

## Maximizing Throughput of UAV-Relaying Networks with the Load-Carry-and-Deliver Paradigm

Chen-Mou Cheng    Pai-Hsiang Hsiao    H. T. Kung    Dario Vlah  
{doug, shawn, htk, dario}@eecs.harvard.edu  
School of Engineering and Applied Sciences  
Harvard University

### ABSTRACT

*We consider the task of using one or more Unmanned Aerial Vehicles (UAVs) to relay messages between two distant ground nodes. For delay-tolerant applications like latency-insensitive bulk data transfer, we seek to maximize throughput by having a UAV load from a source ground node, carry the data while flying to the destination, and finally deliver the data to a destination ground node. We term this the "load-carry-and-deliver" (LCAD) paradigm and compare it against the conventional multi-hop, store-and-forward paradigm. We identify and analyze several of the most important factors in constructing a throughput-maximizing framework subject to constraints on both application allowable delay and UAV maneuverability. We report performance measurement results for IEEE 802.11g devices in three flight tests, based on which we derive a statistical model for predicting throughput performance for LCAD. Due to the nature of commercial off-the-shelf systems, this methodology is of essential importance for allowing better flight-path design to achieve high throughput.*

### I. INTRODUCTION

The low cost and high performance of commercial, off-the-shelf (COTS) wireless equipment, such as the IEEE 802.11 wireless LAN ("Wi-Fi"), make it now practical to use in small, low-altitude Unmanned Aerial Vehicles (UAVs). This new capability has enabled many applications in UAV networking. For example, UAVs can act as relays between ground stations that could not otherwise communicate due to distance or obstructed line of sight. Multiple UAVs could simultaneously detect, record and track wildfires. Last but not least, UAV networks can be deployed on demand to create an instant communication infrastructure. This can be useful in emergency situations, such as following a hurricane, or even in everyday scenarios, such as during a major sporting event.

There are uncertainties with these UAV-based networks that go beyond the usual capacity and quality-of-service concerns found in wireless mobile networks. There are UAV-specific issues, such as rapid changes in link quality due to UAV's

banking and traveling at relatively high speeds, as well as the relatively low tolerance of the 802.11 receivers to radio interference [1][2].

Here, we consider a new type of networking paradigm, called "load-carry-and-deliver" (LCAD), that is specifically tailored to the task of using one or more UAVs in relaying messages from a source to a destination ground node. Under LCAD, a UAV will load data from the source ground node, carry it while flying towards the destination, and finally deliver it to the destination ground node.

LCAD is similar to previously proposed data ferrying schemes. However, in this paper we consider for the first time the throughput maximization of basic loading and delivery steps using realistic link models and validate with field experiments. Most previous works use simplified communication models, such as the ideal unit-disk network model, that are considered to be inadequate for practical COTS radios [3]. Our results are complementary to the works on message ferrying [4][5][6] by Zhao et al., which examine how the non-randomness of mobility, or even controlled mobility, can improve network-wide message delivery and energy consumption under simplistic link models.

A second unique result of our work is the link models we derived from empirical measurement data, which will allow us to construct and characterize actual flight paths that the UAVs can execute. This is unlike previous works that only consider finding an "abstract path," that is, a sequence of nodes to visit or a representative path defined by simple way points. For example, the latter is the approach that Brown et al. [7][8] used for roaming UAVs to deliver packets via controlled mobility.

Although the LCAD paradigm incurs a longer data delivery delay than conventional store-and-forward, LCAD does have a number of important advantages. First, LCAD can achieve high throughput performance by ensuring that UAV's communication with the source and destination ground nodes is free of interference from other nodes in the same networking system. In contrast, other 802.11-based multi-hop networks usually suffer from severe interference problems [1][2]. Second, LCAD can scale its throughput by using multiple relaying UAVs in a pipelined fashion for data delivery, while other approaches often cannot due to interference and medium sharing constraints [9][1]. For these reasons, LCAD is attractive for those delay-tolerant applications that demand high networking bandwidth, such as bulk data transfer.

This work represents a step towards an optimization-oriented design framework for a UAV-assisted relaying net-200

This material is based on research sponsored by Air Force Research Laboratory under agreement numbers FA8750-05-1-0035 and FA8750-06-2-0154, and by the National Science Foundation under grant number #ACI-0330244. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, the National Science Foundation, or the U.S. Government.

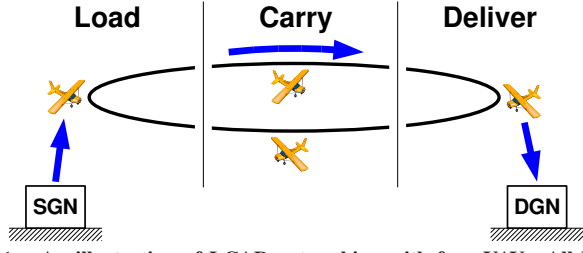


Fig. 1. An illustration of LCAD networking with four UAVs. All UAVs fly along the same oval-shaped path between the source ground node (SGN) and the destination ground node (DGN). The three stages of communication, load, carry, and deliver, are highlighted.

work. We will present analysis on fundamental necessary conditions of optimality and put forward a perspective through which we can design practical protocols and UAV flight paths to achieve such optimality. Furthermore, our work includes protocol provisions specifically designed for efficient loading and delivery of data in practical settings.

