 routes. Furthermore, we assume that there are no delivery deadlines for these messages.

4.2.3 Discussions on scenarios with dynamic session arrivals

The above discussion assumes static scenarios where the number of active sessions does not change with time. In real network scenarios, sessions arrive and depart dynamically. Thus, we discuss here how our algorithm deals with dynamic session arrivals and departures. We assume that each regular node stores new session arrivals (*e.g.* the destination nodes of these new sessions, the session lifetimes, the QoS class *etc.*). When a message ferry visits a node, the node informs the message ferry of any new session arrivals and their associated session information. The message ferry stores all active session information for which buffer allocations need to be made. At each location, the message ferry removes expired sessions from its active session list and reclaim the buffers that have been assigned to expired sessions. The buffer allocation for each active session is only re-computed once per ferry cycle time when the ferry visits a node.

Using the same network example in Fig. 3 and assume that there are 10 active sessions when the message ferry is at location L_0: 2 sessions are from node G1, 4 sessions not from G1 but whose packets have been delivered, and 4 sessions not from G1 whose packets have not delivered yet. Thus, packets from the last 4 sessions still remain in the ferry buffer. Let us denote the occupied buffer space as B_a. Further assume that 2 new sessions have arrived at node G1 since the last ferry was at location L_0. Therefore, the ferry will make new buffer allocation of the remaining buffer (BS-B_a) for the 4 sessions from G1 (including the 2 new ones) and the 4 sessions whose packets have been delivered. Sessions that use route segments that support fewer sessions will enjoy higher throughput than sessions that happen to use congested route segment. In addition, we assume that only BS worth of data will be transferred during each direct connected route opportunity. In the next section, we explore the performance of using this slightly varied buffer allocation scheme in a dynamic session arrival/departure scenario.

5. SIMULATION STUDIES

In this section, we present our simulation studies of the integrated buffer and route management scheme we have designed. First, we describe our simulation setup. Then, we present our simulation results for different scenarios.

5.1 Simulation Setup

To study how our fair allocation scheme performs in an environment where sessions can arrive and depart, we resort to simulations. We simulate the DTN shown in Fig. 3. The regular nodes are assumed to be stationary. The default scenario is that there is only one message ferry. We assume that the sessions arrive to the regular nodes according to a Poisson process. The session inter-arrival time is assumed to be exponentially distributed with a mean of 15 seconds. The session duration time is assumed to be exponentially distributed with a mean of 450 seconds. The ferry cycle time is assumed to be 325 seconds. We assume that packets within a session arrive periodically at a certain rate. We allow new sessions to arrive until a simulation time of 3,000 seconds (about 200 sessions) and run the buffer allocation algorithm until 4,000 seconds. The first two ferry cycles are used for warming up period where the ferry gather useful information and run initial route selection and buffer allocation scheme.

In our simulation study, we consider two routing schemes, (a) the Only Stored-and-Forward scheme (OSFRS) where only the stored-and-forward route will be used to deliver messages between any two nodes, and (b) the buffer efficient routing scheme (BERS) which uses the path metric with certain w_1 and w_2 settings to select the best route and also the fair buffer allocation scheme that we describe earlier.

The performance metrics we use to compare different routing schemes are as follow:

a) Average Path Delay: Path delay represents the average time for the packets of a session to travel along the path from the source to the destination. We compute the path delay for each session, and average the path delays over all admitted sessions.
b) Average Ferry Buffer Utilization: Ferry Buffer Utilization (FBU) is defined as follows:

$$FBU = \frac{\sum_{i=1}^{N} UsedBuffer_i * T_i}{T_{cycle}}$$

where N is the number of nodes visited by the ferry in each ferry cycle, $UsedBuffer_i$ is the total normalized ferry buffer utilized at node i; T_i is the time it takes for the ferry to visit the next node in the sequence from the current node i where the buffer utilized does not change; and T_{cycle} is the ferry cycle time. The average ferry buffer utilization is calculated by averaging the ferry buffer utilization over all the different ferry cycles.

c) Average Session Throughput: Session throughput is calculated by dividing the total buffer share allocated in each ferry cycle over the session duration assuming that there is enough data from each session to fill the buffer share allocated. The values for all sessions are averaged and reported.

