

# Delay Tolerant Mobility Aware Routing/Mobility Dissemination Protocol for the Airborne Network

Kevin C. Lee<sup>†</sup>, Adam Piechowicz<sup>‡</sup>, Mario Gerla<sup>†</sup>, Abhishek Tiwari\*, Anurag Ganguli\*, David Krzysiak<sup>‡</sup>

\*UtopiaCompression Corporation

11150 West Olympic Boulevard, Suite # 680, Los Angeles, CA 90064

{abhishek, anurag}@utopiacompression.com (**Corresponding Authors**)

**Abstract**—Despite “airborne network (AN) topology design” and careful planning of AN trajectories, unexpected disruptions (from hardware failures to changes in mission requirements and hostile attacks) may cause nodes *not* to connect to one another directly or indirectly either because they are out of one another’s range or because nodes do not meet one another according to their preplanned trajectories. Since an end-to-end path within the AN is not always guaranteed, packets have to be delivered in a delay-tolerant fashion, namely, some intermediate nodes will need to buffer packets during times of disconnectivity. In our earlier work we developed Mobility Aware Routing Protocol and Mobility Dissemination Protocol (MARP/MDP) that used preplanned trajectories of airborne nodes to make intelligent routing decisions preemptively. In this paper we present a delay-tolerant strategy (MARP/MDP+DTN) to predict the minimum end-to-end delay and obtain the corresponding path. In addition, MARP/MDP+DTN accounts for local queueing (MARP/MDP+DTN+QC) to minimize congestion and further improves end-to-end delay with the *positive* side effect of load-balancing. Simulation results have shown an improvement of 52% in packet delivery ratio in MARP/MDP+DTN. MARP/MDP+DTN+QC also exhibits extremely short latency, about 90% reduction from MARP/MDP+DTN in highly congested network. Moreover, MARP+DTN+QC balances local traffic 67% better than MARP+DTN in high traffic load scenarios.

## I. INTRODUCTION

The Airborne Network (AN) will form an essential part of the Global Information Grid in the future, thus providing information and decision superiority to US armed forces. AN, an enabling technology for Network Centric Warfare, differs from the terrestrial mobile ad-hoc networks (MANETs) and the wire-line Internet, both in terms of network capability and underlying assumptions. For one, the AN backbone nodes which provide

almost-permanent connectivity are envisioned to fly in pre-planned orbits. Moreover, the transmitter power of these nodes are significantly higher than battery operated MANET nodes.

Mobility Aware Routing Protocol/Mobility Dissemination Protocol (MARP/MDP) protocol suite [4] is designed specifically for the airborne networks. MARP/MDP routes traffic based on the knowledge of planned nodes’ trajectories with respect to time and makes preemptive decisions to minimize packet losses due to link failure and discover better routes. For every topology entry, MARP computes Dijktras shortest paths between itself and every receiver and stores this information in a T-time routing table, whose routing entries change as a function of time. In MARP, therefore, a successful route lookup is the output of the current time instant and the destination’s address. Since it is possible that some airborne platforms may not adhere to the pre-decided flight plans or links go down unexpectedly because of weather conditions and link quality, MDP is used to inform all network nodes of any deviation from the preplanned behavior so that their T-time routing table can be updated accordingly.

In spite of “airborne network (AN) topology design” and careful planning of AN trajectories by MARP/MDP, unexpected disruptions (from hardware failures to changes in mission requirements and hostile attacks) may cause nodes not to connect to one another at all directly or indirectly either because they are out of one another’s range or because nodes do not meet one another according to their preplanned trajectories. Since an end-to-end path within the AN is not always guaranteed, packets have to be delivered in a delay-tolerant fashion, namely, some intermediate nodes will need to buffer packets during times of disconnectivity. This delay tolerant operation assumes that the applications routed on these paths tolerate the associated delay. With the preplanned trajectories of nodes in an

<sup>†</sup> Department of Computer Science, University of California, Los Angeles, Los Angeles, CA 90095

<sup>‡</sup> Air Force Research Lab, Rome Research Site/RIGC - Networking Technology

airborne network (AN), a delay-tolerant strategy can be obtained to predict the minimum delay and to quantify resource (buffer) usage for any set of “failures” or of “known departures” from the initial schedules. In the face of an Unattended Airborne Vehicle (UAV) failure, this prediction is used to determine whether the AN satisfies the application requirement. Efficient resource usage is also important to ensure that there is enough buffer space for data communication on ever-improving link technology. By integrating knowledge of trajectories of the AN backbone, we can quantify buffer space, route more intelligently, and thereby reduce packet failure rate and efficiently manage network resources.

