Design and Implementation of Adaptive MAC Framework for UAV Ad Hoc Networks

Weijun Wang Chao Dong Hai Wang Anzhou Jiang
College of Communications and Engineering, PLA University of Science and Technology, Nanjing, Jiangsu 210007, China
Email: weijunwang_paper@163.com dch999@gmail.com haiw.wang@gmail.com anzhou_j@163.com
Corresponding Author: haiw.wang@gmail.com

***Abstract*—Due to the agility and low-cost, small Unmanned Aerial Vehicle (UAV) has recently captured great attention of academia and industry. However, since the capability limitation of single device, an ad hoc network formed by small UAVs is very promising. But compared to ordinary ad hoc networks, because of the unmanned characteristic and the diversity of missions, the protocols of UAV ad hoc networks require higher adaptive ability, i.e., the MAC protocol. In this paper, first, we verify that different MAC protocols have respective performance advantage under various network scenarios during the UAV reconnaissance mission. Then, we propose an adaptive MAC framework which allows multiple MAC protocols to switch mutually based on some kind of information you want. After that,in order to demonstrate this framework we design an adaptive MAC protocol called CT-MAC following the proposed framework. CT-MAC allows UAVs to switch between CSMA and TDMA based on their own positions when performing reconnaissance mission. Finally, we implement CT-MAC with Raspberry Pi and MDS Radio. The experiment results show that CT-MAC can always keep desirable performance compared to single MAC protocol through the fast and transparent MAC switching and the proposed adaptive MAC framework is feasible and effective.**

I. Introduction

In recent years, greater attention has been attracted to unmanned systems. Compared to manned ones, unmanned systems (including unmanned aerial vehicles and unmanned vehicles, etc.) can shoulder in-depth investigation of harsher complex environments and even combat missions [1-3]. Furthermore, compared with large unmanned systems such as the US military's MQ-1C "Gray Eagle" unmanned aerial vehicles, small unmanned systems have advantages of lower cost, smaller size and more concealment. Therefore, more attention has been put into the small unmanned systems by academia and industry [4-6], for instance, the "Gremlins" UAVs developed by US military. However, the capability of a single UAV is limited. Coordination and collaboration of multiple UAVs can create a network which is beyond the capability of only one UAV. Consequently, building a network formed by small UAVs is important to guarantee the accomplishment of UAV missions.

Networks formed by UAVs are usually ad hoc networks relying on wireless communication [7]. MAC layer is at the bottom of network protocol stack and its main function is to address the issue of media contention. Thus, the performance of UAV ad hoc networks, i.e., throughput and delay, is related to the employed MAC protocol directly. As the matter of fact, single MAC protocol shows various performance under different network environments and different MAC protocols have respective performance advantage under various network scenarios. Moreover, the UAV application versatility poses vastly varying demands on the design of the aerial network [8]. For instance, a reconnaissance mission consisted of two applications with different requirements of network performance is illustrated by Fig. 1. We named the first application flight phase and the second application data gathering phase. During flight phase, the interaction among UAVs is mainly security information which has high delay but low network traffic requirement. However, during data gathering phase, the network traffic is extremely high because every UAV begins to transmit reconnaissance data to the command base. UAVs should select appropriate MAC protocol under these two different applications adaptively to accomplish the reconnaissance mission. Hence, adaptive MAC protocol, for example, a MAC protocol can fit various demands, is beneficial for UAV ad hoc networks.

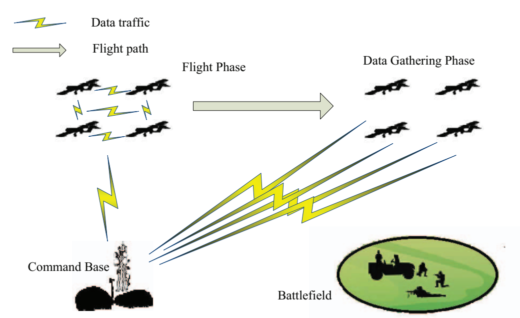


Fig. 1. Reconnaissance mission

Adaptive MAC protocol has been extensively investigated for many years [9-11]. However, the current works mainly focus on the theory and only few of them implement their protocols. In this paper we design and implement an adaptive MAC framework which can combine any MAC protocols and let them switch to each other for UAV ad hoc networks. First, we verify that different MAC protocols have respective performance advantage under various network scenarios in reconnaissance mission by simulation. Then, we propose an adaptive MAC framework which allows multiple MAC protocols to switch mutually based on some kind of information.

