Intelligent Wireless Ad Hoc Routing Protocol and Controller for UAV Networks

Abhinandan Ramaprasath[1], Anand Srinivasan[2],
Chung-Horng Lung[1](✉), and Marc St-Hilaire[1]

[1] Department of Systems and Computer Engineering,
Carleton University, Ottawa, Canada
{abhinandan.ramaprasath, chlung}@sce.carleton.ca,
marc_st_hilaire@carleton.ca
[2] EION Inc., Ottawa, Canada
anand@eion.com

Abstract. In this paper, we propose a novel UAV to UAV communication approach that is based on the concept of Software Defined Networking (SDN). The proposed approach uses a controller as a central source of information to assign routes that maximize throughput, distribute traffic evenly, reduce network delay and utilize all network elements. Simulation results of the proposed methodology were compared to the performance of two common ad hoc routing protocols, namely AODV and OLSR. Performance analysis shows that the proposed methodology improves throughput by over 300%. Simulation results also show a reduction in network delay for delay sensitive packets by nearly 25% and a 26 times increase in packet delivery ratio for packets with higher priority.

Keywords: UAV networks · SDN · Wireless Ad hoc networks · OLSR · AODV · Routing protocols · Performance · Simulation

1 Introduction

Networks of Unmanned Aerial Vehicles (UAVs), also known as Unmanned Aeronautical Ad hoc Networks (UAANET), can be used as an alternative when ground communication is not possible (e.g., disaster recovery, forest fire, etc.). A network of UAVs can span areas of many square kilometers and should be resilient to changes at the ground level. Since a UAV can be moved around the area on demand, a network using UAVs is flexible and scalable. Additional UAVs can be deployed on demand to expand the area of connectivity or replace dying UAVs. The density of UAVs can also be increased in areas where there is a higher demand for network resources. Moreover, this network must be able to provide Internet or network connectivity to users and be able to support transmitting priority packets that need to reach the destination before other packets.

To efficiently utilize network resources and maximize throughput, a central repository, also known as "controller", is used to store and process all the information. A controller monitors the network providing data to network administrators and

Y. Zhou and T. Kunz (Eds.): ADHOCNETS 2016, LNICST 184, pp. 92–104, 2017.
DOI: 10.1007/978-3-319-51204-4_8

assigning routes within the UAV network for data and control packet transmissions. The controller is the main entity in Software Defined Networks (SDN) and therefore, applying SDN concepts to UAV networks can provide advantages.

In this paper, we propose a new UAV to UAV communication scheme based on the concept of SDN. In our approach, UAVs create a backbone infrastructure that is scalable, provides high network efficiency in terms of bandwidth and latency, and supports packets with different priority levels. To that end, the rest of this paper is organized as follows. Section 2 describes a brief review of relevant work. Section 3 discusses the proposed UAV to UAV communication scheme. Section 4 presents the performance evaluation of our proposed scheme and two commonly used routing protocols. Finally, Sect. 5 concludes the paper.

2 Related Work

Networks of UAVs are not new. The authors in [17] provide a good survey of flying ad hoc networks and point out that the most important design aspect is communication. An efficient communication protocol must be used to enable proper cooperation between UAVs. A typical example of UAV to UAV communication protocol was proposed in [18]. The authors proposed to combine AODV and greedy geographic forwarding (called Reactive-Greedy-Reactive (RGR)) in order to improve delay and packet delivery ratio. However, the protocol inherits from some of the drawbacks of AODV. Moreover, there is no notion of centralize controller and priority levels.

The concept of SDN has been used to improve different kind of networks. For example, the authors in [14] show the benefits of having Wireless Local Area Networks (WLANs) on top of the SDN/OpenFlow infrastructure. A SDN controller can manage the access points (APs) and the way they behave. By switching the routes and the traffic flow pattern beforehand, the authors demonstrated that a SDN-based WLAN can reduce the switching time from 2.934 s to 0.85 s. However, the problem with this approach is that the switch from one AP to the next is made by the client. This means that the network has no control over which user device is connected to which AP. In our problem, there is a need to load balance the network to ensure that users are distributed evenly and the traffic among the UAVs is also evenly distributed.

The approach proposed in [15] explains how nearby controllers can be used to create a scalable architecture using a WiFi SDN network. A similar architecture could be useful for our proposed approach as the UAV network scales. A nearby controller only controls its immediate environment. As the network scales, a hierarchical structure of controllers can be used to obtain information from these nearby controllers. The proposed approach in [15], however, does not solve the problem of finding the optimal route to the destination, providing seamless roaming of users within a network, or providing any energy management techniques. The authors in [16] also proposed a hierarchy of controllers. The proposed approach enables the deployment of UAV based WiFi networks in places where there is no existing WiFi infrastructure. It also enables the transfer the user device over to a different WiFi network when connectivity is available to reduce the load on the UAV network. The approach is proposed to solve the problem of optimizing connectivity in a dense and heterogeneous network.