In this paper, we propose MARP/MDP+DTN that enables AN routing in the presence of intermittent connectivity. The contributions of the paper are two-fold. First, MARP/MDP+DTN is proposed to provide the best next forwarding hop regardless of current network condition. Unlike MARP/MDP, MARP/MDP+DTN tolerates delay and therefore provide better end-to-end packet delivery. Second, MARP/MDP+DTN is traffic-aware. DTN feature comes at a cost of queueing. To minimize delay that packets experience in queueing, MARP/MDP+DTN is made sensitive to queueing. It routes traffic around congested network region.

The remaining of the paper is organized as follows: Section II provides our delay tolerant strategy which exploits the AN node preplanned trajectories. Section III considers queueing condition at each node to further optimize MARP/MDP+DTN by reducing end-to-end delay. Section IV evaluates MARP/MDP+DTN with pure MARP/MDP in different connectivity scenarios and different queueing loads. Section V concludes the paper and presents future plan for MARP/MDP+DTN.

## II. DTN WITH MODIFIED DIJKSTRA ALGORITHM

The AN belongs to the forwarding-based DTN approach since the preplanned trajectories are known. Furthermore, we can think of the topology as a time-varying graph [2]. As time progresses, links between nodes either go up or down, forming a particular graph in that time duration. According to [3], a modified Dijkstra’s algorithm with time-varying edge costs can be constructed to predict the minimum delay between nodes and quantify the buffer space at each node. MARP/MDP+DTN uses the modified Dijkstra’s algorithm that accounts for link duration to compute the next forwarding hop which yields minimum end-to-end delay. When there is no end-to-end connectivity, MARP/MDP+DTN would compute the best next forwarding hop to minimize end-to-end

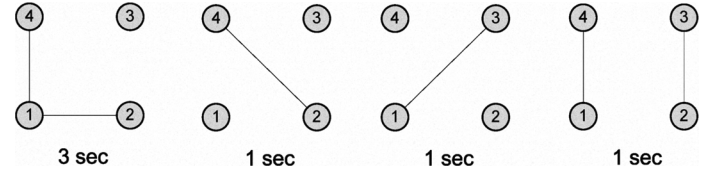


Fig. 1. 4-node time-varying topologies, single-encountering.

delay. Each node is equipped with a queue to store the packets when appropriate.

---

### Algorithm 1 Modified Dijkstra

---

**Require:**  $G = (V, E)$ ,  $s$ ,  $T$ ,  $w(e, t)$

- 1:  $Q \leftarrow V$
- 2:  $L[s] \leftarrow 0$ ,  $L[v] \leftarrow \infty \forall v \in V$  s.t.  $v \neq s$
- 3: **while**  $Q \neq \emptyset$  **do**
- 4:    $u \leftarrow x \in Q$  s.t.  $L[x] = \min_{y \in Q} L[y]$
- 5:    $Q \leftarrow Q \setminus \{u\}$
- 6:   **for all**  $e \in E$  s.t.  $e = (u, v)$  **do**
- 7:     **if**  $L[v] > (L[u] + w(e, L[u] + T))$  **then**
- 8:        $L[v] \leftarrow L[u] + w(e, L[u] + T)$
- 9:     **end if**
- 10:   **end for**
- 11: **end while**