5.2 Comparison of BERS and OSFRS Schemes

In our first set of experiments, compare the performance achieved in the DTN using two routing schemes, namely (a) the only store-and-forward scheme (OSFRS) that uses only Type 2 routes and (b) the BERS scheme that uses the path metric defined in section 3.2.5. Figs. 6 and 7 plots the average session throughput and the average packet delay achieved using OSFRS and BERS when the packet arrival rate is varied (note that the session arrival rate remains unchanged). The results show that BERS can achieve 300% more session throughput than the OSFRS. The average packet delay achieved in BERS is 36% smaller than that achieved using OSFRS.

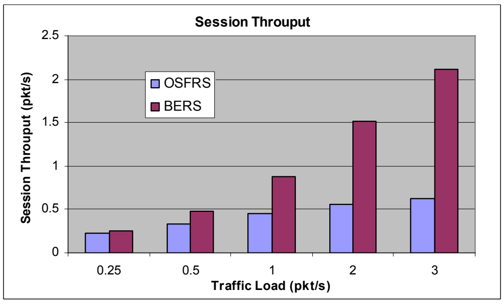


Fig. 6. Average session throughput vs. traffic load.

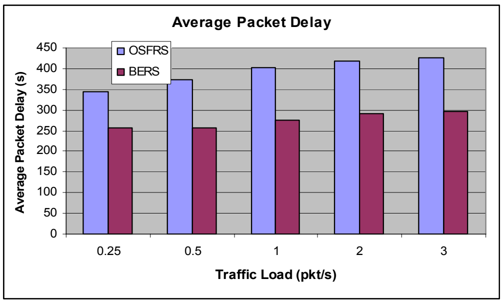


Fig. 7. Average packet delay vs. traffic load.

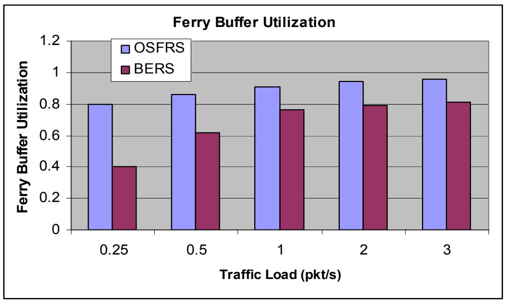
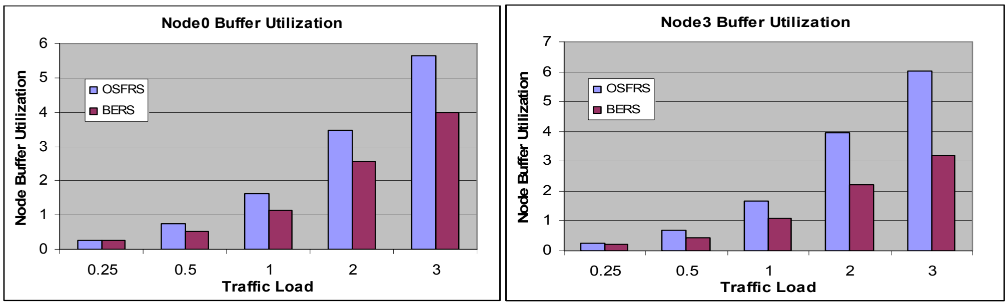


Fig. 8. Ferry buffer utilization.

We also compute the ferry and regular node buffer utilization. Fig. 8 shows the ferry buffer utilization while Figs. 9 (a), (b) show the regular node buffer utilization at node 0, and node 3 respectively. The plots in Fig. 8 show that the BERS allows ferry buffer to be used more efficiently and hence the ferry buffer utilization is lower when BERS is used.



(a) Node 0 buffer utilization. (b) Node 3 buffer utilization.

Fig. 9. Regular node buffer utilization.

Similarly, we see that the regular node buffer utilization using BERS is smaller. Thus, as a whole, the BERS is efficient in achieving higher throughputs and more efficient usage of ferry and regular buffers.

5.4 Session Fairness

Next, we evaluate the effectiveness of the fair buffer allocation scheme. Specifically, we compare the data delivery performance with and without the fair buffer allocation scheme. The data generation rate within a session is varied from 0.1 pkt/sec to 3 pkt/sec. The ferry buffer size is set to 1,000 packets. Each regular node buffer size is set to unlimited but we monitor its usage during our simulation. Each data point reported in this section is the average of five runs. The following two scenarios are simulated:

Scenario1 – FIFO (without Fair Buffer Allocation scheme): in this scenario, all packets are treated equally at the regular nodes; all the packets from different sessions at a node are placed in a single FIFO queue. The packets will be served according to their arrival times. When a MF arrives, the packets with oldest arrival time will be sent first. The ferry will also treat all packets equally and always try to fill its buffer with as many packets as possible.