The rest of this paper is organized as follows. In Section II, we will provide a detailed description of the LCAD paradigm, along with an analysis leading towards a throughput-maximizing framework based on empirical link performance modeling. In Section III, we will describe the methodology we have used for our UAV flight experiments. In Section IV, we will report the measurement results from these experiments, based on which we will construct a statistical performance model. We will then conclude this paper in Section V.

## II. THROUGHPUT-MAXIMIZING LCAD NETWORKING

In this paper, we focus on the use of IEEE 802.11 radios, and we will assume a fixed data transmission rate in the following analysis. Although 802.11 radios can change the transmission rate dynamically in response to changes in the channel condition, we believe the performance may not be as good in a rapidly changing environment due to lack of efficient channel information feedback mechanisms. For example, several of the current rate adaptation algorithms select transmission rate by sending packets at different rates to probe the channel and observing their packet error rates [10]. In a rapidly changing environment such as a UAV network, the observation is likely to be noisy and will soon become obsolete, resulting in poor rate adaption performance. For similar reasons, we also exclude the use of transmission power adaption techniques and assume the use of a fixed transmission power.

LCAD is a communication paradigm for delivering a stream of packets from one fixed ground node to another. Under LCAD, a mobile UAV node buffers packets received from a source ground node (SGN), carries them while flying towards the destination, and finally delivers them to the destination ground node (DGN). Figure 1 illustrates the working of the LCAD paradigm.

Communication in LCAD networking is divided into three stages. The first is the *load* stage, where SGN transmits to UAV. The second is the *carry* stage, during which no one transmits. The third is the *deliver* stage, in which UAV

transmits to DGN those packets it received from SGN. The important design decisions in LCAD networking include the time allocation for each stage, as well as the design of the UAV trajectory.

When a UAV cyclically flies along the same path, such as the oval-shaped flight path shown in Figure 1, the achieved long-term throughput  $\mathbb{T}$  can be expressed as  $Pkt_{dld}/T_{cycle}$ , where  $Pkt_{dld}$  is the total number of packets delivered to DGN by UAV in one cycle, and  $T_{cycle}$  is the flight time for one cycle.

Let us decompose  $T_{cycle}$  into  $T_{load}$ ,  $T_{carry}$ , and  $T_{deliver}$ , each denoting the time that UAV spends in the respective LCAD stage. For simplicity,  $T_{carry}$  includes the time on the return leg from DGN to SGN. When necessary, we will use  $T_{idle}$  to refer to the time spent in this return leg.

A first necessary condition for achieving maximal throughput is that  $T_{carry}$  must include *only* the time during which the UAV is out of communication range of either SGN or DGN. Otherwise, the system is wasting transmission opportunities. Thus,  $T_{carry}$  is determined by UAV speed, the distance between the two ground nodes, and the communication range of the wireless devices used.

Let  $\mathbb{D}$  be the transit delay budget the application imposes on packet delivery. The transit delay is defined to be the amount of time a packet spent in transit through the network—the time between when the packet is first enqueued at SGN and when the packet is delivered to DGN. The worst case transit delay is  $T_{cycle} + T_{carry} - T_{idle}$ , experienced by the first packet after a load stage, and so the combined time allocated for the load and deliver stage must satisfy the following condition:  $T_{load} + T_{deliver} < \mathbb{D} - 2T_{carry} + T_{idle}$ .

Another necessary condition for achieving maximal throughput is that the load stage does not overrun or underrun the subsequent delivery stage. In either of these cases, the time for the two stages could have been better allocated. Suppose that the single-hop throughput is constant during both  $T_{load}$  and  $T_{deliver}$ . Then to achieve the maximal end-to-end throughput, UAV must spend an equal amount of time in the two stages, i.e.,  $T_{load} = T_{deliver} = (T_{cycle} - T_{carry})/2$ . If the single-hop throughputs,  $R_{load}$  and  $R_{deliver}$ , are unequal but are constant, then  $T_{load} = \frac{R_{deliver}}{R_{load} + R_{deliver}}(T_{cycle} - T_{carry})$  and  $T_{deliver} = \frac{R_{load}}{R_{load} + R_{deliver}}(T_{cycle} - T_{carry})$ .

In reality, the single-hop throughput is likely neither constant nor symmetric. A popular, more realistic path-loss model assumes that the single-hop throughput is a monotonic function of distance—it increases monotonically as distance decreases until it reaches the full rate. In this case, it is straightforward to design a flight path and the associated schedule to maximize the overall throughput. An example is to have the UAV flying around SGN in the smallest-possible circular path during the load stage. Under this model, the single-hop throughput is higher than that when the UAV is moving towards or away from SGN. As a result, the average single-hop throughput for the period of  $T_{load}$  increases with circling time. Consequently, to achieve maximal throughput, the UAV must fly a cycle with maximum  $T_{cycle}$  that satisfies the delay constraint, and allocate the time for  $T_{load}$  and

$T_{deliver}$  according to the average single-hop throughput in the respective stages. One can also trade delay for throughput because  $\mathbb{T}$  increases as  $\mathbb{D}$  increases.