978-1-5090-5696-5/16 $31.00 © 2016 IEEE
DOI 10.1109/MSN.2016.38

Next, in order to verify this framework, we design an adaptive MAC protocol called CT-MAC which allows UAVs to switch between CSMA and TDMA based on their own information from GPS. This protocol can be used in UAV ad hoc networks to execute a reconnaissance mission as shown in Fig. 1. Final, we implement CT-MAC with Raspberry Pi and MDS Radio and evaluate its performance.

In summary, the main contributions in this paper are:

1. We propose an adaptive MAC framework for UAV ad hoc networks. Under this framework UAVs can choose the most appropriate MAC protocol and switch to it based on some kind of information you want.

2. We design an adaptive MAC protocol called CT-MAC following the proposed framework. CT-MAC allows UAVs to switch between CSMA and TDMA with their own GPS location information. We implement CT-MAC with Raspberry Pi and MDS radio. Especially to deserve to be mentioned, CT-MAC is just an example to demonstrate whether the adaptive MAC framework can work well or not and you can combine any numbers of any MAC protocols with this framework not just CSMA and TDMA.

3. We run experiments to evaluate the performance of CT-MAC and verify the practicability of the proposed framework. The experiment results show that CT-MAC can always keep desirable performance compared to single MAC protocol through the fast and transparent MAC switching and the proposed adaptive MAC framework is feasible and effective.

The remaining sections of this paper are constructed as follows. In section II, we introduce the related works of adaptive MAC protocol. Section III compares the performance of different MAC protocols under various scenarios by simulation. The adaptive MAC framework and details about implementation of CT-MAC are discussed in section IV and V. In section VI, the experimentation and the results are given. Finally, section VI concludes the paper.

II. Related Works

Over the past several years there has been increasing interest on adaptive MAC protocol. In METAMAC [10], A. Farago et al. first present a systematic and automatic method which can dynamically combine any existing MAC protocol into a single layer. In MULTIMAC [12], C. Doerr et al. propose a framework and theoretically discuss how to implement the experimental platform for evaluating algorithms that dynamically reconfigure MAC and physical layer properties. At last, researchers implement a peer-to-peer network, but not a practical protocol, to verify the efficacy of MULTIMAC. A. Jamali et al. propose a novel MAC protocol which let every node dynamically adjusts its backoff window size according to the current network status in [13]. Prior research just present frameworks and verified them through theoretical analysis or simulation, but lack experiment.

From another point of view to look at this issue, there are a large amount of works have been proposed before on how to hybrid the CSMA and TDMA for sensor network. For example, ADAPT [14], Z-MAC [15], et al, combine the strengths of TDMA and CSMA. However, these works just concerned about these two MAC protocols, TDMA and CSMA. Our framework can combine any numbers of any MAC protocols and switches between them to get the best performance in various application. In other words, you can even switch between Z-MAC and ADAPT if you want. In section V, we combine TDMA and CSMA as an example to verify if our framework can work well, but the combination is not limited to this example.

The most related work is [16]. Kuo-Chun Huang and D. Raychaudhuri present and implement an adaptive MAC protocol in cognitive radio networks. This protocol is based on the "CogNet" protocol stack [17] which uses a Global Control Plane (GCP) to distribute control information among nearby nodes. The noticeable differences from our work are presented as follows. Since this protocol controls all nodes in the network by GCP, it needs an dedicated channel to exchange control information. Therefore, this is a centralized protocol and researchers use dual-channel devices to implement it. However, our proposed protocol is a distributed one and just needs single-channel devices to be implemented.

III. Performance Comparison of Different MAC Protocols

As we all know, researchers have proposed a number of MAC protocols for wireless network. According to the mode of channel access, current MAC protocols can be divided into two types: random access protocols and controlled access protocols, of which CSMA and TDMA are typical ones respectively. Nodes with CSMA transmit data by channel competition. In this way, when network load is low, using CSMA protocol can achieve higher channel utilization and lower latency [18]. Unlike CSMA, TDMA assigns fixed slot to each node to avoid contention. Consequently, TDMA protocol can achieve higher utilization in high load network [19][20].