Scenario2 – FBA (with Fair Buffer Allocation scheme): in this scenario, all the packets of a session will be placed in a separate FIFO queue at a regular node. When a MF arrives, each session with non-empty packets will be given a certain buffer allocation. The ferry will pick up as many queued packets as allowable by the allocated buffer share.

Additional performance metric called packet transmission fairness is used. It is defined as follows:

Data Transmission Fairness: We use the Jain's Fairness Metric to describe how fair the buffer allocation scheme is to different active sessions. The allocated ferry buffer share translates directly to the number of packets being picked up by the ferry and hence we refer to this fairness as the packet transmission fairness. The packet transmission fairness for a certain ferry cycle is defined as $fairness = \frac{\left(\sum X_i\right)^2}{\left(n \cdot \sum X_i^2\right)}$, where X_i is the number of

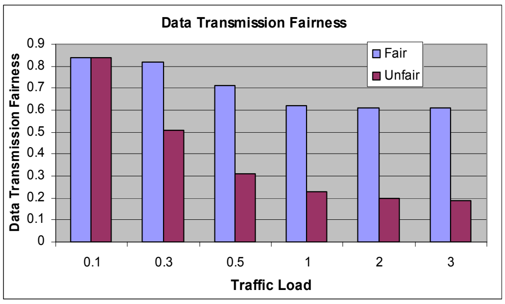


Fig. 10. Data transmission fairness.

transmitted packets for session *i*, and *n* is the number of the sessions picked up in this ferry cycle.

From Fig. 10, we see that the fair buffer allocation scheme allows us to achieve better fairness when compared to the case without the fair buffer allocation scheme. The fairness index decreases with increasing traffic load since the ferry uses latest session arrival information at the node being visited but old session arrival information at other nodes to recompute the session buffer allocation share. At low load, fewer packets arrive so we may not see the impact of using old session arrival information. At higher load, we can see this impact.

5.5 More Complex Scenario

In this section, we present results with a more complex network scenario. In this new scenario (40-node scenario), we have 40 nodes randomly distributed over a 4,000 × 4,000 m^2 area. The default transmission range is 100 m. At this transmission range, the nodes are sparsely distributed so they cannot communicate with one another directly. Using LKH traveling salesman algorithm, a ferry route that visits every node is designed. With a ferry speed of 10 m/s, a ferry cycle time (the time it takes to visit all 40 nodes) is 2,100 second. We then assume there is one session at each node and the session destination is randomly picked from the remaining 39 nodes. So, there is a total of 40 sessions and each session generates a traffic rate that ranges from 0.25 pkts/sec to 3 pkts/sec (depending on whether one or more ferries are used).

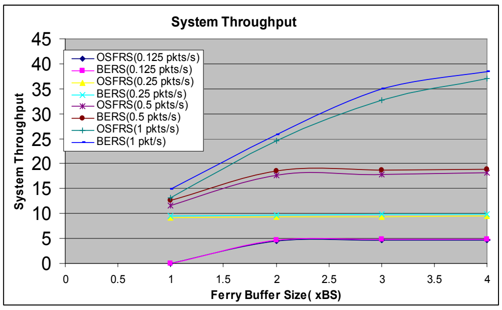


Fig. 11. System throughput vs ferry buffer size.

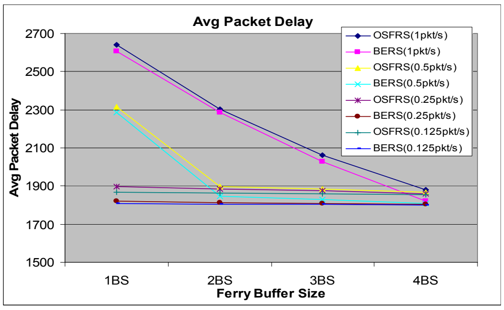


Fig. 12. Average packet delay vs ferry buffer size.

The ferry buffer size is varied from 10,000 pkts (denoted as BS) to 40,000 pkts. The message expiry time is set to 1.5 ferry cycle time (FCT). Figs. 11 and 12 show the system throughput (pkts/sec) and the average packet delay respectively as the ferry buffer size changes from one to four BS. We see that BERS performs slightly better system throughput than OSFRS. 70% of the routes for the 40 sessions are store-and-forward routes. That's why we only see slight improvement with BERS. With larger ferry buffer size, the average packet delay drops since more packets can be carried by the ferry. The ferry buffer utilization is tabulated in Table 2.