We conclude that a good single-hop throughput model is key for throughput-maximizing LCAD networking because it allows for optimal stage-time allocation and flight-path design. It also provides better insight into the delay-throughput trade-off. However, even a distance-based model described above is too simplistic in the face of real-world factors such as relative angles and polarization between the transmit and receive antennas, perturbations in the UAV positions and attitudes, and Doppler effects associated with UAV speed. We will report our evaluation of more detailed model including some of these factors in Section IV-A.

We may use multiple UAVs in a pipelined fashion, as illustrated in Figure 1. We can schedule these UAVs such that a UAV's load stage always overlaps with the other UAVs' carry or deliver stages, and that its deliver stage always overlaps with the other UAVs' load or carry stages. However, when the number of UAVs exceeds a certain threshold, there could be some complications. For example, two adjacent UAVs could be so close to each other, in either the load or deliver stage, that there could be contention and interference between their transmissions. The use of LCAD with a large number of UAVs merits further investigation.

### III. NETWORK TESTBED AND FLIGHT EXPERIMENTS

In this section, we describe our testbed setup and some flight experiments with LCAD. The testbed was built for the purposes of evaluating LCAD throughput performance and characterizing the wireless links.



Fig. 2. (a) A custom-made dipole antenna installed on a ground node; (b) a dipole antenna installed beneath a UAV wing (inside a cardboard bracket); and (c) a GPS receiver mounted on a UAV wing.

Our networking testbed consisted of a UAV node and two ground nodes—SGN and DGN. These nodes were made up of single-board x86 computers made by Thecus, and were equipped with Wistron CM9 802.11a/b/g adapters (Atheros chipset) with 18dBm transmit power. We used a custom-made 2-dBi dipole antenna on all the nodes. The dipole antenna on the ground node was vertically placed and elevated to about 25 inches above ground (cf. Figure 2 (a)). The dipole antenna on the airplane was vertically placed beneath the wing (cf. Figure 2 (b)). The UAV was built from a Senior Telemaster model airplane kit [11], and the computer equipment was installed inside its body compartment. The two ground nodes were placed on the opposite ends of the runway, separated by 550 yards.

The UAV had an on-board GlobalSat BU-353 GPS receiver (cf. Figure 2 (c)), which provided position information at the

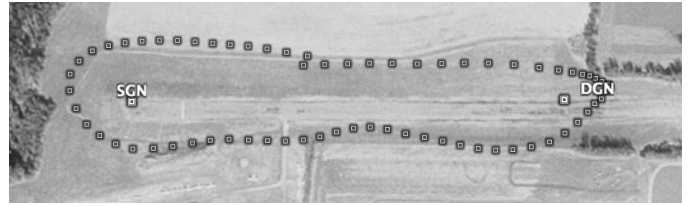


Fig. 3. A sample UAV path in the flight experiments projected onto a U.S. Geological Survey (USGS) satellite map showing the locations of SGN and DGN. The small squares show the per-second positions of the UAV as reported by its GPS. The light-colored horizontal band in the center of the map is an airport runway approximately 25 yards wide and 700 yards long.

resolution of 1 Hz. We had performed a coarse calibration of the GPS, and we found that errors in its reported coordinates were normally within 5 meters. The UAV's GPS trace and the stationary ground nodes' coordinates allowed us to analyze various performance parameters as functions of distance and elevation angle.

The UAV flew in an oval-shaped flight path at an average altitude of 80 yards. An example of such path is shown in Figure 3. The airplane was operated by a human operator on the ground through radio control. Even though the operator tried to follow a predetermined path, there were inevitably noticeable variations in the actual path traversed.

The results reported in Section IV are based on the traces collected in three flight runs that took place on two days one week apart. The first run lasted for 712 seconds on the first day. The second (380 seconds) and the third (637 seconds) runs took place on the second day. During these runs, the airplane completed a total of 20 round trips between the two ground node sites. Although we tried to keep the testbed configuration identical for both days, the weather conditions in terms of wind speed and direction, as well as the airplane operators were different.

#### A. A Lightweight LCAD Protocol

Since the speed of the aircraft and the distance between SGN and DGN were known, we only needed to measure the achieved single-hop throughput for the load and deliver stages in order to compute the overall achieved throughput. Thus, in our experiments, we used an empty carry stage and kept the UAV either in load or deliver stage. To decide the current stage of the UAV node, a daemon process on the UAV node computed the distances from its current GPS coordinates to those of SGN and DGN, respectively. If the UAV was closer to SGN than DGN, it would put itself in the load stage. Otherwise, it would be in the deliver stage. The UAV constantly broadcast beacons at a fixed interval (200 ms) to indicate which one of the two stages it was currently in.

When the UAV is in deliver stage, it sends data packets at full rate to DGN until it enters load stage. If the SGN receives beacons indicating that the UAV is in load stage, it transmits data at full rate to UAV. SGN stops data transmission either when it receives beacons indicating that UAV is no longer in load stage, or after not receiving beacons for 3 consecutive



intervals (600 ms). It is important that SGN stops transmission outside of load stage because its transmission could contend with that of UAV or interfere with reception at DGN. We note that the LCAD protocol here uses a relatively small number of beacon packets so as to minimize the pollution introduced on packet-error measurement and link characterization.