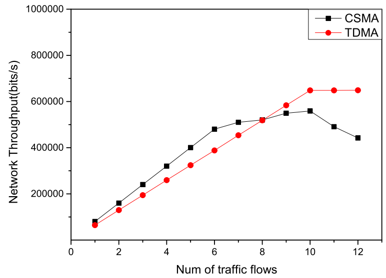
There are some results of different performance of various MAC protocols in previous works, but those works compare different MAC protocols under their own applications. In this section, we take CSMA and TDMA as examples, using QualNet [21], a simulator developed by Scalable Networks Technologies, to compare the different performance of these two kinds of MAC protocols in UAV reconnaissance mission. Thus we can be aware of if the MAC protocols need to be switch to each other according to the simulation result.

A. Simulation Setup

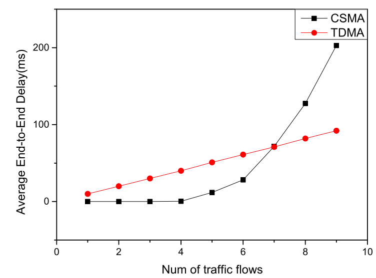
TABLE I
Simulation environment

Communication range (meters)	500
Simulation region (m^2)	1500*1500
Number of nodes (N)	10
Frame data length (Bytes)	900
Slot of TDMA (milliseconds)	10
Physical layer throughput (Mbps)	1

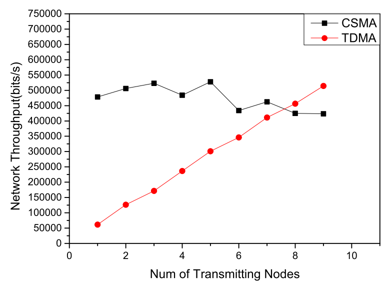
Consider an UAV ad hoc network with 10 UAVs. We consider the 10 UAVs as 10 nodes. For universality, the 10



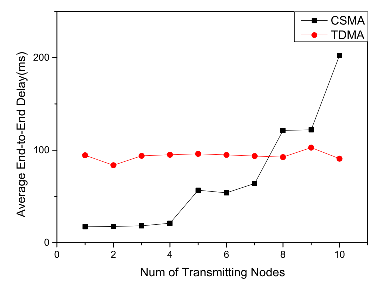
(a) Throughput under different network load



(b) Delay under different network load



(c) Throughput under different flow distribution



(d) Delay under different flow distribution

Fig. 2. Comparison of CSMA and TDMA in different network environments

nodes are distributed randomly in 1500m * 1500m area, the communication range of each node is 500m and the channel rate is set to 1M bit/s. The packet length is 900 bytes. Other major configurations and protocol parameters are summarized in Table I. Due to during the whole reconnaissance mission, UAVs form a formation, the relative position of UAVs are constant. Thus in our simulation, the topology of the UAV ad hoc network is fixed.

During the flight phase, UAVs only send few security information to command base and receive some commands from command base. So in this phase, the requirement of average network delay is high because the commands and security information is latency-sensitive. But as most of the security information and commands are short frames and almost at anytime the traffic just happens between a few UAVs and command base, the requirement of network traffic is relatively low. However, during the data gathering phase, all UAVs transmit data at the same time, so the network traffic is very high. Because of the different network characteristics in the two phases during the reconnaissance mission, we intend to verify the performance of TDMA and CSMA in two aspects as follows.

Simulation I: Fix the traffic distribution and change sum of traffic. The objective of this simulation is to compare the performance of delay and throughput of both protocols under different traffic load. We add traffic flows to each node uniformly. For example, we add 12 flows to network, then, we will guarantee 8 nodes respectively transmit 1 flow and 2 nodes respectively transmit 2 flows.

Simulation II: Fix the sum of traffic and change traffic distribution. The objective of this simulation is to compare the performance of delay and throughput of both protocols under different traffic distribution. We add 9 traffic flows to network, for example, 1 node transmits all flows or 1 node transmits 7 flows and another node transmit 2 flows, etc.