---

Initially, a time-varying graph  $G = (V, E)$ , a source node  $s$ , the start time  $T$ , and the cost of an edge at time  $t$ ,  $w(e, t)$  are given. The modified Dijkstra’s algorithm outputs the path with the minimum delay in an array  $L$ .  $L[x]$  indicates the cost of going to  $x$  from  $s$ . In Line 1, a set  $Q$  that contains all the vertices in  $V$  is created. The cost of source node  $L[s]$ , that is the cost of going to itself, is assigned to 0; the cost of the rest of nodes is assigned to infinity in Line 2. From Line 3 to 11, there is a while loop that terminates when  $Q$  becomes the empty set. While  $Q$  is not empty, a vertex  $u$  that has the minimum cost is picked from  $Q$  in Line 4.  $u$  is then removed from the set  $Q$  in Line 5. For a vertex  $v$  that forms an edge with  $u$  in Line 6, the algorithm checks if the current cost of  $v$ ,  $L[v]$ , is greater than the sum of the cost of going to  $u$  and the cost of going from  $u$  to  $v$ , given that the cost of  $u$ ,  $L[u]$ , and the start time  $T$  in Line 7. If it is,  $L[v]$  is assigned to the sum of the cost of going to  $u$  and the cost of going from  $u$  to  $v$  at time  $(L[u] + T)$  in Line 8. When the algorithm finishes, the array  $L$  will have the minimum delay from  $s$  to each node in the network.

Figure 1 shows an example of 4-node time-varying topologies where the nodes in the time-varying graphs are UAVs flying in their orbit. Links between UAVs change when they fly away from each other in their

orbits. In this particular example, Node 1 is connected to Node 2 and 4 for 3 seconds. Node 2 and Node 4 are connected for the next second. Node 1 and Node 3 are connected the next second. Finally, Node 1 is connected to Node 4 and Node 2 to Node 3 for one second.

Table I shows modified Dijkstra's computation of minimum delay ( $L$ ) from Node 1 to the other three nodes and their predecessor. Each row represents one while loop run in Line 3 of the modified Dijkstra's algorithm. The last row is node's minimum delay and its predecessor. For example, it takes Node 1 a minimum of 5 seconds to reach Node 3, whose previous hop is Node 1.

Note that when  $Q$  contains  $\{3, 4\}$ ,  $L[3] < 1 + w(e(2, 3), 1) = 1 + (4 + 1)$ .  $w(e(2, 3), 1)$  indicates the cost to reach Node 3 from Node 2 at time 1.  $w(e(2, 3), 1) = 4 + 1$  because it takes 4 seconds for the edge between Node 2 and Node 3 to become available since time 1 and it takes 1 second for the packet to be sent from Node 2 to Node 3. This example, based on nodes' predecessor, shows that Node 1 can reach Node 2 starting at 0 second. Node 1 can reach Node 3 starting at 4th second. Node 1 can reach Node 4 starting at 0 second.

When airborne nodes deviate from their planned trajectory in the event of adverse weather conditions or emergency, Mobility Dissemination Protocol (MDP) Discrepancy packets are flooded in the network to update each node's routing table accordingly (See [4] for detail). As we have seen before, MARP predicted topology is a piecewise constant function of time. MDP link state updates do not contain time to live, therefore MARP/MDP+DTN assumes that a Discrepancy indicated by MDP is only valid till the future time instant when the predicted topology changes. Then the modified Dijkstra's algorithm is called on the updated time dependent topology to yield the best route to the destination.

### III. LOCAL QUEUING AWARE SCHEDULING

When disconnectivity increases, there will be large queuing delay. To minimize it, we account for queuing at all edges outgoing from the current node to route around congestion at the first hop. The path traversed thus is sensitive to the queuing present at all edges in the path. Because queuing information is local, we recompute the route at every hop (per-hop routing).

During each periodic beaconing, each MARP/MDP node will forward its queue size to its neighbors. Note that receiving the queue size of its neighbors is just as reliable as receiving hello messages. Unless the neighbor

is out of the node's radio range, a hello message along with the queue size will be received the next beacon period. The modified Dijkstra will select the next hop to each destination with the minimum delay based on the next hop's queue size. " $L[u] + w(e, L[u] + T)$ " in Lines 7 and 8 of the modified Dijkstra will reflect the current queue size of  $u$ ; specifically,  $w(e, L[u] + T)$  will incorporate the cost of sending packets already in the queue at  $u$ , in addition to the one packet that is scheduled to be sent from  $u$ . Packets are dequeued on the first-in first-out basis. Although each packet in the queue may be going through a different neighbor towards a different destination, we assume that they are going to the same destination  $v$  as the last packet. The calculation gives a conservative and upperbound cost to  $v$  through  $u$ . The cost estimation based on the queue will avoid nodes with large queue in general as each packet is subject to the same delay (i.e., the period when  $(u, v)$  is up is the same). The slight difference in queue size between two nodes may favor the one smaller even though packets in the larger queue are dequeued faster because many of them are going through links that come up earlier. We show the computation of the link cost in *single* and *multiple encountering* in the following examples.

