

Integrated Buffer and Route Management in a DTN with Message Ferry

Mooi Choo Chuah Wen-Bin Ma
chuah@cse.lehigh.edu wem2@lehigh.edu
Department of Computer Science and Engineering
19 Memorial Drive West, Lehigh University
Bethlehem, PA 18015, USA

Abstract

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. .

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 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],[9],[10],[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.

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]. There will be one or more message ferry (ferries) that will 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 ferry 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.

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. Figure 1 illustrates a simple scenario where the message ferry can simultaneously communicate with Node 1 and Node 2 but not to Node 3. 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)).

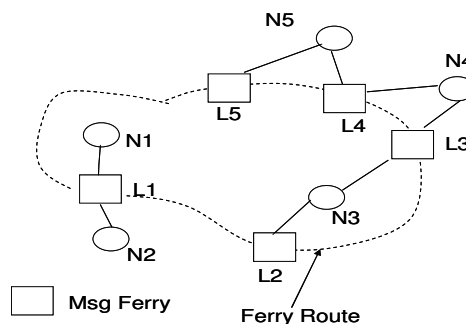


Figure 1: A simple DTN with one message ferry

3.2.2 Type 2 Route: Store-and-Forward

From Figure 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 and Type 2 routes

Another delivery method is via a concatenation of Type 2 and Type 1 route. 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

To choose between different routes, we define a path metric (PM) that consists of two components: (a) path delay (PD), (b) ferry transportation cost (FTC).

$$PM = w_1 PD + w_2 FTC$$

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 normalized buffer share allocated to a session and the normalized ferry occupation time as shown below:

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

(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 chosen¹. The routing scheme that uses this

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.

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.

Next, we discuss our fair buffer allocation strategy.

4 Buffer-Based 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 D_i 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, $D_s = 0$ 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 .

Table 1: Session weight for Network 2

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

Finally, Δr indicates the unitary rate increment and w_s is the session probability of session s .

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.

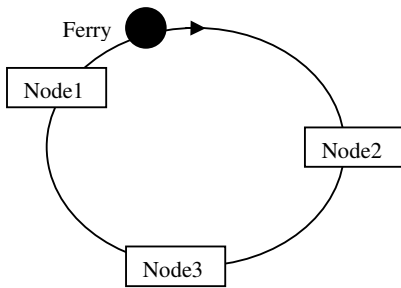


Figure.2a. Network 2

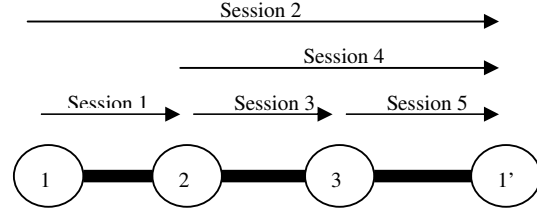


Figure.2b. Contention graph of Network 2 (1' indicate node1 at next round)

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 node3 to node 1'). Since $(BS - B3) = 1.0$ and the sum of session weights $\sum_3 = (1/2 + 1/2 + 1) = 2$ (session 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 session 2, 4 and 5 which goes through this segment are fixed. After updating L and S , we have

$L = \{1, 2\}$; $S = \{1, 3\}$

$D1 = 1/4$; $D3 = 1/4$;

$B1 = 1/2$; $B2 = 3/4$; $B3 = 1$;

$$\frac{BS - B_1}{\sum_1} = \frac{1 - 1/2}{1/2} \quad \frac{BS - B_2}{\sum_2} = \frac{1 - 3/4}{1/2} =$$

Since $\sum_1 = 1/2$ and $\sum_2 = 1/2$, 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 Figure 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 Figure 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 $L8$.

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 Figure 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.

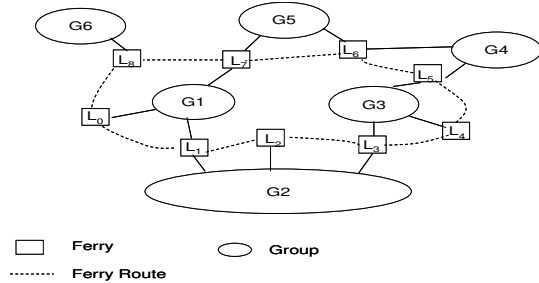


Figure 3: A more complex DTN scenario

From Figure 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.

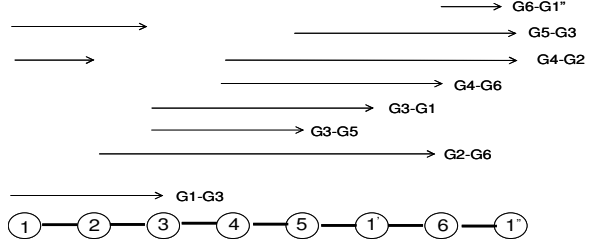


Figure 4: Resource Contention Graph

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.

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.

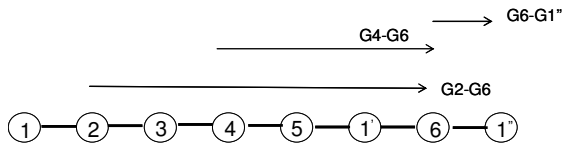


Figure 5: Revised resource contention graph

5 Simulation Studies

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 will store 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 will inform 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 those sessions whose lifetimes have expired from its active session list and reclaim the buffers that have been assigned to these sessions that have ended. The buffer allocation for each active session is only re-computed once per ferry cycle time when the ferry visits a node. Using the network example in Figure 3 and let us 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.