Optimized Link State Routing version 2 (OLSRv2) [3] offers significant performance improvements and other benefits over its predecessor. OLSRv2 is known to show significant improvements in route discovery times, much better performance in terms of bandwidth and data transfer volume, offers support for discovery of the shortest link to a given node and lower power consumption per node [4, 5]. OLSRv2 still does not guarantee that the most optimal route in terms of both bandwidth and latency will be selected. In addition, multi-route packet transfer is still not possible and there is no provision for priority packet transfer.

Ad hoc On-Demand Distance Vector (AODV) routing is a popular routing protocol for reactive routing [1]. AODV was designed for mobile nodes when the network is constantly changing. In order to efficiently use network resources, we may need to switch routes during operation and therefore fixed routes until they expire would be disadvantageous. AODV does not support priority levels or multipath routing. Many other enhancements to AODV fail to address these concerns [6–9].

In our problem, UAVs are intended to provide Internet access over a city or for a disaster area. In most traditional routing protocols, traffic is typically routed through the same paths, causing those UAVs to drain power quickly, while some UAVs are underutilized. Since battery life is crucial for UAVs, it is important to conserve energy by distributing traffic evenly throughout the network. It is also important to prioritize important packets so that they can be delivered first.

3 Proposed UAV to UAV Communication Scheme

This section presents the proposed approach. Figure 1 shows a simplified ad hoc network. We have a source that needs to transmit information to the destination (Internet) and the packet has to go through a series of hops (as directed by the controller) to reach the sink. The sink then relays the packet to the destination completing the packet transfer. In the next sections, we will discuss different aspect of the communication scheme.

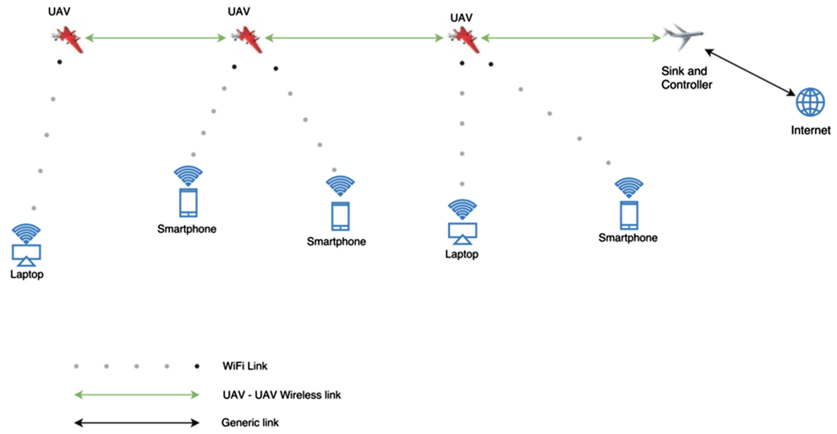


Fig. 1. Base network architecture

3.1 Priority Levels

In order to maintain the stability of the network, packets have to be prioritized. In our approach, since decisions are made by the controller, we need to make sure that these decisions are made and communicated quickly without delay. To facilitate this, packets transmitted through the network are classified into one of the four categories described below:

1. Priority control packets: These are packets with the highest priority and require immediate attention.
2. Non-priority control packets: These are control packets that are sent to the controller at regular intervals (every 30 s for example).
3. Priority data packets: Data packets are categorized into two levels: priority and non-priority. These levels are determined by the UAVs by monitoring the data sent over the network.
4. Non-priority data packets: All data packets that do not fall under the priority data packets fall under the non-priority category. For example, requests to access a web page or streaming music from the Internet.

3.2 Network Setup and Discovery

Initially, we assume that all UAVs are dispatched from a base station. During dispatch, each UAV receives an initial location from the controller. Once all UAVs have reached their destination, each UAV starts transmitting HELLO messages. UAVs will also listens to HELLO messages for a specified amount of time (e.g. 30 s) to discover neighbors.