#### A. Single Encountering

The example follows the same time-varying graph as Figure 1. However, assume that there are two messages in Node 2's queue for the sake of illustration of how MARP/MDP+DTN with queueing awareness enabled would react differently to traffic congestion. Furthermore, assume that the transmission rate is 1 msg/sec<sup>1</sup> and propagation delay is negligible. Table II shows the table of modified Dijkstra's computation from Node 1 to the other three nodes. At the last row, the minimum delay along with the node's predecessor is computed.

Note that when  $Q$  contains  $\{3, 4\}$ ,  $L[3]$  is compared with  $1 + w(e(2, 3), 1) = 1 + 5 + 6 + 6$ .  $w(e(2, 3), 1) = 5 + 6 + 6$  because it takes 5 more seconds to dequeue the first packet, 6 seconds to dequeue the second packet, and another 6 seconds to dequeue the last packet.  $L[4]$  is compared with  $1 + w(e(2, 4), 1) = 1 + 3 + 6 + 6$ .  $w(e(2, 4), 1) = 3 + 6 + 6$  because it takes 3 more seconds before the first packet is dequeued, 6 seconds before the second packet is dequeued, and 6 seconds before the last packet is dequeued. As described above, the packets in Node 2's queue will have the same destination as the last

<sup>1</sup>We use transmission rate of 1 msg/sec for simplicity of illustrating our example. The link capacity can be higher, but the computation of cost will still be the same.

Q	L[1], Predecessor(1)	L[2], Predecessor(2)	L[3], Predecessor(3)	L[4], Predecessor(4)
{1, 2, 3, 4}	0,1	$\infty$	$\infty$	$\infty$
{2, 3, 4}	0,1	1,1	5,1	1,1
{3, 4}	0,1	1,1	$5 < 1 + w(e(2,3), 1) = 1 + (4 + 1) = 6$	$1 < 1 + w(e(2,4), 1) = 1 + (2 + 1) = 4$
{3}	0,1	1,1	5,1	1,1
$\emptyset$	0,1	1,1	5,1	1,1

TABLE I  
ROUTING TABLE OF NODE 1 TO THE OTHER NODES IN ONE CYCLE.

Q	L[1], Predecessor(1)	L[2], Predecessor(2)	L[3], Predecessor(3)	L[4], Predecessor(4)
{1, 2, 3, 4}	0,1	$\infty$	$\infty$	$\infty$
{2, 3, 4}	0,1	1,1	5,1	1,1
{3, 4}	0,1	1,1	$5 < 1 + w(e(2,3), 1) = 1 + 5 + 6 + 6 = 18$	$1 < 1 + w(e(2,4), 1) = 1 + 3 + 6 + 6 = 16$
{3}	0,1	1,1	5,1	1,1
$\emptyset$	0,1	1,1	5,1	1,1

TABLE II  
ROUTING TABLE OF NODE 1 TO THE OTHER NODES, WITH QUEUEING CONSIDERATIONS IN A SINGLE-ENCOUNTERING SCENARIO.

Q	L(1), Predecessor(1)	L(2), Predecessor(2)	L(3), Predecessor(3)	L(4), Predecessor(4)
{1, 2, 3, 4}	0,1	$\infty$	$\infty$	$\infty$
{2, 3, 4}	0,1	1,1	5,1	1,1
{3, 4}	0,1	1,1	$5 < 1 + w(e(2,3), 1) = 1 + 3 + 2 + 4 = 10$	$1 < 1 + w(e(2,4), 1) = 1 + 3 + 6 + 6 = 16$
{3}	0,1	1,1	5,1	1,1
$\emptyset$	0,1	1,1	5,1	1,1

TABLE III  
ROUTING TABLE OF NODE 1 TO THE OTHER NODES, WITH QUEUEING CONSIDERATIONS IN A MULTIPLE-ENCOUNTERING SCENARIO.

packet that is just inserted into the queue. For the packet that is going to Node 3, the first two packets queued are going to Node 3 as well. Similarly, for the packet that is going to Node 4, the first two packets queued are going to Node 4 as well.