### B. Trace Collection

In our experiments, all nodes used channel 11 of 802.11g, with the link-layer transmission rate fixed at 6 Mbps. All data packets (1,500 bytes, including IP/UDP headers) and beacon packets (64 bytes, including IP/UDP headers) were generated with sequence numbers. In addition, all packets were sent to a network broadcast address, so there was no link-layer retransmission. As a result, we will report the raw packet error rate at the physical layer without ARQ (Automatic Repeat-Request). To shorten the control loop between UAV and SGN, we reduced the `madwifi` driver's transmission queue size to 4 packets and the Linux socket buffer size to 3 packets. These settings reduced queueing in the operating system and avoided excessive delays for both beacon and data packets.

We also collected the timestamp and sequence number of each data or beacon packet sent and received. The timestamp of a sent packet is generated when the socket function call `sendto()` returns, while that of a received packet is generated when `recvfrom()` returns. In addition, UAV's positions were logged along with timestamps for interpolating distance between transmitter and receiver when a particular packet is received.

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we will begin with a summary of LCAD performance measured during 20 complete cycles performed over three runs. A detailed list of the results and time breakdown for each stage is available in Table II of the Appendix. We will then continue with link characterization and construction of statistical packet error models.

We first summarize the throughput utilization for the three runs. The throughput utilization for each cycle is the ratio of the total number of packets delivered to DGN divided by cycle time. The total number of packets delivered is computed by taking the minimum between the number of packets received by UAV and that by DGN—at most that many packets sent by SGN eventually reached DGN. The result shows that the average throughput utilization for the first run is  $0.2283 \pm 0.0369$ ,  $0.2837 \pm 0.1345$  for the second run, and  $0.3176 \pm 0.0278$  for the third run.

From our earlier experience with 802.11g, we learned that the maximum distance between two ground nodes with similar configurations cannot exceed 50 yards if they need to communicate at a reasonably low packet loss rate using 802.11g. In this testbed, it would take additional ten ground nodes to form a relay chain connecting SGN and DGN. Li and et al. reported that the throughput utilization of a 7-node relay chain using 802.11b radios is about 0.25 [1]. The throughput results from the three runs show that LCAD can perform better than the traditional multi-hop ground relay chain.

We observe that the utilization is lower than the average packet error rate (PER) suggests. The average PER for the first run is 0.4223 for load stage and 0.3001 for delivery stage, 0.105 and 0.3389 for the second run, and 0.1147 and 0.2426 for the third run. These error rates should allow an even higher throughput than measured. However, there is an additional loss of efficiency due to buffer underruns or overruns. Figure 4 shows the histogram for the buffer occupancy of the 20 cycles. Negative buffer occupancy indicates that a buffer underrun occurred, and the UAV node could have sent that many more packets if the buffer were not empty. Positive buffer occupancy is the number of packets left in the UAV's queue at the end of a cycle. These packets will be discarded. The average number of packets delivered in a cycle is about 7000 packets, so the buffer occupancy values in the figure indicate a serious imbalance between LCAD stages, which leads to a significant loss in utilization.

The average length of the cycles was  $63 \pm 8.7$  seconds. Within these cycles, the time devoted to load and deliver stages of LCAD was slightly biased toward the load stage. The average ratio of  $T_{load}/T_{deliver}$  was  $1.23 \pm 0.16$ .

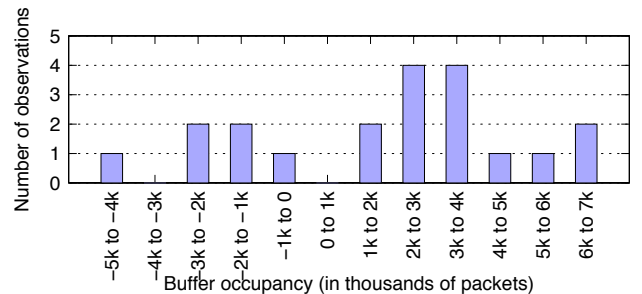


Fig. 4. Buffer occupancy histogram constructed from the 20 flight cycles. Negative values represent an idle, underutilized cycle, while the magnitude indicating the extent of the underutilization.

Figure 5 shows the details of the third cycle in the second run (labelled as “2-3”). The performance results are reported at 1-second intervals for the purpose of investigating correlation at a finer granularity than the numbers reported in Table II. Among the six rows, the fifth row compares efficiency loss and modeled path loss. Efficiency loss incorporates packet errors and halted SGN transmission due to lost beacons. The modeled path loss is the amount of signal attenuation in dB, normalized with respect to the maximal observed attenuation in the experiments. Such attenuation is predicted by a free-space propagation model plus an approximate antenna gain pattern, which we will describe in more detail in the following section.