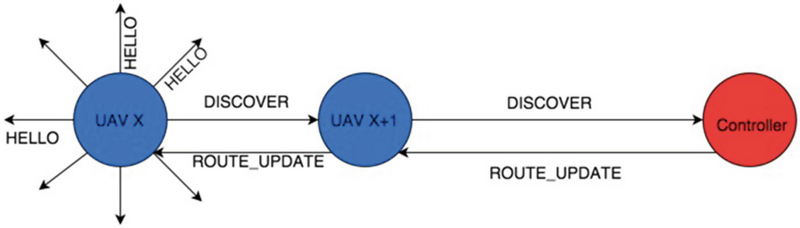
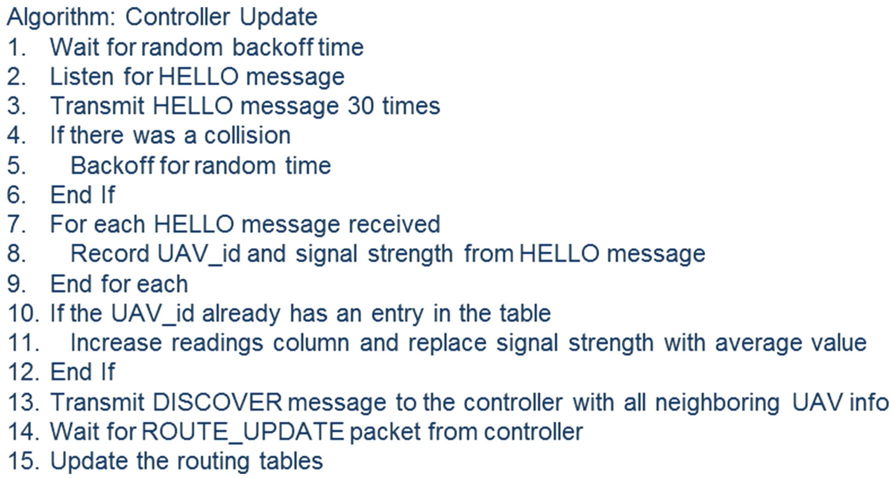


Fig. 2. Network setup and discovery

Once a UAV has discovered its neighbors, it transmits a DISCOVER packet to the controller as shown in Fig. 2. Each DISCOVER packet contains the following 6 fields: (1) UAV ID, (2) number of HELLO messages received, (3) average signal strength (dBm), (4) variance of signal strength (dBm), (5) highest received signal strength (dBm) and (6) lowest received signal strength (dBm).

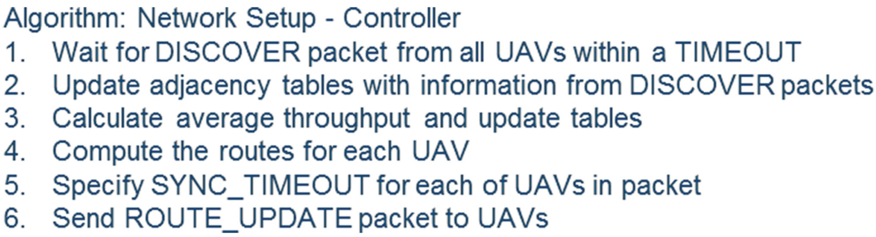
Once a DISCOVER packet is constructed, all UAVs forward this packet to the controller using AODV. AODV is used because the routes have not been established by the controller yet. The algorithm for UAV network setup is summarized below.



```
Algorithm: Controller Update
Wait for random backoff time
Listen for HELLO message
Transmit HELLO message 30 times
If there was a collision
   Backoff for random time
End If
For each HELLO message received
   Record UAV_id and signal strength from HELLO message
End for each
If the UAV_id already has an entry in the table
   Increase readings column and replace signal strength with average value
End If
Transmit DISCOVER message to the controller with all neighboring UAV info
Wait for ROUTE_UPDATE packet from controller
Update the routing tables
```

The controller waits for DISCOVER packets from all the UAVs in the network until a timeout has elapsed. Any UAVs that failed to communicate are marked as lost so the network administrators can take appropriate actions. Using the information within the DISCOVER packets, the controller constructs an adjacency table. Using the average signal strength values, the average modulation scheme for the communication is calculated (explained below). Since the frequency of communication is fixed, the maximum throughput for the given measurements is calculated using the Shannon-Hartley theorem [13]. This way, the controller knows how much information can be transmitted with each link.

