Enabling Multi-packet Transmission and Reception: An Adaptive MAC Protocol for MANETs

Hui Xu[1,*], J.J. Garcia-Luna-Aceves[1], and Hamid R. Sadjadpour[2]

[1] Computer Engineering Department
[2] Electrical Engineering Department
University of California, Santa Cruz
Santa Cruz, CA 95064, USA
`{xuhui,jj,hamid}@soe.ucsc.edu`

Abstract. To increase network capacity, advanced physical layer (PHY) techniques have been developed to support new transmission paradigm where one node could send different packets concurrently to multiple receivers (multi-packet transmission: MPT) or receive packets concurrently from multiple senders (multi-packet reception: MPR). To exploit and support them for high performance, new type of medium access control (MAC) protocol is needed. Especially, MPT or MPR requirements are dynamic according to traffic conditions. In this paper, an adaptive MAC approach (AMPTR) is proposed to enable dynamic MPT or MPR requirements for mobile ad hoc networks (MANETs). The proposal includes two main parts: access coordination and data transmission. The access coordination process comprises channel access contention and coordination to make handshakes for multiple concurrent transmissions. Once channel access coordination is completed, multiple transmissions can then be carried out concurrently, where frame aggregation is used for network capacity and throughput improvement and accordingly block acknowledgement is employed for efficiently reporting multiple packet reception status. We evaluate the performance of AMPTR through simulations, and show that the AMPTR scheme has much higher network throughput and smaller packet delivery delay than currently widely used multiple access preventing schemes.

---

[*] This research was partially sponsored by the U.S. Army Research Laboratory under the Network Science Collaborative Technology Alliance, Agreement Number W911NF-09-0053, by the Army Research Office under agreement number W911NF-05-1-0246, by the National Science Foundation under grant CCF-0729230, and by the Baskin Chair of Computer Engineering. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

X. Zhang and D. Qiao (Eds.): QShine 2010, LNICST 74, pp. 188–203, 2011.

1 Introduction

In mobile ad hoc networks (MANETs), there are two directions to increase network throughput: one is to increase the number of concurrent transmissions; the other is to increase throughput for each transmission. Much work has focused on developing advanced physical layer techniques to support multiple concurrent transmissions while providing more reliable data transmission. For example, multiple-input multiple-output (MIMO) uses multiple antennas at the wireless transmitter and receiver, and exploits the spatial dimension and multi-path channel signals to enable the ability of the transmitter transmitting multiple data streams to the receiver concurrently and the receiver decoding more than one data stream with higher success probability [1, 2].

When data streams aim for different receivers (multi-packet transmission: MPT) or come from different transmitters (multi-packet reception: MPR), coordinations among nodes are needed [3], and the effectiveness of multiple concurrent transmissions (such as MPT or MPR) on improving network capacity is theoretically analyzed and proved in [4]. These kinds of MPT and MPR transmission paradigms attract more interest and have not been fully exploited. There are many MAC protocols proposed to exploit the advanced physical ability of MPR and allow multiple transmitters to concurrently access medium/channel [5,6,7,3]. Particularly, [8,3] have tried to modify IEEE 802.11 DCF scheme [10] to support MPR by negotiating concurrent transmissions with extra bytes in handshake control packets (such as RTS, CTS and ACK). However, all the above proposals use fixed transmission architecture, i.e., supporting MPT or MPR while not both concurrently; in addition, they neglect traffic conditions and network layer transmission requirements. While in practice, a node may want to transmit to multiple receivers or receive multiple packets from multiple transmitters concurrently. One unexploited multiuser transmission direction is to adaptively support MPT or MPR, which is exploited in our proposal AMPTR.

In this paper, we aim to provide an adaptive MAC protocol which can provide medium access according to traffic flow requirements. When a node which obtains access permission has packets for multiple receivers, MPT will be carried out; otherwise, the node will give right to its receiver to initiate more neighbors transmitting to it concurrently, i.e., MPR will be triggered. The adaptability and compatibility of this proposal is also represented in that when MPT or MPR handshake fails, normal peer-to-peer transmission could still be carried out. Our design principle is that each node performs Carrier Sensing Medium Access (CSMA) scheme [10], and that once channel access is permitted, the maximum number of streams the channel could support will be exploited, i.e., fulfill the channel by scheduling that number of concurrent transmissions. The process includes two parts: access coordination and data transmission. The access coordination is combined with both contention and coordination, where intended transmitters follow CSMA scheme to compete channel, and use handshakes to coordinate its neighbors' behaviors. Once coordination is complete, the transmitter or multiple transmitters could access the channel concurrently and multiple transmission could be completed.

