UAV-Aided Cross-Layer Routing for MANETs

Yuan Guo Xiaolong Li Homayoun Yousefi'zadeh Hamid Jafarkhani
Center for Pervasive Communications and Computing
University of California, Irvine
[yuang,xiaolonl,hyousefi,hamidj]@uci.edu

***Abstract*—In this paper, we present UAV-aided Cross-Layer Routing Protocol (UCLR) that aims at improving the routing performance of a ground MANET network with aid from an Unmanned Aerial Vehicle (UAV). The UAV is added to a connected backbone formed by a collection of designated nodes in order to combat link failures and wireless link effects detected at PHY/MAC layers before the routing table adapts to the changes. In the context of the UCLR protocol, we introduce a UAV-aided cross-layer routing scheme, an associated cross-layer routing metric, and a UAV load-balancing algorithm. We implement UCLR using Linux Quagga routing suite along with OSPF MANET Designated Routing (MDR) and demonstrate its performance improvements compared to the original MDR through emulation studies.**

I. Introduction

Over the past few years, a large body of work has been proposed for efficient routing in Mobile Ad-hoc Networks (MANETs). These solutions generally fall into one of the two categories of reactive (on-demand) routing or proactive (table-driven) routing schemes. Reactive routing protocols, such as Ad hoc On-Demand Distance Vector (AODV) [1] and Dynamic Source Routing (DSR) [2] only perform route discovery from a source node when it needs to send packets to a destination node. While reactive protocols can avoid the overhead associated with disseminating flooding topology information, they introduce rather high latency characteristics. Moreover and as pointed out in [3], these protocols do not scale well because they may generate more overhead than the actual throughput when the network is large and mobility is high. On the other hand, proactive routing protocols can readily refer to their routing tables populated based on some routing criterion such as the shortest path or hop count. However, proactive routing protocols pay a relatively high price in terms of bandwidth overhead in order to keep the topology information up-to-date through the use of flooding algorithms. Some protocols introduce hierarchical structures to reduce both the size of Link State Updates (LSAs) and the number of flooding participants. Two examples of this kind of protocols are HSR [4] and LANMAR [5]. Another trend is to only select partial neighboring nodes for exchanging routing information such that the overall overhead is significantly reduced. In addition, two popular extensions of OSPF [6] are OR [7] and MDR [8]. Experiments show that MDR outperforms OR in terms of scalability and flexibility [9].

Nonetheless, both categories of routing protocols described above face similar challenges imposed by the underlying wireless media. There are issues such as fading, shadowing, interference, jamming and mobility that cause delay, corruption, and loss of packets in an unpredictable fashion. To that end, the lack of availability of link quality information at the network layer may result in selecting a "best" path which is actually of poor quality. The idea of cross-layer routing is motivated by making more accurate routing decisions at the time of forwarding through the use of link quality information available at the MAC/PHY layer. A natural solution is to develop a new routing metric to capture link effects. However, this solution may not be practical for proactive routing protocols which rely on topology flooding. The existing proactive schemes typically propagate interface-based metrics, by which all links going out of the same interface have the same cost. To construct routing tables based on link quality, per-link metrics need to be broadcast instead of per-interface metrics. In the case of a large and dense network, the number of possible links between two nodes dramatically outweighs the number of interfaces and huge extra traffic loads will be generated. Even if the overhead is tolerable, the routing table may never be able to catch up with the pace of link quality variations, which is identified as Time Scale Difference (TSD) problem in [10]. This observation has motivated us to find an alternative capable of utilizing cross-layer information to improve the efficiency of MANET routing.

In this paper, we propose a cross-layer solution utilizing an existing proactive routing protocol with a hierarchical or backbone structure. A routing capable UAV is assumed to be available to the backbone. In our solution, the UAV is used to compensate against wireless link effects within the backbone. Backbone nodes may turn to UAV as the next hop for packet forwarding when a ground connection fails or the link quality greatly degrades. A load-balancing algorithm is also utilized to avoid the overloading of the UAV. Under emergent circumstances such as battlefield or disaster, such deployment is practical, flexible, and responds quickly to inefficient routing caused by bad link qualities. Although our solution works most efficiently within a backbone structure, it can also operate with any proactive routing protocol by treating all nodes as backbone nodes.