B. Simulation Results

Results of Simulation I are shown in Fig. 2(a)(b). As illustrated by Fig. 2(a), as more of traffic flows are generated, both the throughput of CSMA and TDMA gradually increase. Even though, CSMA offers marginally higher throughput at the beginning (as evidenced for the number of traffic flows less than 8), the performance differences for number of flows larger than 9 highlights TDMA is more appropriate in high-contention network. In Fig. 2(b), first as expected, the end-to-end delay of CSMA is better than TDMA. However, as the number of traffic flows increases, the end-to-end delay increases significantly for CSMA because contention occur in the network.

Fig. 2(c)(d) illustrate the results of Simulation II. In this simulation, 9 traffic flows are distributed to 10 nodes, where every node can get 0 to 9 flows. As shown in Fig. 2(c), at the beginning, CSMA offers much higher throughput. However, as the number of transmitting nodes increases, the throughput of CSMA degrades while the throughput of TDMA increases steadily. The throughput of TDMA is more than that of CSMA for 9 transmitting nodes. In addition, the end-to-end delay is shown in Fig. 2(d). As illustrate by the figure, the end-to-end delay of TDMA is nearly independent of the number of transmitting nodes because there is no collisions occur, but of CSMA is increasing as the number of transmitting nodes increases.

Through extensive simulation we find that TDMA protocol works well in high-contention network environment, i.e., the data gathering phase, and CSMA protocol fits low-contention environment, i.e., the flight phase. Furthermore, results in simulation II, as Fig. 2(b)(d), show that TDMA is more suitable in the condition of homogeneous distribution of network traffic. In contrast, when the traffic flows are distributed on a few nodes, CSMA shows better channel utilization.

IV. Adaptive MAC Framework

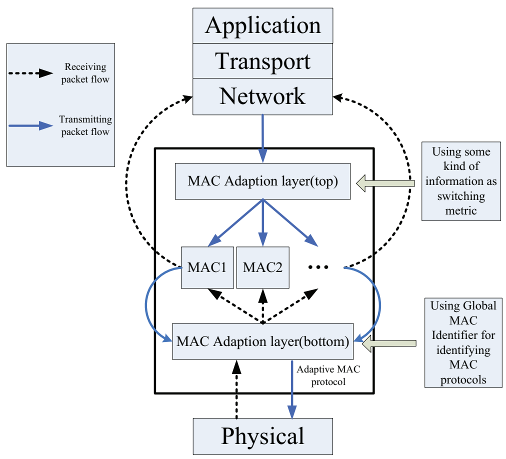


Fig. 3. Adaptive MAC framework

In this section, we present an adaptive MAC framework which can guarantee to use the appropriate MAC protocol under specific network conditions. It respectively adds MAC adaptation layer above and below the existing MAC layer. The MAC adaption layer can select the appropriate MAC protocol and allow multiple MAC protocols to switch mutually based on some kind of information.

With the help of MAC adaption layers, sending and receiving traffic flows are independent to each other as shown in Fig. 3. In order to distinguish the receiving frames sent by different MAC protocols, we add 1-byte Global MAC Identifier (GMI) to the header of the frame for identifying MAC protocols. We will discuss the advantage of GMI later in this section.

At the sending end, when top MAC adaption layer receives a packet from network layer, before passing the packet to MAC layer, it will select suitable MAC protocol according to the predefined metric. After the MAC protocol is selected, the top MAC adaption layer will pass the packet to the selected MAC protocol for encapsulation. Then, the MAC protocol will pass the frame to the bottom MAC adaption layer to add the corresponding GMI to the header of the frame. Finally, the frame with GMI will be passed to the physical layer.

At the receiving end, when bottom MAC adaption layer receives a frame from the physical layer, it will first read the GMI to identify the MAC protocol that should be employed. Then cut the GMI and pass the frame to the appropriate MAC protocol. The following process happens the same way as in an unmodified network stack. It deserves to be mentioned that the meaning of GMI here is different from the sending end. At the sending end, GMI marks the current MAC protocol used to send frames, while it marks the MAC protocol used to decode receiving frames here.

Fig. 4. Frame format of adaptive MAC protocol

The frame format is shown in Fig.4. Adding GMI to the header of frames can avoid conflicts among competing MAC layers, and also ensure that the nodes which use different configurations (and therefore MACs) for transmitting frames are still able to interpret and decode incoming frames from other nodes encoding with different MACs. Because at receiving end, bottom MAC adaption layer will pass the receiving frame to the corresponding MAC according to the GMI.