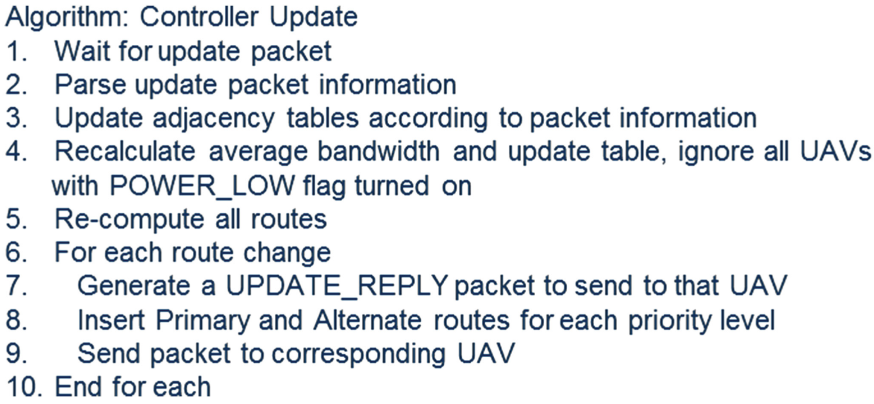
Dijkstra's algorithm [10] and the Ford-Fulkerson algorithm [12] are used to calculate next hops for control and data priority levels, respectively. Since control packets are required to reach the destination at its earliest, we need a route with the smallest number of hops to the controller. An alternate route is also calculated by removing all the links from the main route and re-running Dijkstra. This is based on the assumption that there are multiple connections available between the UAVs. If a UAV has no other connections, then the main and alternative hops point to the same UAV in the network. Similarly, the Ford-Fulkerson algorithm is used to calculate the main and alternate routes for data packet priority levels. After the calculations, SYNC_TIMEOUT is set by the network administrator. SYNC_TIMEOUT specifies how often a UAV sends reports to the controller under stable conditions. If the network is not stable, reports are generated immediately. All of the above information is bundled into a ROUTE_UPDATE packet that contains the following information: (1) UAV ID, (2) SYNC_TIMEOUT (in seconds or ms), (3) Optimal route for priority control packets (UAV ID), (4) Alternate route for priority control packets (UAV ID), (5) Optimal route for non-priority control packets (UAV ID), (6) Alternate route for non-priority control packets (UAV ID), (7) Optimal route for priority data packets (UAV ID), (8) Alternate route for priority data packets (UAV ID), (9) Optimal route for non-priority data packets (UAV ID) and (10) Alternate route for non-priority data packets (UAV ID). This process is summarized in the algorithm below.



```
Algorithm: Network Setup - Controller
Wait for DISCOVER packet from all UAVs within a TIMEOUT
Update adjacency tables with information from DISCOVER packets
Calculate average throughput and update tables
Compute the routes for each UAV
Specify SYNC_TIMEOUT for each of UAVs in packet
Send ROUTE_UPDATE packet to UAVs
```

3.3 Reports and Route Updates

Once a route is established, communications can take place in the network. Data can flow from the source to the sink using the routes defined by the controller during the setup. In order to keep the network functional and for routes to be periodically updated, it is necessary that each UAV sends regular reports to the controller which processes them as shown by the algorithm below.

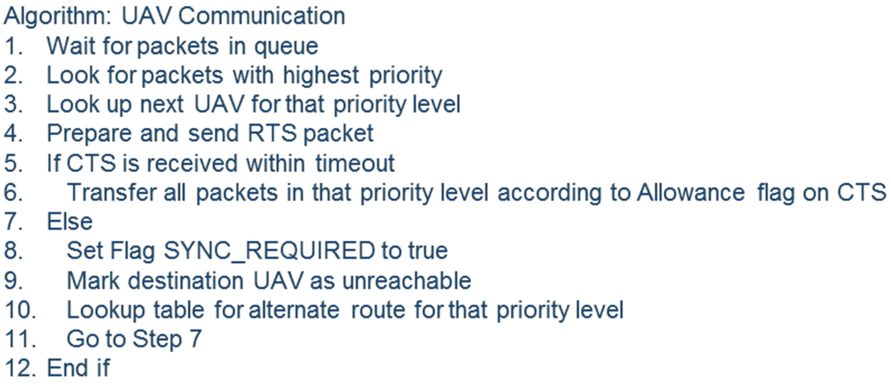


```
Algorithm: Controller Update
Wait for update packet
Parse update packet information
Update adjacency tables according to packet information
Recalculate average bandwidth and update table, ignore all UAVs
with POWER_LOW flag turned on
Re-compute all routes
For each route change
    Generate a UPDATE_REPLY packet to send to that UAV
    Insert Primary and Alternate routes for each priority level
    Send packet to corresponding UAV
End for each
```

Other components of the UAV (such as flight control or altimeter) need to be synchronized with the controller. We also developed an algorithm for UAVs to generate an UPDATE packet that is sent to the controller. The algorithm is not presented here due to the page limit.