Note that in this time-varying graph, Node 2 only meets Node 3 and Node 4 once, at 6th second and 4th second, respectively. We characterize a node as *single encountering* if it only meets another node in the network once in the period of the time-varying graph. In this example of the time-varying graph with a period of 6 seconds, Node 2 is single encountering with Node 3 and Node 4 as Node 2 only meets with them *once* during a period. We will discuss the more general case where a node meets another node *more than once* in the next section.

Now the cost of  $L[v]$  should be compared with  $L[u] + w(e, L[u] + T)$ , where  $w(e, L[u] + T) = \text{time}(1, e, L[u] + T) + P \times (Qsize - 1)$ .  $\text{time}(x, e, L[u] + T)$  is the function that outputs the time it takes to deliver  $x$  number of packets by the edge  $e$  formed by Node  $u$  and Node  $v$  since time  $L[u] + T$ .  $P$  is the period of the time-varying graph.  $Qsize$  is the size of the queue at

Node  $u$ , including the packet going to Node  $v$  through Node  $u$ .  $P \times (Qsize - 1)$  is the time to dequeue the rest of  $(Qsize - 1)$  packets. In the example above,  $w(e, L[u] + T) = 5 + 6 \times 2 = 17$  is consistent with what is shown Table II<sup>2</sup>.

### B. Multiple Encountering

We examine node's meeting another node more than once in one period of a time-varying graph. In other words, a node meets another node if and only if there is communication between them. For example, in Figure 2, Node 2 meets Node 3 *more than once* at the 4th second and 6th second. We characterize Node 2 as one that is *multiple encountering* with Node 3 during one period of the time-varying graph. If two nodes encounter each other multiple times in a period, certainly that the time it takes for all the packets queued up in one node to be delivered to another is less than if the two nodes encounter only encounter each other once in a period. Since

<sup>2</sup>The table for  $L[3]$  entry when  $Q = \{3, 4\}$  is comparing  $5$  with  $1 + w(e(2,3), 1) = 1 + 17 = 18$ . The 1 that is added to 17 is the shortest time to reach Node 2 plus the shortest time then from Node 2 to Node 3.

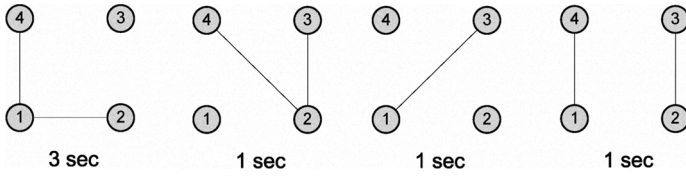


Fig. 2. 4-node time-varying topologies, multiple-encountering.

for each encounter they have, packets can be offloaded from one node to the other. Note that in a period, there can be both single and multiple encountering nodes. The characterization depends on the node under examination and on the neighbor with which it encounters. The example illustrates a general case of computing the cost of “ $L[u] + w(e, L[u] + T)$ .” Once again, we assume that there are two messages in Node 2’s queue, the transmission rate is 1 msg/sec, and propagation delay is negligible.

When  $Q$  contains  $\{3, 4\}$ ,  $L[3]$  is compared to  $1 + w(e(2, 3), 1) = 1 + 3 + 2 + 4$ .  $w(e(2, 3), 1) = 3 + 2 + 4$  because it takes 3 more seconds to dequeue the first packet, 2 more seconds to dequeue the second packet, and another 4 seconds to dequeue the last packet. Since Node 2 encounters Node 3 more than once in one time-varying graph period, the first two packets are able to be dequeued in one period. Thus, the number of packets that are dequeued in one period depends on the number of times a node meets the other node.

#### IV. EVALUATION

We implemented and integrated delay tolerant capability to MARP/MDP routing protocol on Qualnet 4.5 [1]. Local queuing aware scheduling was also implemented. The following subsections provide the comparison between MARP/MDP and MARP/MDP+DTN and the comparison between MARP/MDP+DTN and MARP/MDP+DTN+QC<sup>3</sup>, respectively.