To better understand why GMI can ensure the communication among the nodes using different MACs, we will analysis a system with two nodes, one uses CSMA and the other uses TDMA.

Case I: the sending end using CSMA and the receiving end using TDMA. In this case, when the receiving end receives RTS, this node will return CTS with GMI is equal to CSMA at its TDMA slot. Similarly, after the sending end receives CTS it will transmit data with GMI is equal to CSMA. Then, when the receiving node receives the data, it will send an ACK with GMI equal to CSMA at its TDMA slot too.

Case II: the sending end using TDMA and the receiving end using CSMA. In this case, the sending end transmit data with GMI is equal to TDMA in its slot. When the receiving end receive the frames, it read the GMI is TDMA, then the GMI will be cut and transfer to TDMA protocol and TDMA decode the frame and pass to the upper layer.

Our adaptive MAC framework can guarantee the communication of the nodes using different MAC protocols without packets loss, but the conflict will also occur if the sending nodes using different MAC protocols and we will discuss this issue in section VI.

V. CT-MAC

In order to verify the adaptive MAC framework whether can switch MAC protocols mutually or not, in this section, we design and implement an adaptive MAC protocol as an example, called CT-MAC following the proposed framework, to execute the reconnaissance mission in Fig. 1. CT-MAC allows UAVs to switch MAC protocols between CSMA and TDMA to achieve desirable performance throughout the mission. In order to implement CT-MAC, overall, there are three issues we need to address:

- What information we use to switch the MAC protocol?

- How to assign slots for TDMA?
- How to save the wireless spectrum?

1) Using GPS location information for MAC switching

There are a number of metrics to select the appropriate MAC, such as queue length, bit-error frames and traffic load, etc. In this paper, we utilize GPS location information to indicate MAC switching. First, considering different requirement of two phases in reconnaissance mission, we use CSMA and TDMA in flight phase and data gathering phase respectively. Moreover, due to the different cruising range of UAVs and AEW (Air Early Warning), to reconnaissance mission UAVs is appropriate to monitor the battlefield in the air after the war has begun, while AEW is more appropriate to search the enemy. Thus before reconnaissance we have known the battlefield location, setting the time when the UAVs reach the battlefield location to be MAC switching time is practical. Second, using GPS location information each UAV can switch MAC protocol independently. In this way, there will not be lots of MAC switching information occupying the channel. Therefore, using GPS information is simple and effective.

2) Assigning slots for TDMA with GPS time information

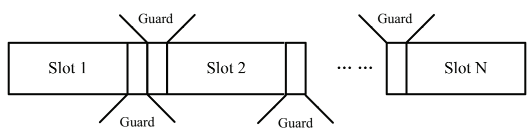


Fig. 5. TDMA Slot allocation

TDMA assigns fixed slots to individual nodes and every node can only transmit data during their assigned slots. Therefore, slot allocation is critical to TDMA. Using GPS time information can avoid exchanging clock signal and guarantee all nodes have the same timing accuracy as the GPS signal is 1 pulse per second. The TDMA slot allocation is shown in Fig.5. Slot 1, Slot 2, up to Slot N are the slots allocated to N nodes, and the guards between the slots are used to avoid the conflict which can be set with longest packet transmit time. In addition, our goal is to demonstrate adaptive MAC framework if can switch MAC mutually through CT-MAC. Considering that in the MAC switching mechanism, we have used GPS location information to indicate the MAC switching using GPS time information to assign slots is efficient and convenient.

3) Reformatting RTS,CTS and ACK frame

1 byte	1 byte	1 byte	1 byte
Destination node No.	Source node No.	Message type	QoS

Fig. 6. RTS CTS and ACK formation

CSMA protocol uses RTS-CTS handshake procedure to alleviate the hidden and the exposed terminal problems. Using ACK mechanism CSMA can ensure there will not be collisions during sending end transferring data. Because the number of nodes in UAV ad hoc networks is quite limited, we do not need to use 6-byte MAC address as the IEEE 802.11 DCF which is a typical CSMA-like protocol. CT-MAC uses one-byte mapping for each node to replace the ordinary 6-byte MAC address and reformats the RTS, CTS and ACK frame with just 4 bytes. The RTS, CTS and ACK formation is shown in Fig. 6. We use 2 bytes to be the addresses of destination node and source node, 1 byte to be the message type and 1 byte to label the QoS which is used to distinguish whether the transmission requires ACK or not.