3.4 UAV-UAV Communication

UAV to UAV communications take place using all the components mentioned above. Once a packet is queued at a UAV, an RTS packet is generated with an Allowance flag according to the priority level of the packet. It waits for a CTS packet with a certain timeout. If the timeout is exceeded, then the sender assumes that the receiving UAV is lost and restarts the RTS/CTS communication using the alternate route. A UAV marked as lost triggers the SYNC_REQUIRED flag to be set forcing the UAV to notify the controller of this change described in the UAV communication algorithm.



```
Algorithm: UAV Communication
1.  Wait for packets in queue
2.  Look for packets with highest priority
3.  Look up next UAV for that priority level
4.  Prepare and send RTS packet
5.  If CTS is received within timeout
6.    Transfer all packets in that priority level according to Allowance flag on CTS
7.  Else
8.    Set Flag SYNC_REQUIRED to true
9.    Mark destination UAV as unreachable
10.   Lookup table for alternate route for that priority level
11.   Go to Step 7
12. End if
```

4 Performance Evaluation

This section begins with the simulation of a single source with a single priority level to compare base performance against AODV and OLSR. Then, the complexity of the network is increased by adding more priority levels and more sources. The simulations were performed in a Linux environment using NS-3 version 3.24 (NS-3.24).

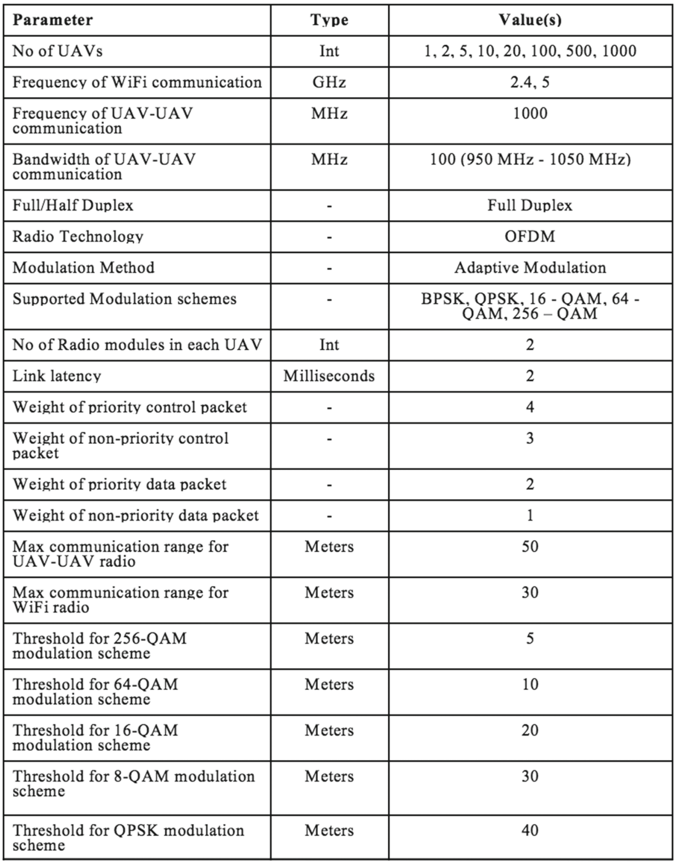
4.1 Network Characteristics and Parameters

The controller was implemented as a class called the “UAVController”. The frequency used for the UAV-UAV wireless link was 1 GHz and a bandwidth of 100 MHz. Each UAV was positioned before the simulation began and UAVs were made to form connections as soon as the simulation began. The wireless channels follow properties defined by the NS-3 framework, which are listed in Table 1.

To visualize the network during simulations, we have used a tool called NetAnim. NetAnim is a Qt based visualizer tool and is part of the “ns3-allinone” package. This paper uses NetAnim version 3.106.

Table 2 describes the network discovery parameters used in the proposed approach. In this paper, we will generate 30 HELLO messages with intervals of one second between each HELLO message. If there is a collision of HELLO packets, the proposed approach backoff for a random time (any value from 5 ms to 500 ms). Since packets are transmitted at regular intervals if there are no collisions during the transmission of the first HELLO message, it is not likely that collisions will happen during subsequent transmissions.

Table 1. Network characteristics



Parameter	Type	Value(s)
No of UAVs	Int	1, 2, 5, 10, 20, 100, 500, 1000
Frequency of WiFi communication	GHz	2.4, 5
Frequency of UAV-UAV communication	MHz	1000
Bandwidth of UAV-UAV communication	MHz	100 (950 MHz - 1050 MHz)
Full/Half Duplex	-	Full Duplex
Radio Technology	-	OFDM
Modulation Method	-	Adaptive Modulation
Supported Modulation schemes	-	BPSK, QPSK, 16 - QAM, 64 - QAM, 256 – QAM
No of Radio modules in each UAV	Int	2
Link latency	Milliseconds	2
Weight of priority control packet	-	4
Weight of non-priority control packet	-	3
Weight of priority data packet	-	2
Weight of non-priority data packet	-	1
Max communication range for UAV-UAV radio	Meters	50
Max communication range for WiFi radio	Meters	30
Threshold for 256-QAM modulation scheme	Meters	5
Threshold for 64-QAM modulation scheme	Meters	10
Threshold for 16-QAM modulation scheme	Meters	20
Threshold for 8-QAM modulation scheme	Meters	30
Threshold for QPSK modulation scheme	Meters	40