The fifth row shows the correlation between the efficiency loss and modeled path loss. We do notice that there are discrepancies in some samples. For example, while the modeled path loss does not change as much, PER increases sharply around 370 second in load stage and 402 second in deliver stage. We believe the discrepancies mainly result from an effect of antenna cross-polarization when the UAV banks while making a turn. There are other minor effects, such as velocity and ground reflection, that may contribute to efficiency loss. We will construct a statistical model to better quantify the

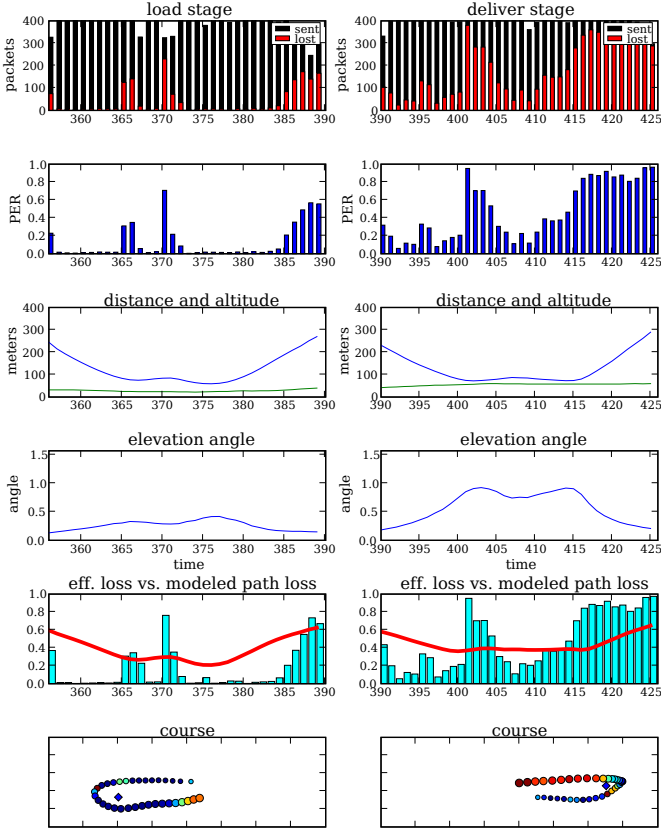


Fig. 5. Details of cycle 2-3. The left column contains results from load stage, whereas the right column contains results from deliver stage. The X-axis of the first five rows represents time offset into the run, and Y-axis shows either the measured performance or the settings of the flight path in the stage. The first row shows the number of packets sent and lost at one second intervals. The second row plots the packet error rate at one second intervals. The third row plots the distance between transmitter and receiver, along with the UAV's altitude above ground over time. The fourth row depicts the UAV's elevation angle relative to the ground node over time. The fifth row compares efficiency loss and modeled path loss. Lastly, the sixth row plots the UAV's trajectory during the stage viewed from the top. The diamond marker indicates the location of the ground node. The circles give the UAV's locations projected onto the ground at the resolution of 1 Hz. The circles are drawn in increasing sizes as time progresses. The color represents PER during that second—red for highest and blue for lowest PER.

correlation in the next subsection.

#### A. Link Characterization

In this subsection, we present an empirical model for link performance prediction. Link performance models are of essential importance to throughput-maximizing flight-path design. In order to help flight-path design, these models can only use information that is available at design time. This may include characteristics of system components such as wireless transceiver and antenna. It may also include trajectory of UAV, which can be obtained at the output of the flight-path design and used in the next design iteration. It can not use, for example, instantaneous signal strength because that information is not available prior to flight. Because of this, our model will predict link performance for the particular system at hand solely based on UAV trajectory information.

From trajectory, we first derive two important factors that

influence link performance. The first one is the distance between the UAV and ground nodes. In many models, distance is the only factor considered. The second factor is the elevation angle  $\phi$  of the UAV, as seen by the ground node, which plays a role because we use vertical dipole antennas. Specifically,  $\phi$  is defined as the angle between the direction of the antenna and the incident direction of the radio waves from the UAV. For example,  $\phi = 0$  when the UAV is directly above the ground node. In this case, the link performance is usually very poor because the UAV and ground nodes are in the antenna “null” of each other [12]. We can confirm this by looking closely into the visualization of one of the cycles in Figure 5.

We will combine the effect of these two factors into a variable called the “modeled path loss.” First, because the UAV usually maintains a line of sight to the ground node that it intends to communicate with, the free-space propagation model should give a good prediction on the propagation loss. On top of that, we will add the signal loss due to the elevation angle  $\phi$  perceived by the ground node, computed as follows. The magnitude of the electrical field due to radiation from a half-wavelength dipole antenna at an elevation angle  $\phi$  is approximately proportional to  $\cos(\frac{\pi}{2} \cos \phi) / \sin \phi$  [12], which can be further approximated to  $|E(\phi)| = \sin \phi$ . Combining these two losses together, we will have the modeled path loss:  $L = 20 \log_{10} f + 20 \log_{10} d - 20 \log_{10} (|E(\phi)|) - 147.56$ , where  $f$  is the central frequency of the channel [12].

We will use efficiency loss instead of packet error rate as our performance metric because the latter tends to underestimate the true packet error rate when SGN is not transmitting due to lost beacons. We will seek a complete statistical characterization of the relationship between modeled path loss and efficiency loss using the multivariate kernel density estimation technique [13].