The CT-MAC is executed as follows. After the UAVs take off from command base, all UAVs use CSMA to exchange information, for instance, security information, etc, with each other. When the UAVs reach the battlefield location, each UAV will switch its MAC protocol to TDMA according to its own GPS information and begin to transfer reconnaissance data.

VI. Performance Evaluation

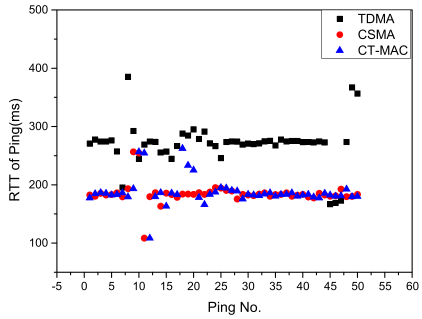
Fig. 7. Communication node in UAV

In order to evaluate the performance of the proposed framework and CT-MAC protocol, we utilize Raspberry Pi, MDS Radio and GPS device to implement a communication node in UAV as shown in FIG. 7. First, we realize the protocol in the Linux operating system over a Raspberry Pi. Then, we connect a MDS radio to the Raspberry Pi with a RS232 to USB cable. Finally, we connect a GPS device to the Raspberry Pi with a USB cable to get GPS signal.

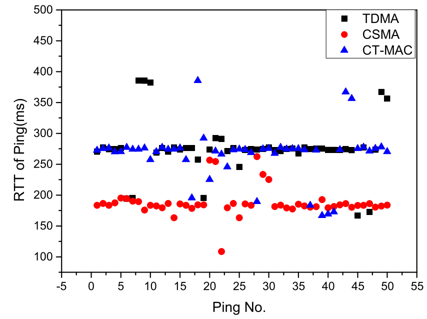
Fig. 8. UAV equipmented with Communication node

A. Experiment Setup

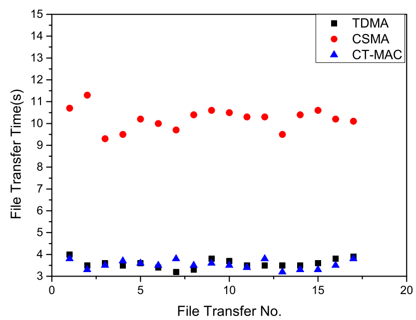
Our experiment uses 3 UAVs installed communication nodes whose configurations are all the same except IP address and MAC address to evaluate CT-MAC in the open air. Fig. 8 shows one of the three fully equipped experiment UAVs. To model real-world connection, our experiment consider one UAV as command base and the other two UAVs send data



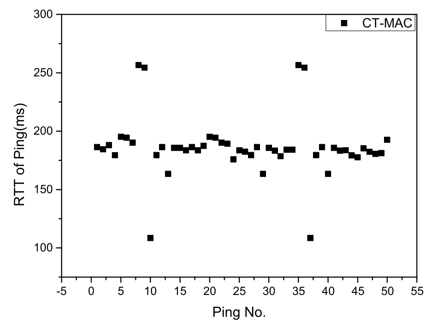
(a) Ping in flight phase(ms)



(b) Ping in data gathering phase(ms)



(c) Time of file transport(s)



(d) Ping in MAC switch(ms)

Fig. 9. Ping time and time of File Transfer with each MAC protocol

TABLE II
TIME OF DIFFERENT APPLICATIONS IN DIFFERENT PROTOCOLS

Protocol	Ping in flight phase(ms)	Ping in MAC switch(ms)	Ping / Throughput in data gathering phase(ms/Kbps)	Time of file transport(s)
CT-MAC	183	186	281 / 6.18	3.64
TDMA	273	N/A	279 / 6.18	3.63
CSMA	182	N/A	188 / 1.79	10.34

to command base simultaneously. Moreover, we consider a setting area as battlefield.

In our experiment, we use ”Ping” to simulate the security information because most of these security frames are short and pay more attention on delay. We also use the file transfer to model the reconnaissance data.In addition, in order to analyse the performance of the framework and the efficacy and practicability of MAC switching, we use Round-Trip Time (RTT) of ”Ping”, the time cost of transferring the same file as metrics.