##### A. MARP/MDP vs. MARP/MDP+DTN

In the first experiment, we evaluate the performance of MARP/MDP and MARP/MDP+DTN on a realistic scenario. This scenario has a fairly complex topology with nodes on varying paths and with intermittent connectivity. A network flow was configured to source at node GlobalHawk and sink at node AWACs 2. Connectivity between these nodes is particularly sporadic and a complete end-to-end path is rarely available. This configuration provides an excellent opportunity to show the improvement possible with MARP/MDP+DTN over

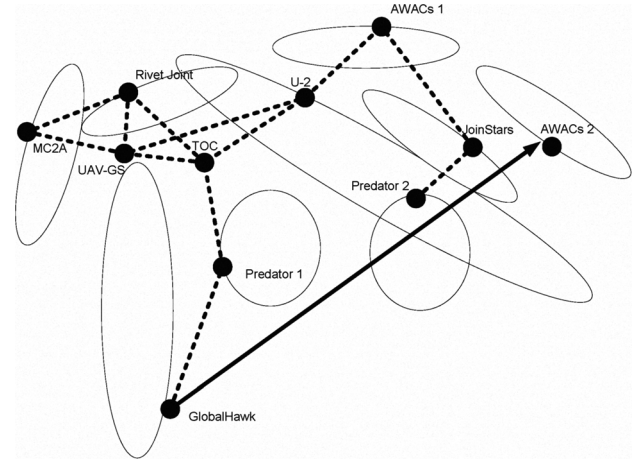


Fig. 3. A realistic representative airborne network scenario.

a non-delay-tolerant solution. Figure 3 shows the topology where the oval shapes are the routes of the nodes, the thick arrow shows the desired network flow, and the dotted lines are current available connections, which will change during the simulation as the nodes move.

The simulation was run over three experimental variables — radio range, delay tolerance, and flow volume. Varying the radio range will change the connectivity of the network and the available paths. Varying the delay tolerance will show the difference in performance due to protocol optimization. Flow volume is used in order to illustrate some behaviors of the network which would be difficult to grasp in only one fixed trial. A low volume flow, consisting of 300 byte packets sent one every ten seconds for the duration of the simulation will be used to show the overall route availability between GlobalHawk and AWACs 2, while a higher volume flow of one packet every 0.75 seconds is used to show how a more realistic flow would behave. In simulation the volume of the high-traffic flow was limited by available simulation hardware, since queues must be maintained for each of the simulated nodes and storing packets for delay-tolerant transmission has large memory requirements.

As shown in Table IV, delay-tolerance represents a large increase in delivery percentage in all scenarios. The improvement is as high as 52% comparing column 1 and 2. 100% delivery cannot be achieved due to finite queue space for holding delayed messages, signal attenuation leading to packet errors or drops, and the network topology not presenting any route to the destination over the final few hundred seconds (as the destination is fully disconnected). The trade-off is that holding the messages in queue causes a large latency.

By fixing the range and type (e.g., look at column 1 and 3), one observes that low flow has higher delivery

<sup>3</sup>QC stands for Queuing Compensation.

TABLE IV  
PACKET DELIVERY AND LATENCY AS A RESULT OF RADIO RANGE, DELAY-TOLERANCE, AND FLOW.

Column	Range (km)	Type	Flow	Packet Delivery (%)	Latency (s)
1	250	Delay-tolerant	Low	58.33%	975.56
2	250	Non-delay-tolerant	Low	6.67%	0.23
3	250	Delay-tolerant	High	33.31%	1498.58
4	250	Non-delay-tolerant	High	6.61%	0.23
5	300	Delay-tolerant	Low	79.33%	425.82
6	300	Non-delay-tolerant	Low	37.67%	0.17
7	300	Delay-tolerant	High	65.39%	451.42
8	300	Non-delay-tolerant	High	34.93%	0.14

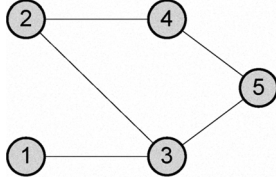


Fig. 4. Network topology to evaluate queuing awareness.

ratio and lower latency than high flow does. This is because the high volume of traffic at each node causes the queue to be filled up that further incoming packets have to be dropped. Since the queue at most of these nodes is saturated, it takes longer average time for packets to arrive at the destination. Furthermore, by fixing the type and flow (e.g., look at column 1 and 5), one also observes that high radio range has higher delivery ratio and lower latency than the low radio range does. This is because the high radio range provides fewer hops to the latency. This consequently improves delivery ratio and end-to-end latency.