Our cross-layer routing solution contributes in several ways. First, we propose a novel idea of using an advantaged backup node to combat wireless link effects. This avoids the extra overhead generated by flooding per-link quality information. Second, we use a load-balancing scheme aimed at preventing the UAV buffer from overflowing by monitoring the number of packets queued within the UAV buffer. Third, our routing solution is compatible with any existing proactive routing

This work was sponsored in part by a research grant from the Boeing Company.

scheme. Finally, we implement UCLR in Quagga routing suite, set up an emulation testbed with CORE [11], and use NetEm [12] emulator for link quality control.

The rest of the paper is organized as follows. In Section II, we describe literature work most closely related to our work. Section III discusses the details of UCLR protocol. In Section IV, we present our testbed and experimental results. Finally, conclusions are drawn in Section V.

II. Related Work

In this section, we discuss literature work including a number of cross-layer routing techniques and a UAV-aided MANET routing scheme that we believe are most closely related to our work.

Several protocols that use the concept of cross-layer routing have been proposed in recent years. These works mostly focus on developing new routing metrics that represent link quality at MAC/PHY layer to assist in making routing decisions. The work of [13] utilizes the delay information at the MAC layer as the routing metric. The routing metric is integrated with DSR which is an on-demand routing protocol for MANETs. However, this approach is not practical for a proactive routing protocol where the routing metric must be broadcast to calculate shortest paths. If link-based routing metrics are flooded, a significant overhead is generated degrading network performance.

A large body of work focuses on developing cross-layer metrics for Wireless Mesh Networks (WMNs). Routing in WMNs differs from MANETs because only the fixed Wireless Mesh Routers (WMRs) perform route discovery. As a result, mobility is not an issue for inter-WMR routing. In the works of [14] and [15], the authors propose a triple-metric that integrates interference, Packet Error Rate (PER), and data rate perceived at the PHY layer is defined and implemented to find paths that offer reduced interference, reliable transmission, and high throughput. The same authors also propose a cross-layer heuristic named Mesh Routing Strategy (MRS) [16] that uses interference, PER, and data rate to construct a path metric instead of integrating them into a link metric. On the other hand, the authors of [17] and [18] propose two well-known metrics: Expected Transmission Count (ETX) and Expected Transmission Time (ETT). ETX and ETT indirectly use link quality information such as packet delivery rate and bandwidth by sending probe packets at the network layer.

Intelligent Hierarchical State Routing (IHSR) with UAV [19] is closest to our work. Similar to our protocol, it also utilizes a UAV to serve each area of ground ad-hoc networks and allows only backbone nodes to access the UAV. Other than that, it differs from our solution UCLR in several aspects. First, while IHSR with UAV focuses on designing a route discovery mechanism that involves UAV, UCLR intends to provide an alternative route through the UAV without changing the original route discovery. Second, with UAV IHSR uses UAV for name resolution of all members of multiple subsets and dissemination of LSAs to backbone nodes. On the other hand, UCLR uses UAV to improve connectivity based on link quality information maintained at the MAC layer. Third, IHSR with UAV does not limit the access to UAV at the NETWORK layer but uses a MAC layer protocol Centralized Intelligent Channel Assigned Multiple Access (C-ICAMA) [20] to dynamically allocate channel bandwidth. In contrast, we design a need-based UAV access algorithm that adapts to the load at the UAV to control routing decisions at the NETWORK layer. Most importantly, UCLR is a cross-layer scheme and is hence able to make intelligent routing decisions to improve the overall efficiency of the network.

III. UAV-aided Cross-Layer Routing

In this section, we present the design of our proposed UCLR Protocol.

A. OSPF-MDR Background

The cross-layer solution described in this paper is primarily builds on OSPF-MDR [8]. OSPF-MDR is a new OSPF interface type designed for MANETs based on OSPFv3 [6]. OSPF-MDR inherits the Designated Router (DR) mechanism from OSPF, adds modifications to overcome the limitations of wireless interface, and also copes with wireless characteristics such as low bandwidth and frequently changing topology. logic.