Table 2. Proposed approach - discovery and setup

Parameter	Type	Value(s)
INIT TIMEOUT	Seconds	90
No of HELLO Messages	Int	30
HELLO INTERVAL	Milliseconds	1000
HELLO COLLISION INTERVAL	Milliseconds	Rand (5, 500)
NETWORK REDISCOVERY	Seconds	30

4.2 Single Priority Test

In this setup, there is a single source that produces packets and all packets have the same priority level. None of the intermediary nodes generate traffic; they are present only to relay the packets generated by the source to the sink. The source is set to

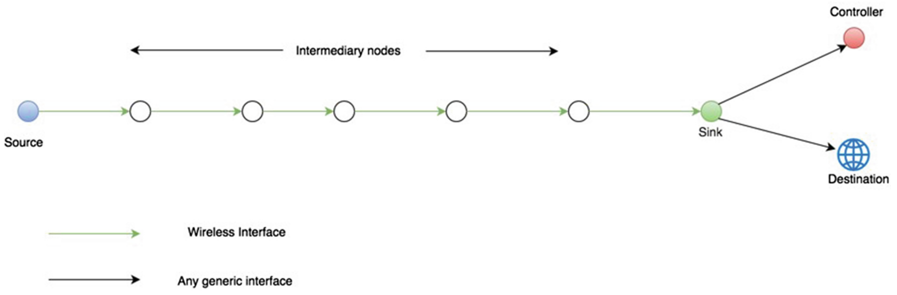


Fig. 3. Linear topology for simulation

produce packets at the rate of 100 Mbps with 5 intermediate hops (see Fig. 3). We set the priority level to the lowest to compare the proposed approach against AODV and OLSR.

Each wireless link is capped at 16 Mbps maximum throughput. The UAVs are placed sufficiently close to each other so that the modulation scheme is no longer a factor. Throughput was measured in the sink in intervals of one second. In order to average out any errors and randomness, the experiment was run 10 times and the average was calculated and plotted in Fig. 4. As can be seen, the throughput performance of the proposed approach is comparable to AODV and OLSR. The average throughput for AODV and OLSR were 15.065 Mbps and 14.892 respectively, while the average throughput of the proposed approach was 14.452 Mbps. This is in line with the expected results for the proposed approach, i.e., the proposed approach has a slightly lower throughput in this case due to the overhead of transmitting update packets.

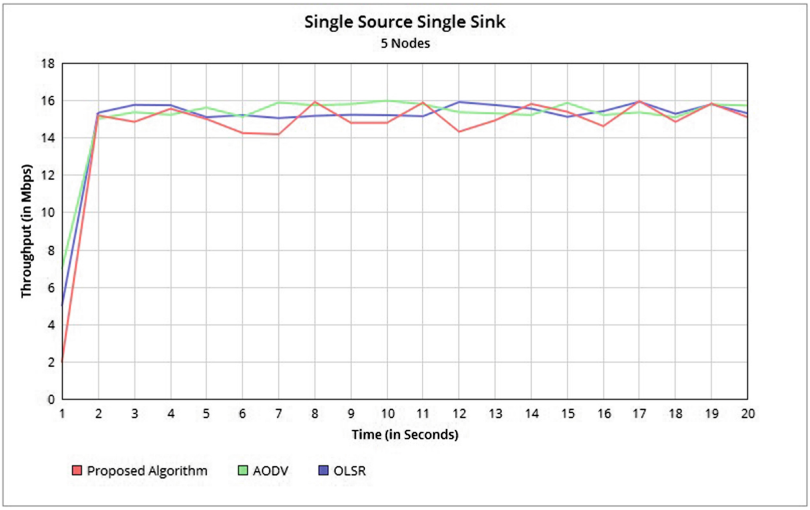


Fig. 4. Results for a single source single sink linear topology

4.3 Multiple Priority Levels

We conducted the same experiment as conducted above with packets of different priority levels to test the packet drop among different priority levels (single source network with linear topology). The program was designed to accommodate 4 UDP packet generators in this scenario producing traffic at 25 Mbps each.

For AODV and OLSR, the results were almost identical to each other as shown in Fig. 5. For AODV and OLSR, the percentage of packets dropped was almost equally distributed amongst the priority levels. The variation is due to the Random Early Detection (RED) queueing mechanism [11]. For the proposed approach, the percentage of priority packets dropped as a percentage of total packets is significantly lower than AODV and OLSR, which is demonstrated in Fig. 6. Instead of dropping higher priority packets, the algorithm dropped the lowest priority packets more often. As shown in Fig. 6, a packet marked as a priority control packet is 20 times less likely to be dropped than a packet marked as a non-priority data packet when the proposed approach is used. This is due to priority differentiation built into the network.