The improvement in delivery ratio for MARP/MDP+DTN is due to the buffering of temporarily-undeliverable packets in intermediate nodes in order to deliver them later. When links come up, a burst of DTN traffic occurs as queued packets are advanced towards the source. This effect is visible in huge spikes in bytes delivered in the DTN case when the data is viewed in timeslices based on delivery time. The figure is not shown for the sake of space.

#### B. MARP/MDP+DTN vs. MARP/MDP+DTN+QC

In the second experiment, we evaluate the performance gain from enabling MARP/MDP+DTN with local neighbor queuing awareness, queuing compensation (QC). The topology is set up as shown in Figure 4 where two separate CBR traffic flows originate from Node 1 and Node 2, respectively, to the sink Node 5. Furthermore, the link between Node 4 and 5 goes down

for a short duration<sup>4</sup> periodically. MARP/MDP+DTN's and MARP/MDP+DTN+QC's PDRs and latencies are compared with different transmission rates, ranging from 2.048 Mbps to 1024 kbps. For each transmission rate, 20 simulation runs were conducted to obtain the average with 95% confidence interval.

Figure 5(a) shows the aggregate PDR from the two flows in decreasing transmission rates for both MARP/MDP+DTN and MARP/MDP+DTN+QC. When the transmission rates from both Node 1 and 2 are high, the PDR for MARP/MDP+DTN is close to 0%. Only until the transmission rates begin to drop to 204.8 kbps that packets start to be received at Node 5. Figure 6, which shows the difference of packets forwarded from Node 3 and Node 4, offers an explanation to why MARP/MDP+DTN is unable to handle high traffic volume. According to the figure, both flows from Node 1 and 2 forward through Node 3 heavily in MARP/MDP+DTN. In MARP/MDP+DTN+QC, however, Node 2 is able to know the local queuing conditions of Node 3 and Node 4. Knowing that packets are heavily queued up at Node 3, Node 2 then diverts its traffic to Node 4, thus increasing the overall packet delivery ratio. Due to space constraint, results for individual PDR for both traffic flows are not shown. However, MARP/MDP+DTN+QC does provide higher PDR for both traffic flows for every transmission rate.

Figure 5(b) shows the latency from the two flows in decreasing transmission rates for both MARP/MDP+DTN and MARP/MDP+DTN+QC. The results clearly show the need for local queuing awareness. In MARP/MDP+DTN, since Node 2 keeps forwarding its packets to Node 3, at 2.048 mbps transmission rate, Node 3's queue is constantly full. Incoming packets are dropped at the fully saturated queue. This results in the low packet delivery ratio. As the transmission

<sup>4</sup>The setup is to demonstrate that despite frequent disconnectivity between Node 4 and 5, MARP/MDP+DTN will still utilize such a link to load balance the network and at the same time minimizes latency.



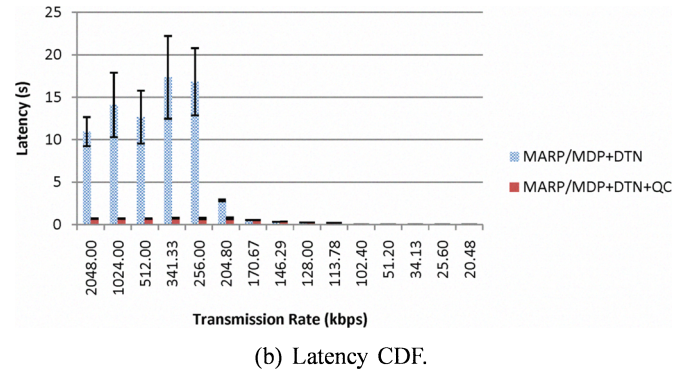
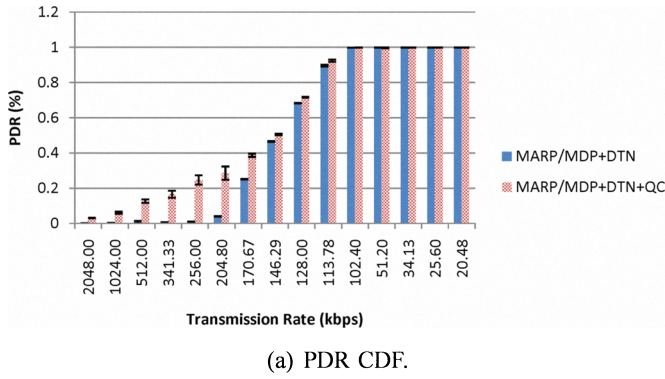