In OSPF-MDR, MDRs are selected locally based on information contained within Hello messages. Each selected MDR forms adjacencies with a subset of its MDR neighbors to ensure that all MDRs form a Connected Dominating Set (CDS). If bi-connected adjacencies option is chosen, MDRs and Backup MDRs (BMDRs) together form a bi-connected backbone for robustness. Nodes outside the backbone referred to as non-MDR or MDR-other nodes select at least two BMDRs and form adjacencies with them. The MDR neighbor that an MDR-other node selects is called its Parent. The second (Backup) MDR neighbor is called its Backup Parent. As a result, any node in a MANET either belongs to the backbone, or is one hop away from it.

As a proactive routing protocol, OSPF-MDR constructs routing tables by calculating minimum cost paths from a source node to each known destination. The routing cost propagated in router-LSAs is interface-based. In fact, the cost of a path from a source to a destination is calculated by adding up interface costs along the path. However, since link quality information is associated with each link, a single interface cost cannot be used to represent the quality of all links going out of the interface. Consequently and regardless of the type of interface cost used, the routing protocol cannot ensure that a best quality path is selected. If link-based routing cost is propagated instead, a significant flooding overhead will be introduced and scalability is negatively impacted, especially in dense networks where the average degree of node separation is high. MDR is implemented with Quagga routing suite, making it an ideal target for our emulation experiments.

UCLR takes an alternative to improve the efficiency of routing under low link quality conditions. UCLR relies on a UAV for packet forwarding over low quality links. In essence, the UAV serves as a transient packet forwarding alternative for as long as the link quality remains poor. The use of

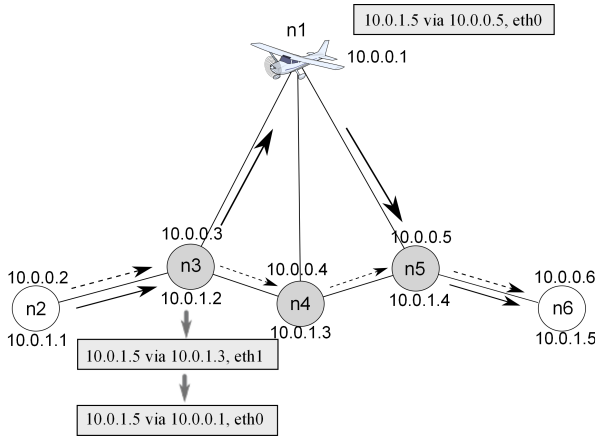

Fig. 1. A UAV-aided cross-layer routing example.

UAV continues until the link quality improves or the routing protocol finds a new route. We also note that the use of UAV can significantly reduce the average hop distance of any pair of arbitrary nodes distributed in either a hierarchical or a backbone structure.

B. UAV-aided Cross-layer Routing

In this section, we present the details of UCLR. First, a proactive routing protocol, such as OSPF-MDR generates the routing tables at each node based on the hop count metric. Then, a backbone node periodically checks the link quality information for the links formed between itself and the next hops in its routing table. Link quality information is provided by the so-called Cross-Layer Interface (CLI) that bridges the MAC layer to the NETWORK layer. If a cross-layer metric calculated from the link quality information is lower than a given threshold, all packets destined to that next hop are forwarded to the UAV until the link quality exceeds the given threshold. The threshold is pre-configured and adaptive to the outgoing queue length of the UAV. In the case of OSPF-MDR, only MDRs are allowed to communicate with the UAV. Notably, cross-layer routing can be enabled or disabled at each node independently without the awareness of its MDR status allowing for flexible deployment.