Recall that a multivariate kernel density estimator with kernel  $k$  and window width  $h$  is defined by

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n k\left(\frac{1}{h}(\mathbf{x} - \mathbf{x}_i)\right),$$

where  $\mathbf{x}_i \in \mathbf{R}^d, \forall i = 1, 2, \dots, n$ , are the observations,  $n$  the number of observations,  $d$  the dimension of  $\mathbf{x}$ , and  $\hat{f}$  an estimate of the joint probability density function of  $\mathbf{x} = (x_1, x_2, \dots, x_d)$ . We will use the multivariate Gaussian kernel:

$$k(\mathbf{x}) = \frac{1}{\sqrt{d/2\pi}} e^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}}.$$

The choice of the window width  $h$  for the multivariate Gaussian kernels will follow the rules of thumb described by Silverman [13].

We briefly give our intuition behind the density estimation technique. We assume that the observed samples are drawn from an unobserved distribution. We could use a multi-dimensional histogram to approximate the density function. However, due to error and noise in observation and measurement, each sample may have actually been contributed by the probability mass from its vicinity regions. The histogram will need to be “smoothed” somehow to properly take into account such erroneous offsets. Also, in the case of continuous

random variables such as path loss and efficiency loss, there are numerous “gaps” between samples. We will need a way to “interpolate” the histogram for the values that are missing from the observations. The kernel density estimate technique provides a way to smooth and interpolate histograms. The kernel function serves as the weighing function in averaging the contributions from neighboring samples for predicting the probability density for a particular point. The choice of width is important because it determines the size of the neighborhoods for averaging and thus will influence smoothness and fidelity of the resulting density estimate. As pointed out by Silverman [13], there does not appear to be a universally good way of choosing this width. All methods are making certain trade-offs in one kind or another. In our case, we have experimented with several different ways of width choosing before we eventually settled down to our decision. Our choice appears to be able to produce reasonable results that agree with our experience with and understanding of the testbed system.

We are now ready to state our modeling approach. For each run, we randomly select about half of the cycles as the training set. We use the multivariate Gaussian kernel density estimation technique to produce an estimate for the joint probability density of modeled path loss and efficiency loss in this training set. Based on this estimated density, we compute the conditional mean efficiency loss conditioning on modeled path loss and use it as the predictor for efficiency loss given modeled path loss. Figure 6 shows a few example predictors

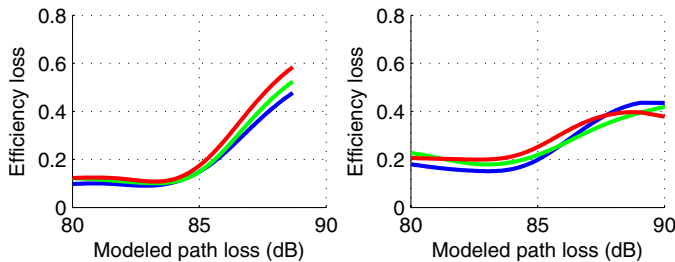


Fig. 6. The estimated mean efficiency loss, derived from different training sets chosen randomly from the measurement data, as a function of modeled path loss for the load (left) and deliver (right) stages in the third run in Table II. The different colored curves in each plot represent different training sets. The similarity in these curves shows that our approach is rather robust to the choice of training sets.

produced by this approach. We show the mean efficiency loss obtained using different training sets from the third run in Table II. We note that first, predictors obtained using different training sets are very similar, so our approach is robust in the sense that it is rather insensitive to the choice of training sets. Secondly, we tend to have a higher efficiency loss when the link quality is poor in the load stage. This is because loss of 3 consecutive beacons can result in efficiency loss of 1 for the next beacon period. It could also be because the source and destination ground nodes are not symmetric in terms of their relative positions with respect to the UAV flight trajectory, as well as the difference in hardware components of these two nodes due to inevitable variations in manufacturing and deployment.

Figure 7 shows several samples of estimated probability densities of efficiency loss for various modeled path losses.

Training set	Our model	Distance	Fixed
Cycle 1–5	6.42%	14.88%	56.75%
Cycle 6–10	2.50%	4.56%	36.21%
Cycle 1,2,5,6,8	5.50%	1.32%	14.82%

TABLE I

The percentage errors between predicted and measured average buffer occupancies for the three models using different training sets from the third run in Table II. Our model not only produces predictions with smaller error, but also has a robust performance that is insensitive to the choice of training sets. Distance-based model can also perform quite well sometimes, showing that distance is indeed the most important performance determining factor.

These plots shows that the general quality the prediction based on modeled path loss is reasonably good. If the conditional density under a particular modeled path loss is highly concentrated around a certain value, then the prediction error in this case will be small. Contrarily, if the density spreads out across a wide range of modeled path losses, like the curves corresponding to the higher path losses in the load stage in Figure 7, the prediction can not be very accurate.