In conclusion, for the single source linear topology, the proposed approach has a comparable throughput but drops less important traffic. This behavior was expected since, as mentioned, there is only a single path and the proposed approach was designed to perform well when multiple paths exist.

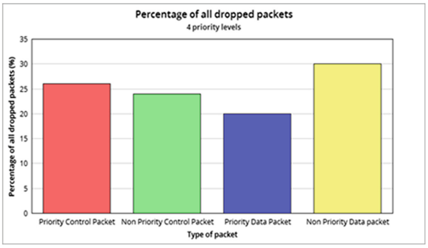


Fig. 5. Dropped packets for each priority level-AODV and OLSR

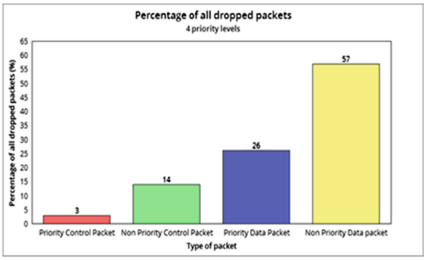


Fig. 6. Dropped packets for each priority level-proposed approach

4.4 Dual Source, 4-Tier Network

Now, let us consider a 4-tier network with two-source UAVs that produce non-priority data packets at the rate of 40 Mbps, control packets at the rate of 10 Mbps and non-priority control packets at the rate of 10 Mbps. An illustration is shown in Fig. 7. A total of 35 UAVs (including the controller) were generated and placed in a grid.

Figure 8 displays the average throughput per second at the sink for both sources combined. On average, the proposed approach provides approximately 15% greater throughput over AODV and OLSR. This is because different packets have different routes. AODV and OLSR have a fixed route for sending all types of packets. The throughput of priority packets however is interesting. We define higher priority packets

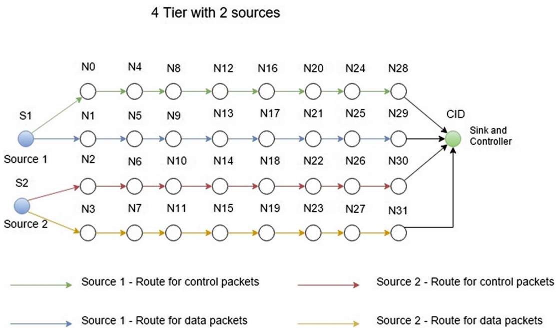


Fig. 7. Experiment with 4 tier network consisting of 2 sources and 1 sink

as any packet with a priority level greater than non-priority data packets. The proposed approach transmitted ~95% of all higher priority packets when compared to at most 55% in AODV and OLSR, as shown in Fig. 9.

Figure 10 shows the average delay of priority packets. Evident fluctuations in delay are due to the Random Early Detection (RED) queuing mechanism in a single queue for all packets when we use AODV and OLSR. However, for the proposed approach, the delay is consistently low with minimal variations. This consistency is due to the refined MAC protocol that prioritizes transmission of priority packets before other packets. A priority control packet is 4 times more likely to be transmitted than a non-priority data packet. For priority control packets, the routes are calculated using Dijkstra's algorithm which guarantees the shortest path to the controller.

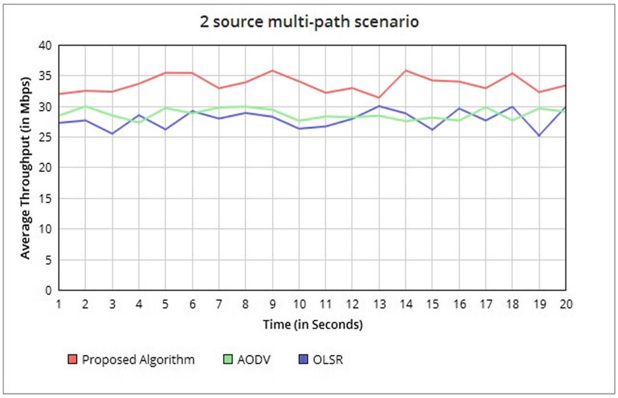


Fig. 8. Combined throughput of sources 1 and 2 (in Mbps)

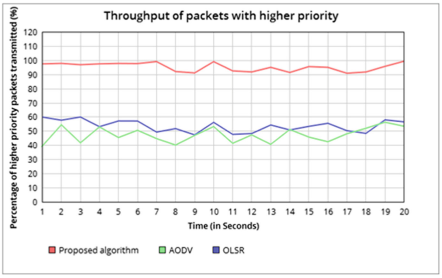


Fig. 9. Priority packets successfully transmitted

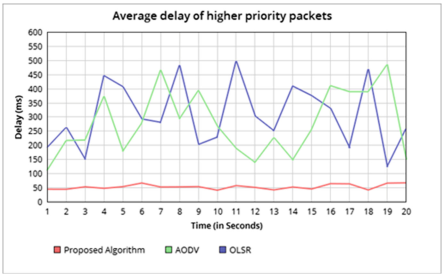


Fig. 10. Average delay of higher priority packets

5 Conclusions

In this paper, we proposed a novel UAV-UAV communication scheme based on the concept of SDN. The UAV backbone is scalable, provides high network efficiency in terms of bandwidth and latency, and supports packets with different priority levels.