The rest of this paper is organized as follows: In Section 2, we present our adaptive MPT and MPR medium access control mechanism in detail. Then, Section 3 gives some experimental results and validates the effectiveness of the proposed scheme in greatly improving throughput and reducing end-to-end delay. We finally conclude in Section 4.

2 Adaptive Multiple Packet Transmission and Reception (AMPTR)

2.1 Scheme Overview

Our scheme includes four phases which is shown in Fig. 1. They are access contention, access coordination, data transfer and block acknowledgement. During access contention period, any node with packets in transmission queue is competing the access to channel using carrier sensing like that in 802.11 DCF; once a node gets the access to channel, transmission handshakes among multi-users which try to trigger M (the maximum number of supported concurrent transmissions) different data streams concurrently will be carried out, that is access coordination phase; after coordination is done, multiple concurrent data transmissions with frame aggregation scheme for transmitting multiple packets together can then be carried out; after that, receivers should send block acknowledgment back to transmitters to report the reception condition for each packet since frame aggregation is carried out.

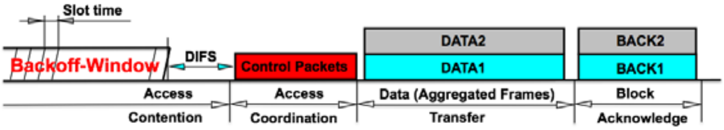


Fig. 1. The AMPTR scheme

We will present each phase of AMPTR in detail in the following subsections. To make our presentation easier, without loss of generality, we assume that $M = 2$, that is, each node can at most support two concurrent transmissions, and is half-duplex, i.e., can only transmit or receive at one time.

2.2 Access Contention

Assume that a node S has packets ready in its transmission queue, similarly to the IEEE 802.11 DCF scheme, it first listens the channel signal. When the perceived power is lower than a threshold for a period of DIFS, node S regards the channel idle. After the medium is sensed idle for a time interval (a distributed interframe space, i.e., DIFS), node S thinks the channel available. Otherwise, the node needs to defer transmission attempt with a random backoff timer. Its value is uniformly distributed in $[0, CW]$, where CW stands for contention window and is initially set to CW_{min}, and doubled after each time the transmission incurs a transmission failure and schedules a retransmission, up to its maximum value CW_{max}. The backoff timer is suspended whenever the channel becomes busy, and reactivated after the channel is sensed idle for a time interval (a distributed interframe space or DIFS) and decremented by one for each physical slot time. The node starts to make handshakes for transmission when its backoff timer reaches zero.

The big difference between AMPTR and IEEE 802.11 DCF scheme is that in AMPTR there is one special case to decrease the value of backoff timer and therefore to increase the transmission chance. When a receiver tries to trigger one more neighbor node to transmit packets to it, if a node has packet for the receiver and wins the transmission chance among multiple nodes which also have packets for the receiver and are also ready for transmissions, its backoff timer can then be decreased directly to zero.

2.3 Access Coordination

Assume that each node has two additional queues for storing unicast packets and that only packets with the same next hop destination could be stored in the same queue, that is, each node has two separate queues for two different destinations. Whenever there are arrival network layer packets, the node will try to dequeue them into the two separate queues until meet broadcast packets or unicast packets aiming for the third next hop destination.

Given that a node's backoff timer decreases step by step into zero, i.e., it wins the channel access by carrier sensing and backing off, it will again try to dequeue packets in the transmission queue into the two separate queues as many as possible, and then check whether there are unicast packets in the two separate queues. If the two queues are both empty, there must be a broadcast packet in the transmission queue, and the node can just send it out. Otherwise, handshakes before transmitting data should be carried out to coordinate the transmission behaviors among multiple users.

Our scheme is adaptive with traffic conditions and could dynamically support MPT or MPR; it is also compatible since when multiple concurrent transmissions could not be carried out, the normal point-to-point packet transmission could still be executed.

Suppose that a node S is ready for making handshakes, if its two separate queues both are not empty, i.e., there are packets ready for transmission to two different receivers, denoted by A and B, efficient handshakes among node S, A and B for MPT can then be carried out: node S first sends one Request to Send (RTS) packet to both A and B with the indication of response order for them to avoid response packet collision; when node S receives response packets of Clear to Send (CTS) from both of them successfully, it can then send packets concurrently to both A and B, i.e., MPT can then be carried out. The sketch of our scheme about how to support MPT process by making handshakes among one transmitter and two receivers is shown in Fig. 2.