We further quantitatively evaluate the effectiveness of our approach by measuring how good it is in predicting buffer occupancies. For each training set, we use the efficiency loss predicted by our model to compute the average buffer occupancy for the remaining cycles. We then compare the prediction with the measured buffer occupancy. We also make similar predictions using two other straightforward models that employ a number of commonly used techniques for predicting the packet loss rate. The first one is solely based on distance. Specifically, we divide distance into ten fixed-sized bins and use the average efficiency loss in each bin obtained from the training set to predict the efficiency loss in the rest of the cycles. We call this model the “Distance” model. The second model is even simpler—we just predict that the efficiency loss in the rest of the cycles will be the same as the average efficiency loss in the training set. We call this model the “Fixed” model. Table I shows the error percentage between predicted and measured average buffer occupancies for the three models using different training sets from the third run in Table II. We can see that our model significantly outperforms the simple Fixed model in that the errors are much smaller. The performance of the Distance model, on the other hand,

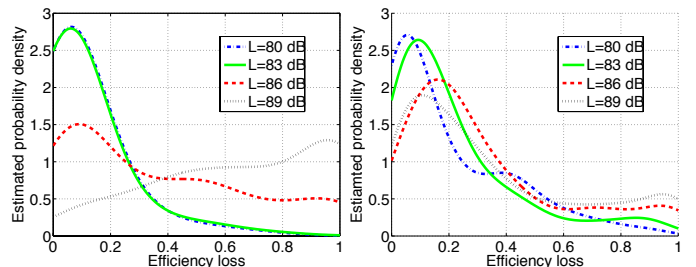


Fig. 7. The estimated conditional probability densities of efficiency loss under several different modeled path losses for the load (left) and deliver (right) stages. The general quality of the prediction is fairly good, as can be seen from the high concentration of probability mass around the peaks in most curves. However, the precision of the prediction does drop as modeled path loss increases.



is fairly close to that of our model. Furthermore, although our model usually performs better, we do see a situation as shown in the last training set where distance does a better job. This conforms with the wide-accepted intuition that distance is the most important factor in determining the performance of a wireless link. However, its performance is less robust across different training sets. The error can be quite significant sometimes, as shown by the first training in Table I. We believe that this is because of its failure of taking elevation angles into accounts in this case.

## V. CONCLUSION

In this paper, we presented the load-carry-and-deliver (LCAD) networking paradigm that is specially designed for maximizing the throughput of UAV-relaying networks. One necessary condition for throughput maximization in such networks is having no overruns or underruns in the UAV node's buffer. To achieve throughput maximization in practice, we will need a performance model for the communication channel, so that we can adjust flight paths in order to minimize the surplus or deficit. This is especially so for our case because the COTS radios we used are not designed and engineered for the rapid changing UAV networking environment.

By using model airplanes and IEEE 802.11g radios, we performed several sets of experiments for the purpose of evaluating LCAD performance, as well as collecting data for deriving an empirical link performance model. The measured performance suggests that the proposed LCAD paradigm can be used to provide high throughput communication between two ground nodes, as compared with the conventional multi-hop, store-and-forward relay chain. The reason for such a result is that we can schedule UAV's transmissions to avoid interference and medium access contention. The trade-off is the higher packet delivery latency. We also showed that the model we derived from several randomly selected training subsets can predict the buffer occupancy of the rest of the data set with small errors. This is an encouraging result because it suggests COTS radio can potentially be used in LCAD application scenarios.

In summary, the contributions we made in this paper include: (1) the design of a light-weight LCAD protocol; (2) the analysis on a few fundamental necessary optimality conditions; (3) UAV flight experiments and throughput measurements for LCAD; (4) demonstrating LCAD's throughput advantage over conventional wireless multi-hop relay protocols; (5) an empirical performance model for predicting the achievable LCAD throughput for UAV networks; and (6) the feasibility of using low-cost COTS radio for UAV networking applications.

## REFERENCES

- [1] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *ACM MobiCom*, July 2001.
- [2] C. M. Cheng, P. H. Hsiao, H. T. Kung, and D. Vlah, "Parallel Use of Multiple Channels in Multi-hop 802.11 Wireless Networks," in *IEEE MILCOM*, October 2006.
- [3] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, and D. Estrin, "Statistical model of lossy links in wireless sensor networks," in *ACM/IEEE Fourth International Conference on Information Processing in Sensor Networks (IPSN05)*, April 2005.

- [4] W. Zhao and M. H. Ammar, "Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks," in *FTDCS '03: Proceedings of the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03)*, 2003, p. 308.
- [5] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in *ACM MobiHoc*, May 2004, pp. 187–198.
- [6] —, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network," March 2005.
- [7] D. Henkel and T. X. Brown, "On Controlled Node Mobility in Delay-Tolerant Networks of Unmanned Aerial Vehicles," in *International Symposium on Advance Radio Technologies (ISART)*, March 7–9, 2006.
- [8] E. W. Frew, T. X. Brown, C. Dixon, and D. Henkel, "Establishment and Maintenance of a Delay Tolerant Network through Decentralized Mobility Control," in *IEEE International Conference On Networking, Sensing and Control*, April 23–25, 2006, pp. 584–589.
- [9] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.
- [10] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004, pp. 126–134.
- [11] "Senior Telemaster R/C Airplane by Hobby Lobby International, Inc." <http://www.hobby-lobby.com/srtele.htm>, 2006.
- [12] R. C. Johnson, *Antenna Engineering Handbook*. McGraw-Hill Professional, 1992.
- [13] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.