By means of an example, Fig. 1 illustrates how UCLR operates with OSPF-MDR using a sample set of IP addresses. In the figure, filled circles represent MDRs while unfilled circles represent non-MDRs. Solid lines that connect pairs of nodes represent adjacencies. The node $n1$ is the UAV node while all other nodes are ground nodes. Since the UAV can be elevated to a height at which line of sight links can be established with all ground mobile nodes, it can be considered as a one-hop neighbor of all backbone nodes. The original path from $n2$ to $n6$ generated by OSPF-MDR is shown by the arrowed dotted lines. Specifically, the next hop to $n6$ is $n4$ in the routing table of $n3$. We assume that the link between $n3$ and $n4$ is of low quality, potentially causing significant delay, random bit errors, and even loss of packets going through this link. With UCLR enabled on $n3$, the cross-layer routing component of $n3$ periodically examines the link quality between $n3$ and $n4$ and updates $n1$ as the new next hop once low quality is detected. Upon the arrival of a packet at $n3$, it forwards the packet to the UAV according to its updated routing table. The UAV queries its own routing table for the next hop to $n6$ and forwards the packet to $n5$, which is the last MDR on the path to destination. Notably, if the destination node $n5$ is an MDR, the UAV will directly forward the packet to the destination. Since OSPF-MDR ensures that any node either belongs to the backbone or is one-hop away from it, the path length from the UAV to any ground node is no greater than 2 hops. The actual forwarding path is marked with arrowed solid lines.

Accordingly, we sum up the functions of different nodes:

- The **UAV** is responsible for forwarding incoming packets to destinations or the parent nodes of destinations. It is adjacent to all MDRs. Consequently, flooded LSAs from MDRs are able to cover the entire ground network and the UAV will have an entry for any destination in its routing table. The UAV also disseminates the queue length of its egress interface.
- **MDRs** retrieve link quality information from CLIs and calculate the cross-layer metric at a certain time interval. They also update thresholds based on the queue length of the UAV upon receiving such information. More details about the threshold calculation will be discussed in Section III-D. After comparing the cross-layer metric with the threshold, an MDR decides whether to change the next hop within the path to the UAV instead. Therefore, forwarding packets on low quality links is avoided as much as possible.
- **MDR-others** and **BMDRs** do not perform cross-layer routing. Even if the cross-layer option is enabled on these nodes, they will not react to link effects until they become MDRs.

C. Cross-Layer Routing Metric

We use a simple metric that is proportional to the successful delivery ratio over a link for evaluating the link effects. The metric τ is calculated as:

$$\tau = \lambda(1 - PER_f)(1 - PER_r) \tag{1}$$

where PER_f and PER_r are packet error rates on forward and reverse directions, respectively. The parameter λ is a positive constant used to control the range of τ. When PER_f and PER_r increase, τ becomes smaller. Therefore, a lower metric indicates a worse link quality. In fact, τ is a variation of a well-known link quality metric ETX [17]. While ETX measures forward and reverse delivery ratios by sending small probe packets, PER_f and PER_r are measured at the MAC layer and are available at the NETWORK layer via the cross-layer components. An MDR calculates τ for each of its outgoing links to its next hops. The parameter τ is then compared with the threshold Th to decide whether packets should be forwarded to the UAV instead. In the next subsection, we will present an adaptive algorithm to calculate Th.

D. UAV Load-Balancing Algorithm

Since the underlying link quality is unpredictable and changes over time, it is not practical to use a fixed value threshold. If the threshold is too low, the UAV may be idle most of the time resulting in under utilization of its resources. More importantly and if the threshold is too high, it is possible that multiple ground links are affected by link effect degeradations simultaneously and most if not all of the flows that originally go through them are imposed on the UAV. Due to the limited bandwidth of UAV-to-ground links, packets are queued at the UAV and need to wait for their turn to be serviced. Since the queue size is also limited, overflow may occur resulting in direct discarding of those packets that arrive when the queue is full. Consequently, over utilization of the UAV results in worsening network performance.