The proposed methodology relies on the use of a controller acting as the central hub that monitors all the control information. This hub is used for calculation of routes and to monitor information with regards to the network, which is hard to do in a typical ad hoc network. The routes communicated by the controller provide a means to distribute traffic throughout the network evenly, hence increasing the efficiency of the global network. The important contributions of the proposed approach are as follows:

- Design a more scalable approach for UAV-UAV communication with support for packet prioritization.
- Increase overall throughput of network by evenly distributing traffic throughout the network.
- Find and transmit via faster routes for packets with low delay tolerance, i.e., priority packets. Reduce latency by prioritizing transmission of packets with higher priority.

The simulation results showed that the proposed method provided up to four times as much throughput and reduce latency to less than 1/4 for critical packets compared to AODV and OLSR. High throughput is essential for delivering a jitter free experience for the user and low latency for high priority packets is crucial for maintaining the robustness and stability of the network.

References

1. Perkins, C.E., Belding-Royer, E., Das, S.: Ad hoc On-demand Distance Vector (AODV) routing. RFC 3561 (2003)
2. Clausen, T., Jacquet, P., Adijh, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., Viennot, L.: Optimized Link State Routing Protocol (OLSR). RFC Standard 3626 (2003)

3. Barz, C., et al.: NHDP and OLSRv2 for community networks. In: Proceedings of International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 96–102 (2013)
4. Clausen, T., Hansen, G., Christensen, L., Behrmann, G.: The optimized link state routing protocol, evaluation through experiments and simulation. In: Proceedings of IEEE Symposium on Wireless Personal Mobile Communications, pp. 34–39 (2001)
5. Vara, M., Campo, C.: Cross-layer service discovery mechanism for OLSRv2 mobile ad hoc networks. Sensors **15**(7), 17621–17648 (2015)
6. Royer, E.M., Perkins, C.E.: Multicast operation of the Ad hoc On-demand Distance Vector routing protocol. In: Proceedings of International Conference on Mobile Computer and Network (1999)
7. Zapata, M.G.: Secure Ad hoc On-demand Distance Vector routing. ACM SIGMOBILE Mob. Comput. Commun. Rev. **6**(3), 106–107 (2002)
8. Marina, M.K., Das, S.R.: On-demand multipath distance vector routing in ad hoc networks. In: Proceedings of IEEE 9th International Conference on Network Protocols, pp. 14–23 (2001)
9. Narasimhan, B., Balakrishnan, R.: Energy Efficient Ad hoc On-demand Distance Vector (EE-AODV) routing protocol for mobile ad hoc networks. Int. J. Adv. Res. Comput. Sci. **4** (9) (2013)
10. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Section 24.3: Dijkstra's Algorithm. Introduction to Algorithms, 2nd edn., pp. 595–601. MIT Press, Cambridge (2001)
11. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. Netw. **1**(4), 397–413 (1993)
12. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Can. J. Math. **8**, 399 (1956)
13. Hartley, R.V.L.: Transmission of information. Bell Syst. Tech. J. **7**, 535–563 (1928)
14. Monin, S., Shalimov, A., Smeliansky, R.: Chandelle: smooth and fast WiFi roaming with SDN/OpenFlow. A Poster Presented at the US Ignite (2014)
15. Schulz-Zander, J., Sarrar, N., Schmid, S.: Towards a scalable and near-sighted control plane architecture for WiFi SDNs. In: Proceedings of the 3rd Workshop on HotSDN (2014)
16. Ali-Ahmad, H., et al.: CROWD: an SDN approach for DenseNets. In: Proceedings of the 2nd European Workshop on Software Defined Networks, pp. 25–31 (2013)
17. Bekmezei, I., Sahingoz, O.K., Temel, S.: Flying ad-hoc networks: a survey. Ad Hoc Netw. **11**, 1254–1270 (2013)
18. Shirani, R., St-Hilaire, M., Kunz, T., Zhou, Y., Li, J., Lamont, L.: Combined reactive-geographic routing for unmanned aeronautical ad-hoc networks. In: IWCMC (2012)