## APPENDIX

In Table II, we report the performance and time duration for each individual stage, as well as the UAV's buffer occupancy for each cycle, the cycle time  $T_{cycle}$ , and the overall throughput. The packet error rate (the PER column) is computed from number of packets sent and received in each stage. The length of the stage in seconds is reported in the  $T_{load}$  and  $T_{deliver}$  columns. The *dist* column reports the average distance in meters between transmitter and receiver during the stage and is computed from the GPS log and ground nodes' locations. *lost* and *sent* columns report the total number of lost and sent packets in that stage. The last four columns record the total number of packets delivered by the UAV under LCAD (*dlvd*), the UAV's buffer occupancy at the end of the cycle (*BO*), the cycle time  $T_{cycle}$ , and the utilization. The total number of packets delivered is computed by taking the minimum between the number of packets received by the UAV and that by DGN. Buffer occupancy records the number of packets left in the buffer at the end of the cycle. A number inside parentheses means that the UAV could have sent that many more packets if the buffer were not depleted. In other words, buffer underrun has occurred, and the number quantifies the degree of underrun. To get a better sense of throughput performance, we report the utilization in addition to the packet delivery rate of the cycle. From the analysis of collected traces, the average send rate from the transmitter is at most 398 packets per second. For these reasons, the reported utilization is calculated as packet delivery rate divided by 400 packets per second. An average row is inserted into the table after each run to report aggregated average distances and the average utilization of all cycles in that particular run.

cycle	load stage					deliver stage					overall			
	PER	$T_{load}$	dist.	lost	sent	PER	$T_{deliver}$	dist.	lost	sent	dlvd	BO	$T_{cycle}$	utilization (pkt/s)
1-1	0.4756	40.815	200	5340	11227	0.3445	32.192	168	4367	12675	5887	(2421)	73.007	0.2016 (80.636)
1-2	0.4143	40.819	198	4421	10672	0.3906	36.040	165	5518	14126	6251	(2357)	76.859	0.2033 (81.330)
1-3	0.4486	39.012	194	4351	9700	0.3081	35.014	165	4280	13890	5349	(4261)	74.026	0.1806 (72.259)
1-4	0.3891	41.016	191	4869	12514	0.1828	28.970	155	2103	11503	7645	(1755)	69.986	0.2731 (109.236)
1-5	0.4264	40.028	176	4840	11352	0.3808	29.980	169	4522	11875	6512	(841)	70.008	0.2325 (93.018)
1-6	0.3797	43.059	196	4987	13135	0.1939	29.995	163	2297	11845	8148	(1400)	73.054	0.2788 (111.534)
average			198					164						$0.2283 \pm 0.0369$
2-1	0.1133	42.015	144	1870	16498	0.2486	28.963	140	2840	11422	8582	6046	70.977	0.3023 (120.912)
2-2	0.0658	38.026	130	970	14752	0.2771	24.981	122	2670	9635	6965	6817	63.008	0.2764 (110.542)
2-3	0.1186	33.009	122	1503	12675	0.4790	36.019	126	6766	14126	7360	3812	69.029	0.2666 (106.623)
2-4	0.1223	25.855	138	1245	10182	0.3509	22.130	149	3005	8563	5558	3379	47.985	0.2896 (115.829)
average			144					134						$0.2837 \pm 0.1345$
3-1	0.2050	29.006	128	2126	10370	0.2572	22.038	149	2070	8048	5978	2266	51.044	0.2928 (117.115)
3-2	0.1273	30.008	117	1389	10908	0.1959	28.014	136	2022	10320	8298	1221	58.022	0.3575 (143.015)
3-3	0.0572	32.077	129	694	12135	0.2353	29.014	138	2502	10632	8130	3311	61.091	0.3327 (133.081)
3-4	0.0813	36.820	146	1140	14022	0.1885	27.041	142	1873	9938	8065	4817	63.861	0.3157 (126.289)
3-5	0.0606	32.070	133	735	12128	0.2985	24.065	135	2589	8673	6084	5309	56.135	0.2710 (108.382)
3-6	0.1180	31.833	128	1368	11597	0.2027	26.010	138	1940	9570	7630	2599	57.843	0.3298 (131.910)
3-7	0.0995	29.076	126	1057	10627	0.1856	28.010	132	1902	10250	8348	1222	57.086	0.3656 (146.236)
3-8	0.1544	30.009	131	1695	10980	0.3432	30.011	120	3767	10975	7208	2077	60.020	0.3002 (120.093)
3-9	0.1444	25.855	123	1379	9548	0.2564	22.020	149	2061	8039	5978	2191	47.876	0.3122 (124.865)
3-10	0.0995	31.009	126	1146	11512	0.2622	25.024	145	2381	9081	6700	3666	56.032	0.2989 (119.574)
average			129					138						$0.3176 \pm 0.0278$

TABLE II

Performance summary of the 20 cycles from 3 test runs. The “cycle” column indicates the run and the sequence of the particular cycle in that run. The “PER” column shows the overall packet error rates. The “dist.” column is the average distance between the UAV and ground nodes. The “lost” and “sent” columns show the total number of packets lost and sent during that cycle, whereas the “dlvd” column is the total number of packets delivered from SGN to DGN via UAV during that cycle. The “BO” column shows the UAV’s buffer occupancy at the end of that cycle (number in parenthesis is deficit), and the “utilization” column is the normalized throughput based on the number of packets delivered and the length of the cycle. The three “average” rows summarize the average distance and the average utilization for that particular run.