We have developed a simple yet adaptive algorithm to adjust the threshold Th dynamically based on the queue length of the UAV. According to our algorithm, the queue length information is broadcast at some time interval, for example, by appending to Hello or LSA updates. Each MDR starts with a pre-configured Th and performs the following upon receiving an update for the value of q:

$$\begin{cases} Th' = Th + \theta & \text{if } q = 0 \\ Th' = Th - \delta \frac{q}{q_m} & \text{if } q > 0 \; and \; \mu < \rho \end{cases} \quad (2)$$

In the equations above, Th and Th' are the original and updated thresholds, respectively. Positive constants θ and δ control the speed of growth of Th, q is the number of queued packets by the UAV, q_m is the queue capacity measured in a fixed packet size, μ is a uniformly distributed random number between [0,1], and ρ is a constant value within the interval $(0, 1]$.

The threshold is increased by θ if the queue is empty, indicating that the UAV is able to accommodate a larger number of packets. A higher threshold allows more ground links to be considered as "intolerable" and more routes use the UAV as an intermediate node. As a result, more packets are forwarded to and queued at the UAV. If the queue is not empty, the threshold will decrease to prevent the queue from continuing to grow. The higher the q is, the faster the threshold decreases. Moreover, to reduce the chance of having an oscillatory value of q, we allow only a subset of MDRs to update their thresholds at the same time using a certain probability.

Then, we truncate Th after performing an update in (2) as:

$$\begin{cases} Th' = \alpha & \text{if } Th' > \alpha \\ Th' = \beta & \text{if } Th' < \beta \end{cases} \quad (3)$$

where α and β are positive constants. We note that Equation (3) limits the range of Th in the range of [α, β]. The lower bound controls the recovery speed of Th. The upper bound represents the limit of the cross-layer metric above which the link quality is considered good and there is no need to use the UAV. To make sure that the UAV is utilized, β should be a smaller value than λ. Clearly, β and α should cope with the value of λ and only the ratio between them matters.

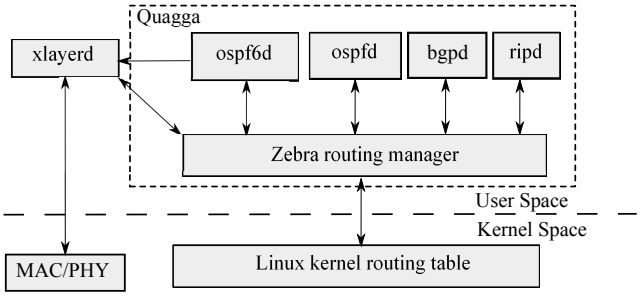

Fig. 2. The implementation of UCLR cross-layer routing in Quagga.

IV. Experimental Studies

A. Implementation

Fig. 2 shows the detailed implemenattion architecture of UCLR within OSPF-MDR. OSPF-MDR is implemented in the Quagga routing suite as a part of the $ospf6$ daemon which in turn implements OSPFv3. In Quagga, all routing protocols communicate with the Linux kernel via Zebra routing manager that utilizes Netlink socket interface. The Netlink socket interface also serves as the CLI to pass on link quality information from the MAC/PHY layer to the NETWORK layer. For the purpose of our emulation experiments, we assume that the MAC/PHY layer is running in the kernel space. However, we note that the MAC/PHY layer can also run in the user space without requiring to modify the interfaces. In our experiments, the MAC/PHY layer is replaced by the NetEm simulator. The cross-layer routing component is implemented as a new daemon in Quagga to which we refer as $xlayer$ daemon.

We create a single-area random network topology consisting of 20 ground nodes and one UAV node in an area of 1500×1125 meters. The random topology is generated by CORE [11] while virtual hosts are created using Linux network name spaces. Our routing experiments run in real-time and use live network traffic. Each virtual node instantiates Quagga version $0.99.16mr1.0$ [21] with our modifications to provide cross-layer functionality. Our experiments run on Fedora Core 14 distribution of Linux operating system and the kernel version is 2.6.35.