Fig. 5. The cumulative distribution functions of both PDR and Latency.

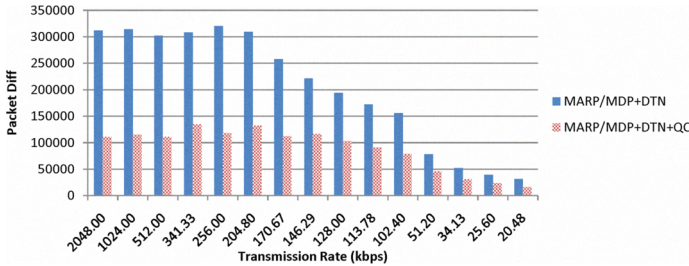


Fig. 6. The difference in forwarded packets between MARP/MDP+DTN and MARP/MDP+DTN+QC.

rate decreases, Node 3's queue is still constantly full; however, incoming packets are coming at the rate below the packet servicing rate that they are admitted into the queue successfully. This explains the higher PDR as the transmission rate decreases. Since these received packets experience long wait time, they attribute to the higher average latency. Thus, there is an increasing trend of average latency as the rate decreases up to 341.33 kbps. From 341.33 kbps onward, the latency starts to drop, indicating that pumping packets out at a rate of 341.33 kbps *exactly* saturate Node 3's queue without any packet drop.

Since MARP/MDP+DTN+QC is able to divert traffic from Node 3 to Node 4 to achieve load balancing, Node 3's queue is never full. In fact, the queue is always kept at a low level that the average latency is never above 1 second. Figure 6 shows the difference in packets between packets forwarded by Node 3 and packets forwarded by Node 4 for both MARP/MDP+DTN and MARP/MDP+DTN+QC. MARP/MDP+DTN's higher difference than MARP/MDP+DTN+QC's indicates the wide disparity in forwarded packets from Node 3 and Node 4. While in MARP/MDP+DTN, a lot more packets are forwarded to Node 3 than Node 4, in MARP/MDP+DTN+QC, packets are evenly distributed

between Node 3 and Node 4, showing further evidence of load balancing provided by queuing compensation.

## V. CONCLUSION

The paper has enabled MARP/MDP with delay tolerant capability, plus local queuing awareness. Results have shown the benefits of delay tolerance in improving packet delivery ratio. By being aware of local queuing conditions, nodes are able to route around traffic congestion and minimize their traffic delay. As future work our team will tune parameters/routing metrics of MARP/MDP+DTN+QC protocol in accordance with flight and link data obtained from real flight tests like Capstone II.

## ACKNOWLEDGEMENT

The work presented in this paper was sponsored in part by the Air Force under the Small Business Innovation Research (SBIR) Phase II program, Air Force Research Laboratory (AFRL) at Rome Lab contract number FA8750-08-C-0127. This work was in collaboration with Professor Mario Gerla and his team under a subcontract to GPC. The authors would also like to thank MITRE Corporation, Bedford, MA for their guidance.

## REFERENCES

- [1] Qualnet Network Simulator. <http://www.vehicle-infrastructure.org>.
- [2] Scott Burleigh. Dynamic routing for delay-tolerant networking in space flight operations. In *SpaceOps 2008, AIAA 2008-3406*, May 2008.
- [3] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145–158, New York, NY, USA, 2004. ACM.
- [4] A. Tiwari, A. Ganguli, A. Sampath, D.S. Anderson, B.-h. Shen, N. Krishnamurthi, J. Yadegar, M. Gerla, and D. Krzysiak. Mobility aware routing for the airborne network backbone. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov. 2008.