5.1 Simulation Set Up

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 Figure 3. We assume that the sessions arrive according to a Poisson process. The session interarrival 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 allow new sessions to arrive until a simulation time of 3000 seconds (about 200 sessions) and run the buffer allocation algorithm until 4000 seconds.

In our first set of experiments, we compare the performance achieved in the DTN shown in Figure 3, 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. The average session throughput achieved for sessions utilizing the ferry under the 2 schemes are tabulated in Table 2. The results indicate that by using Type 1, 2 and 3 routes, BERS achieves nearly 100% more session throughputs (73.1 Kbps versus 41.3 Kbps with a weight assignment of (0.5,0.5)).

Table 2: Average Session Throughput

	Avg throughput for sessions using Ferry (Kbps)	# of sessions using Ferry	# of sessions not using Ferry
OSFRS	41.3	109	0
BERS with (0,1)	84.7	42	67
BERS with (1_0)	65.4	64	45
BERS with (0.5_0.5)	73.1	57	52

The average ferry buffer utilization is shown in Table 3. The results indicate that since more sessions (all 109 sessions) are using the ferry, the average ferry buffer utilization for OSFRS is high. Only 55 sessions use the store-and-forward route under the BERS scheme, and hence the average ferry buffer utilization is only 81%.

Table 3: Avg Ferry Buffer Utilization

	# of sessions using Ferry	Average Ferry buffer utilization
OSFRS	109	93%
BERS	57	81%

The Average Path Delay for OSFRS and BERS scheme is shown in Table 4. The path delay does not include the waiting time (the time between the arrival of a request and its acceptance by the ferry). This waiting time is the same for both OSFRS and BERS scheme. The average path delay using BERS is 27% smaller than that using OSFRS.

Table 4: Avg Waiting Time/Path Delay

	Average waiting time(s)	Average path delay per session(s)
OSFRS	133.1	140.2
BERS	133.1	102.7

In our second set of experiments, we enhance our simulator to include finite packet arrival process. Each admitted session will have packets arriving periodically at a certain rate. We repeat

the same experiment to compare between the two routing schemes, namely the BERS and OSFRS. Figure 6 and Figure 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.

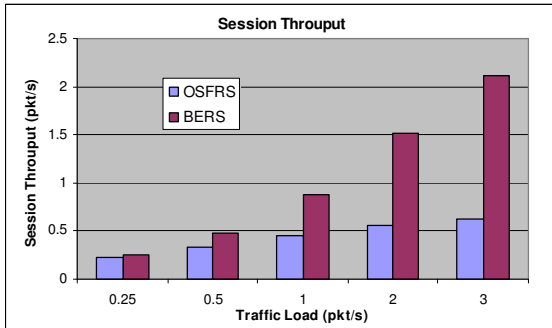


Figure 6: Avg Session Throughput vs. Traffic Load

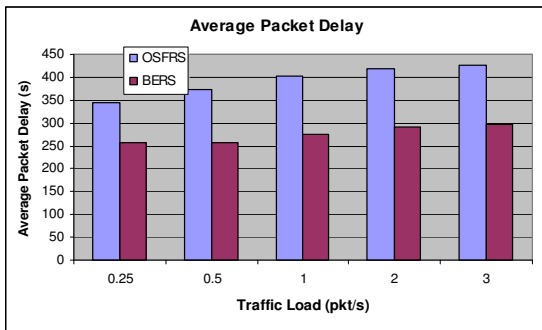


Figure 7: Avg Packet Delay vs. Traffic Load

6 Conclusions

In this paper, we identify 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 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 also 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.

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 is sponsored by Defense Advanced Research Projects Agency (DARPA) under contract 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.

7. References

- [1] C. Perkins, E. Royer, "Ad-hoc On Demand Distance Vector Routing", Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pp 90-100, Feb, 1999.
- [2] D. Johnson, D. Maltz, "Dynamic source routing in adhoc wireless networks", Proceedings of Sigcomm, Aug, 1996
- [3] A. Vahdat, D. Becker, "Epidemic routing for partially connected adhoc networks", Duke University, Tech Report, CS-200006, 2000.
- [4] W. Zhao, M. Ammar, "Message ferrying: proactive routing in highly partitioned wireless ad-hoc networks", IEEE Workhop on future trends in distributed computing systems, May, 2003
- [5] A. Pentland, R. Fletcher, A. Hasson, "DakNet: Rethinking connectivity in developing nations", IEEE Computer, 37(1): 78-83, Jan, 2004.
- [6] R. C. Shah, S. Jain, W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," IEEE international workshop on sensor network protocols and applications (SNPA), Anchorage, May, 2003
- [7] R. Viswanathan, J. Li, M. Chuah, "Message ferrying for constrained scenarios", poster, WoWMoM 2003
- [8] X. Huang, B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation", Proceedings of ACM Mobihoc Oct, 2001.
- [9] H. Luo, J. Cheng, S. Lu, "Self-coordinating localized fair queueing in wireless ad-hoc networks", IEEE transaction on mobile computing, 3(1), Jan-March, 2004
- [10] L. Tassioulas, S. sarkar, "Max-min fair scheduling in wireless networks", Proceedings of IEEE Infocom, 2002, June 2002.
- [11] Y. Yi, S. Shakkottai "Hop-by-hop congestion control over a wireless multihop network", Proceedings of IEEE Infocom, March, 2004.
- [12] K. Xu etc, "TCP behavior across multihop wireless networks and the wired internet", Proceedings of ACM International Workshop on Wireless Mobile Multimedia, 2002