Each ground node has two different interfaces: one is used for communications among ground nodes and the other is used for ground MDRs to communicate with the UAV. In the real world, a different frequency should be used to access the UAV in order to avoid interference with the ground communications. In our emulation, we rely on two WLANs, one for ground nodes only and the other for all nodes communicating with the UAV as well as the UAV itself. This allows us to naturally separate these two categories of nodes. By turning off the second interface on non-MDR nodes and disabling communications between ground MDRs using the second interface, we form a topology similar to the one depicted in 1. For the ground WLAN, the default configuration of CORE is used with a wireless transmission range of 275 meters and a bandwidth of $54Mbps$. NetEm provides PHY/MAC support and propagation loss follows the Rayleigh model. Furthermore, mobility is provided using the random waypoint mobility model [22].

TABLE I
PARAMETERS USED FOR THE CROSS-LAYER METRIC AND THE UAV LOAD-BALANCING ALGORITHM.

λ	100
θ	2
δ	15
α	70
β	95
ρ	0.8

Different rules apply to the LAN that includes the UAV. We have modified CORE to allow the UAV to access all ground nodes rather than deciding whether two nodes are reachable by calculating their physical distance. In this work, a single UAV serves a single ground area and no mobility is imposed on this UAV. The UAV has a bandwidth of $5Mbps$ and an outgoing queue size of 200 packets. The queue serves packets on a FIFO basis. The real-time queue length information is appended to OSPFv3 Hello messages and broadcast by the UAV to all ground nodes. Broadcast queueing length is the average value of 20 samples within a Hello interval, which is 1 second. Additionally and in order to prevent ground MDRs from automatically adding the UAV as the next hop to their original routing algorithm without referring to the underlying link quality, we set a much higher interface cost for the UAV compared to all ground nodes.

Iperf is used to create TCP and UDP traffic and measure network performance. We set the MAC frame size as 1500 bytes. Each emulation runs for 300 seconds. More detailed configuration related to the experiment is discussed in what follows.

B. Experimental Results

First, we evaluate TCP performance by randomly selecting the pairs of source and destination nodes to act as clients-servers and generating TCP traffic using iperf. To ensure that the UAV is used in the case of bad link quality, we choose the first 5 pairs of nodes from a random set that needs to use at least one MDR as an intermediate node. For each TCP flow, client and server window sizes are both set to $8MBytes$.

Fig. 3 compares the average end-to-end TCP throughput of original OSPF-MDR, UCLR with Load-Balancing (UCLR-LB), and UCLR without Load-Balancing. The parameters used for equations (1), (2), and (3) are shown in Table I. As shown in the figure, both UCLR-LB and UCLR approximately double the average throughput of OSPF-MDR. It is worth noting that the load-balancing algorithm does not contribute to the performance in the case of TCP. This is because TCP has its own congestion control mechanism that forces the senders to reduce their transmission rate when packets are congested and dropped at the UAV. As a result, fewer packets are sent to the UAV and overloading is avoided.

Next, we evaluate the performance of UCLR carrying UDP traffic. In contrast to TCP, UDP does not provide reliable transmission using an acknowledgment mechanism and the sending rates are not affected by network conditions. We use the same 5 pairs of sources and destinations to measure the performance of UDP while the rate of each flow is $20Mbps$.

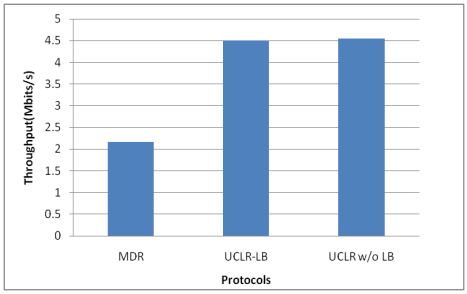

Fig. 3. A performance comparison of TCP throughput.

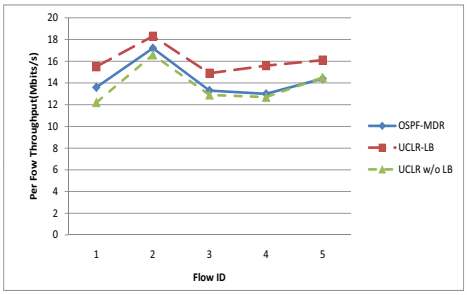

Fig. 4. A per-flow performance comparison of UDP throughput.