In experiment, we intend to evaluate the adaptive MAC framework in two aspects:

Experiment I: Execute the reconnaissance mission with CT-MAC, CSMA and TDMA protocol respectively to evaluate the performance of CT-MAC protocol. The experiment process with CT-MAC, CSMA and TDMA respectively is the same. First, we fly the three UAVs together in 5-meter distance. Then, when arriving the setting area, the three UAVs switch(with CT-MAC) or do not switch(with CSMA and TDMA) the MAC protocol and begin to transfer the file. In addition, for more precise conclusions, we also use Iperf [22], an open source network throughput test software, to test the network throughput in the data gathering phase. We do not need to test the network throughput in the flight phase, because traffic in this phase is very low according to the application.

Experiment II: ”Ping” in the network with two different MACs simultaneously to evaluate the efficacy and practicability of MAC switching. In reconnaissance mission, when UAVs reach the air of battlefield, CT-MAC will switch MAC protocol to TDMA immediately and the UAVs transfer reconnaissance data to command base. Therefore, a UAV sending file with CSMA is impossible, but a UAV with CSMA ”Ping” another UAV maybe occur. So, we use a node with CSMA to ”Ping” another node with TDMA and record the RTT of ”Ping”.

B. Experiment Result

Fig. 9 presents the RTT of "Ping" and the file transfer time in experiments. Table II shows the average time of corresponding RTT of "Ping", average time of file transfer and network throughput in data gathering phase.

The results of Experiment I are shown in Fig. 9(a)(b)(c). As shown in Fig. 9(a), in flight phase RTT of CT-MAC is the same as RTT of CSMA which is better than TDMA. However, compared to CSMA, Fig. 9(c) illustrates CT-MAC and TDMA both cost less time in file transfer. In Fig. 9(b), RTT of CT-MAC is the same as RTT of TDMA because the current MAC of CT-MAC has switched to TDMA already. However, we don't need to worry, because file transfer is the main application in data gathering phase. Moreover, in this phase, in order to execute reconnaissance mission, UAVs have a fair chance of hovering over the battlefield and the positions of UAVs are relatively fixed. Therefore, UAVs do not need to exchange much security information. In addition, according to the network throughput in Table II, although the network throughput is just some Kbps due to the restriction of MDS radio and serial port, we can also get the message that TDMA and CT-MAC are much more appropriate than CSMA in data gathering phase. In general, CT-MAC shows high performance both in low-contention and high-contention network environment.

The results of Experiment II can be seen in Fig. 9(d). As illustrated by the figure, RTT of CT-MAC maintains low latency as CSMA during MAC switching phase. Consequently, the MAC switching in CT-MAC is effective and CT-MAC can guarantee communication of UAVs with different MACs simultaneously. Of course, if the GPS signal is accurate and correct, there will not be different protocols in the network.

In summary, CT-MAC inherits the advantages from TDMA and CSMA and avoids their shortcomings. The adaptive MAC framework can switch MAC swiftly and guarantee communication in the network with different MAC protocols simultaneously. Therefore, although CT-MAC is just an example, according to its satisfactory performance it also demonstrate that following the proposed adaptive framework the implemented adaptive MAC protocol can always achieve desirable performance under various network conditions.

VII. Conclusion and Future work

In this paper, we propose an adaptive MAC framework for UAV ad hoc networks. Following this framework, we design an adaptive MAC protocol called CT-MAC as an example which allow UAVs to switch between CSMA and TDMA based on their own information from GPS to demonstrate the adaptive MAC framework. Then, we implement CT-MAC with Raspberry Pi and MDS Radio. The experiment results of CT-MAC demonstrate that following adaptive MAC framework the implemented MAC protocol has better performance than single MAC protocol through the fast and transparent MAC switching. However, as mentioned in section IV, the electromagnetic environment in the real-world is various and the GPS signal may not be accurate, so the collision may occur due to that UAVs use different MAC protocols to transmit packets. In the near future, we will study how to handle this instable state.

VIII. Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant No.61472445, No.61371124, No.61571463, in part by the Natural Science Foundation of Jiangsu Province under Grant No.BK20140076.