Table 2. Ferry buffer utilization.

Ferry Buffer Utilization	Ferry Buffer 10,000 packets		Ferry Buffer 20,000 packets		Ferry Buffer 30,000 packets		Ferry Buffer 40,000 packets	
Traffic Load (pkt/s)	OSFRS	BERS	OSFRS	BERS	OSFRS	BERS	OSFRS	BERS
0.125	0.37	0.35	0.19	0.17	0.126	0.115	0.092	0.087
0.25	0.75	0.71	0.378	0.348	0.25	0.23	0.18	0.17
0.5	0.92	0.9	0.74	0.69	0.5	0.46	0.37	0.35
1	0.93	0.91	0.92	0.89	0.89	0.86	0.73	0.7

6. CONCLUSIONS

In this paper, we have identified a few route types in a DTN with message ferry. We also define a buffer-based max-min fairness model and propose a fair buffer allocation scheme for DTNs with a message ferry. In addition, we propose a buffer efficient routing scheme that provides both fair buffer allocation and flexible route selection based on the observed path delay and ferry transportation cost. Via simulations, we show that our buffer allocation scheme allows different sessions to receive their fair buffer shares. Our simulation results for a six-node network demonstrate that the BERS can achieve 300% more session throughput than the OSFRS typically used in a message ferry system. In addition, the average packet delivery latency in BERS is 80% smaller than that in the OSFRS. To demonstrate that BERS is equally useful in more complex network scenario, we also performed simulation studies using a 40-node scenario. These 40 nodes are distributed over a 4,000 × 4,000 m^2 area. Our simulation results indicate that with appropriate transmission range, more routes of different types are available and hence better delivery performance can be achieved using BERS.

There are several topics that we are interested in pursuing. For example, in this paper, we only consider one class of traffic. We intend to investigate the impact of having different classes of traffic. We also intend to look into more complex DTN scenarios and scenarios with more than one ferry.

ACKNOWLEDGMENTS

This work was sponsored by Defense Advanced Research Projects Agency (DARPA) under contract No. W15P7T-05-C-P413. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. This document is approved for public release, unlimited distribution.

REFERENCES

1. C. Perkins and E. Royer, "Ad-hoc on demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90-100.
2. D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, 1996, pp. 153-181.
3. A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report No. CS-200006, Department of Computer Science, Duke University, 2000.
4. W. Zhao and M. Ammar, "Message ferrying: proactive routing in highly partitioned wireless ad-hoc networks," in *Proceedings of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems*, 2003, pp. 308-314.
5. A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations," *IEEE Computer*, Vol. 37, 2004, pp. 78-83.
6. R. C. Shah, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 30-41.
7. R. Viswanathan, J. Li, and M. Chuah, "Message ferrying for constrained scenarios," in *Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, 2003, pp. 487-489.
8. X. Huang and B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001, pp. 221-231.
9. H. Luo, J. Cheng, and S. Lu, "Self-coordinating localized fair queueing in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, Vol. 3, 2004, pp. 86-89.
10. L. Tassiulas and S. Sarkar, "Max-min fair scheduling in wireless networks," in *Proceedings of IEEE INFOCOM*, 2002, pp. 320-328.
11. Y. Yi and S. Shakkottai "Hop-by-hop congestion control over a wireless multihop network," in *Proceedings of IEEE INFOCOM*, 2004.
12. K. Xu, *et al.*, "TCP behavior across multihop wireless networks and the wired internet," in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia*, 2002, pp. 41-48.
13. W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2004, pp. 187-198.
14. W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proceedings of IEEE INFOCOM*, 2005, pp. 1407-1418.

Mooi Choo Chuah received her Ph.D. and M.S. degrees in Electrical Engineering from University of California San Diego, in 1991 and 1988, respectively, and her B.S. degree in Electrical Engineering (1st Class Honors) from University of Malaya, in 1984. She is currently an associate professor in Computer Science and Engineering Department at Lehigh University. Before she joined Lehigh, she spent 12 years at Bell Laboratories doing research in wireless networking. She has been awarded 42 US patents in multiple areas *e.g.* resource management, wireless MAC design, mobility management *etc*.

Wen-Bin Ma received his B.S. degree in Electronic Information Engineering from Southeast University, China in 1998. He spent 3 years working as a hardware/software engineer at Jiangsu Multimedia Communications (China Telecom) after graduation. He joined Lehigh University in Jan 2002 and got his M.S. degree in Computer Science in May 2004. He is currently a Ph.D. candidate at Lehigh University.