Fig. 4 shows a comparison of per flow throughput for OSPF-MDR, UCLR-LB, and UCLR (w/o LB). The parameters used for UCLR-LB are shown in Table I while the threshold value used for UCLR w/o LB is fixed at β.

Figure .4 shows that UCLR-LB can increase the throughput of OSPF-MDR by up to 20%. The UDP throughput of UCLR-LB achieves 91.5% of the ideal throughput at Flow 2. Furthermore, UCLR w/o LB performs even worse than OSPF-MDR because ground MDRs are unaware of congestion and keep forwarding packets to the UAV causing a 100% loss of overflowed packets. The result shows the accurate functioning of our protocol and the effectiveness of our load-balancing algorithm. Fig. 5 shows the same trend of improvement by comparing per flow delivery ratios. By avoiding transmitting packets over higher loss rate links, UCLR-LB consistently delivers 8% more packets than OSPF-MDR.

Finally, we investigate the effectiveness of our adaptive load-balancing algorithm for accessing the UAV. We measure the performance of UDP for different values of β over λ

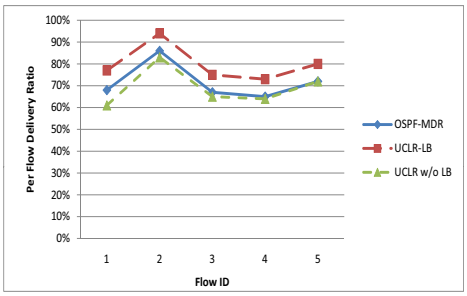

Fig. 5. A comparison of UDP per-flow delivery ratio.

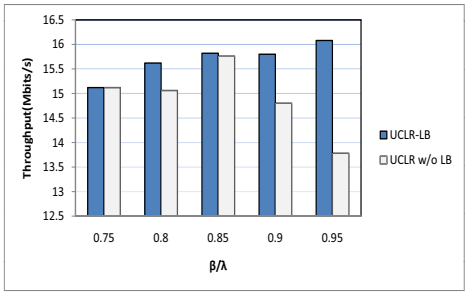

Fig. 6. The average UDP throughput over β/λ.

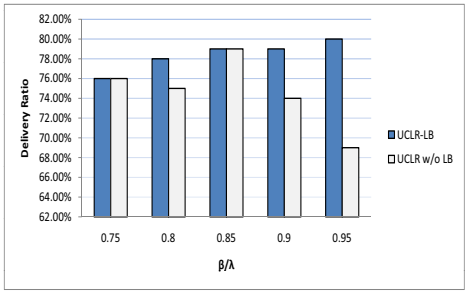

Fig. 7. The average UDP delivery ratio over β/λ.

representing the upper bound of delivery ratio for a poor quality link. To be more specific, if λ is 1, then the cross-layer metric τ is essentially the delivery ratio. Since β is the upper limit of the threshold, packets should not be forwarded to the UAV if τ exceeds β, which indicates that the ground link is good.

Fig. 6 and Fig. 7 show the average throughput and delivery ratio of UCLR with or without load-balancing over β/λ, respectively. As expected, UCLR-LB performs at least as good as UCLR and outperforms UCLR in all cases. When β/λ is too small, the chance of utilizing UAV to improve connectivity is slim. As the value of β/λ grows, congestion may happen at the UAV causing packet loss. As a result, UCLR achieves a peak throughput of $15.76Mbits/s$ at 0.85. It is worth noting that the link effects are unpredictable in the real world, and hence there is no way to guarantee optimal parameters are chosen to yield peak performance. In the case of UCLR, performance can be even worse than that of a case in which UAV is absent. This is shown in Fig. 4. To the contrary, the throughput of UCLR-LB gradually increases as the value of β/λ increases. It reaches a peak throughput of $16.08Mbits/s$ when β/λ is 0.95. This justifies that our load-balancing algorithm is able to make an efficient use of the UAV while reducing the probability of congestion.