References

[1] J. George, P.B. Sujit, J. Sousa, *Search strategies for multiple UAV search and destroy missions*, Journal of Intelligent and Robotics Systems, 2011, pp. 355 - 367.

[2] Z. Sun, P. Wang, M.C. Vuran, M. Al-Rodhaan, A. Al-Dhelaan, I.F. Akyildiz, *BorderSense: border patrol through advanced wireless sensor networks*, Ad Hoc Networks, 2011, pp. 468 C 477.

[3] H. Xiang, L. Tian, *Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle*, Biosystems Engineering, 2011, pp. 174 C 190.

[4] E. P. De Freitas, T. Heimfarth, A. Vinel, F, etc, *Cooperation among wirelessly connected static and mobile sensor nodes for surveillance applications*, Sensors, 2013, pp. 12903 - 12928.

[5] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, *Mobile Internet of Things: Can UAVs provide an energy-efficient mobile architecture?*, IEEE Global Communications Conference (GLOBECOM), 2016.

[6] L. To, A. Bati, D. Hilliard, *Radar cross-section measurements of small unmanned air vehicle systems in non-cooperative field environments*, 3rd European Conference on Antennas and Propagation, 2009, pp. 3637 C 3641.

[7] I. Bekmezci, et al., *Flying Ad-Hoc Networks (FANETs): A survey*, Ad Hoc Netw, 2013.

[8] Samira Hayat, Evsen Yanmaz, Raheeb Muzaffar, *Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint*, Communications Surveys & Tutorials, 2016, pp. 1 - 38.

[9] K.-C. Huang, X. Jing, and D. Raychaudhuri,*Mac protocol adaptation in cognitive radio networks: An experimental study*, ICCCN, 2009.

[10] Weihong Hu, Xiaolong Li, Homayoun, Yousefizadeh, *LA-MAC: A Load Adaptive MAC Protocol for MANETs*, GLOBECOM, 2009.

[11] A. Farago, A. Myers, V. Syrotiuk, and G. Zaruba, *Meta-mac protocols: Automatic combination of mac protocols to optimize performance for unknown conditions*, IEEE JSAC, special issue on: Analysis and Synthesis of MAC Protocols, 2000, pp. 1670 - 1682.

[12] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, D. Grunwald *MultiMAC - an adaptive MAC framework for dynamic radio networking*,New Frontiers in Dynamic Spectrum Access Networks, 2005, pp. 548 - 555.

[13] Amin Jamali, Seyed Mostafa Safavi Hemami, Mehdi Berenjkoub, Hossein Saidi, *An Adaptive MAC Protocol for Wireless LANs*, Journal of Communications and Networks, 2014, pp. 311 - 321.

[14] Injong Rhee, Ajit Warrier, Mahesh Aia, Jeongki Min, Mihail L. Sichitiu, *Z-MAC: A Hybrid MAC for Wireless Sensor Networks*, TRANSACTIONS ON NETWORKING, 2008, pp. 511 - 524.

[15] Injong Rhee, Ajit Warrier, Jeongki Min, Lisong Xu, *DRAND: Distributed Randomized TDMA Scheduling For Wireless Adhoc Networks*, MobiHoc, 2006, pp. 1384 - 1396.

[16] Kuo-Chun Huang and D. Raychaudhuri, *MAC protocol adaptation in cognitive radio networks*, WCNC, 2011.

[17] Xiangpeng Jing and Dipankar Raychaudhuri, *Global control plane architecture for cognitive radio networks*, ICC, 2007, pp. 6466 C 6470.

[18] I. Chlamtec, A. Farago, A. D. Myers,etc. , *A performance comparison of hybrid and conventional MAC protocols for wireless networks*, VTC, 2000, pp. 201 - 205.

[19] Skander Banaouas and Paul Mhlethaler, *Performance evaluation of TDMA versus CSMA-based protocols in SINR models*, Wireless Conference, 2009, pp. 113 - 117.

[20] Katrin Sjoberg, Elisabeth Uhlemann, Erik G. Strom, *Delay and interference comparison of CSMA and self-organizing TDMA when used in VANETs*, International Wireless Communications and Mobile Computing Conference, 2011, pp. 1488 - 1493.

[21] *http://web.scalable-networks.com*

[22] *https://iperf.fr/*