V. Conclusion

In this paper, we proposed the UAV-aided Cross-Layer Routing (UCLR) protocol for MANETs. We explained how UCLR interacts with an existing routing protocol to perform cross-layer routing improvements. Furthermore and through the use of our proposed load-balancing algorithm, we described how UCLR could intelligently utilize the limited resources of a UAV in order to avoid forwarding packets over poor quality ground links. We implemented UCLR in Linux Quagga routing suite with OSPF-MDR. Through experimental studies, we demonstrated that UCLR can significantly improve the throughput and delivery ratio of both TCP and UDP. Lastly, we were able to show that our load-balancing algorithm could effectively reduce the congestion level at the UAV and therefore was necessary for optimal deployment of the UAV.

References

[1] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IETF RFC 3561*, July 2003.
[2] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," *IETF RFC 4728*, Feb. 2007.
[3] C. P. S.R. Das and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *IEEE INFOCOM*, Israel, Mar. 2000, pp. 3–12.
[4] G. Pei, M. Gerla, X. Hong, and C. C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," in *IEEE WCNC'99*, Sept. 1999.
[5] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," in *IEEE/ACM MobiHOC 2000*, Aug. 2000, pp. 11–18.
[6] R. Coltun, D. Ferguson, and J. Moy, "OSPF for IPv6," *IETF RFC 2740*, Dec. 1999.
[7] E. A. Roy and E. M. Chandra, "Extensions to OSPF to Support Mobile Ad Hoc Networking," *IETF RFC 5614*, Mar. 2010.
[8] R. Ogier and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding," *IETF RFC 5614*, Aug. 2009.
[9] P. A. Spagnolo and T. R. Henderson, "Comparison of Proposed OSPF MANET Extensions," *IEEE MILCOM*, Oct. 2006.
[10] X. Li and H. Yousefi'zadeh, "A hybrid cross-layer routing protocol for manets," in *Proc. ICCCN*, July 2011.
[11] T. H. J. Ahrenholz, C. Danilov and J.H.Kim, "CORE: A Real-Time Network Emulator," in *Proc. IEEE MILCOM 2008*, Nov. 2008.
[12] S. Hemminger, "Network emulation with netem," in *Proc. LCA, 2005*, Apr. 2005.
[13] H. L. W. Yuen and T. Andersen, "A Simple and Effective Cross Layer Networking System For Mobile Ad Hoc Networks," in *IEEE IPMRC'02*, 2002, pp. 1–5.
[14] K. S. L. Iannone, R. Khalili and S. Fdida, "Cross-layer routing in wireless mesh networks," in *Proc. Wireless Communication Systems, 2004*, Sept. 2004.
[15] K. K. L. Iannone and S. Fdida, "The Real Gain of Cross-Layer Routing in Wireless Mesh Networks," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Sept. 2006.
[16] L. Iannone and S. Fdida, "MRS: A Simple Cross-Layer Heuristic to Improve Throughput Capacity in Wireless Mesh Networks," in *Proc. ACM CoNEXT'05*, 2005, pp. 21–30.
[17] D. A. J. B. D. De Couto and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," in *Proc. ACM IMobicom*, Sept. 2003.
[18] J. P. R. Draves and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *Proc. ACM IMobicom*, 2004.
[19] D. L. Gu, G. Pei, H. Ly, M. Gerla, B. Zhang, and X. Hong, "UAV Aided Intelligent Routing for Ad-Hoc Wireless Network in Single-area Theater," in *Proc. IEEE WCNC 2000*, Sept. 2000.
[20] D. L. Gu, H. Ly, X. Hong, M. Gerla, G. Pei, and Y.-Z. Lee, "C-ICAMA, A Centralized Intelligent Channel Assigned Multiple Access for Multi-Level Ad-Hoc Wireless Networks with UAVs," in *Proc. IEEE WCNC 2000*, Sept. 2000.
[21] -, "Download Quagga 0.99.16mr1.0," available at http://downloads.pf.itd.nrl.navy.mil/ospf-manet/quagga-0.99.16mr1.0/.
[22] a. X. P.-C. C. Bettstetter, H. Hartenstein, "Stochastic Properties of the Random Waypoint Mobility Model," in *Proc. ACM MSWiM'02*, Sept. 2002.