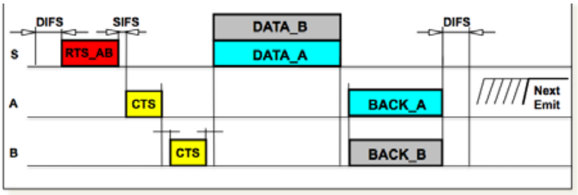


Fig. 2. The sketch of MPT process

If only one of two separate queues is not empty, i.e., there are packets ready for transmission only to one receiver of node A, node S will send a Request to Send (RTS) packet to A with the indication for the receiver A to take over the right of triggering one more transmission to it. The receiver A then broadcasts out a Ready to Receive (RTR) packet to all its neighbors, and any node U which receives the RTR packet and has packets ready for transmission to the receiver A will first make physical carrier sensing. After the channel is sensed idle for a short Interframe space (SIFS), node U will send a Reply to RTR (RRTR) packet to inform the receiver A that it is ready to be the another transmitter, and at the same time decrease its own backoff timer directly to zero for channel access. Since those nodes (excluding the first transmitter S) which have packets ready for transmission to the receiver A start physical carrier sensing for a period of SIFS, the node U which first receives RTR will send RRTR first, and at the same time other nodes will sense the ongoing RRTR transmission and stop their attempts for sending RRTR, i.e., the RRTR from node U could be successfully received by the receiver A. Finally, the receiver A could send one CTS packet to both node S and U, and the two transmitters can then send packets concurrently to the receiver A, i.e., MPR can then be carried out. The sketch of supporting MPR process by making handshakes among two transmitters and one receiver is shown in Fig. 3.

However, in MPT case it is possible that the CTS packets from receivers could not be successfully received or that only one of the CTS packets could be successfully received. For the first situation, the RTS packet could be retransmitted with the increment of Short Retransmission Counter (SRC) by one; for the second situation (e.g., the transmitter S does not receive the CTS packet from node B), a new RTS only for node B will be sent out also with the increment of

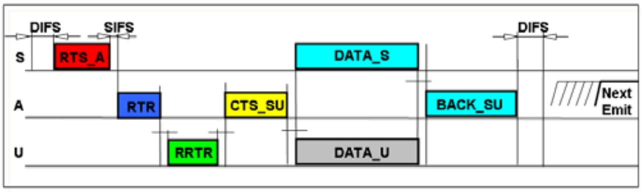


Fig. 3. The sketch of MPR process

SRC by one. The RTS packet keeps on being transmitted until one of the events occur: the handshake for MPT successes, the timer for handshake is out, and the maximum value of SRC predefined is met. When the handshake stops, if there is still only one CTS packet received, e.g., from node A, a normal point-to-point packet transmission between node S and A will be carried out, which shows the compatibility for previous transmission format of our AMPTR MAC protocol; if finally there is no CTS packet received, the transmitter S will lose the channel access for data transmission and a new random backoff timer with a doubled contention window size will be scheduled for retry with the increment of Long Retransmission Counter (LRC) by one.

In MPR case if the transmitter S does not receive the RTR packet from the receiver A, it could be regarded that node A has not successfully received the RTS packet from node S, therefore the RTS packet of node S should be retransmitted until node S receives the RTR packet, the handshake timer is out, or the retry limit is reached, i.e., the maximum value of SRC is met. If the RTR packet is still not received, data packet retransmission will be scheduled with a random backoff timer with a doubled contention window size and the LRC value will be increased by one. After the receiver A sends out the RTR control packet, if it does not receive the RRTR packet from any neighbor node U except the first transmitter S, a retransmission for the RRTR packet will be carried out until the receiver A receives a RRTR packet, the time deadline for sending the CTS packet back is met, or the retry limit is reached. If the RRTR packet is still not received, a CTS packet to the only transmitter S will be send out; otherwise, the CTS packet to both transmitter S and U will be sent out. If the first transmitter S receives a CTS packet, packets for the receiver A will be sent out with the antenna and packet information contained in the CTS packet; if node S does not receive any CTS packet, a random backoff timer with a doubled contention window size will be set for data retransmission and the LRC value will be increased by one. After sending out the RRTR packet, the second transmitter U will wait for the CTS packet. If node U does not receive any CTS packet, a random backoff timer with previous value before sending the RRTR packet will be set for channel access contention; otherwise, packets for the receiver A will be sent out with the antenna and packet information contained in the CTS packet.

Note that to make handshakes for MPT and MPR, four kinds of control packets are utilized: RTS, CTS, RTR and RRTR. Different with simple unicast control packet in previous work, to make handshakes efficiently both RTS and CTS packet should be compatible. The RTS packet in MPR and the CTS packet in MPT cases still aim to one receiver; however, the RTS packet in MPT and the CTS packet in MPR case should be sent out to multiple receivers, i.e., they should follow a multicast transmission format of single transmitter and multiple receivers. To incorporate the special requirement of transmission compatibility, the RTS and CTS packet should spare two address spaces for possible two receivers; when there is only one receiver (the RTS packet in MPR case and the CTS packet in MPT case), the second address space will be assigned a predefined number which is known as invalid address to all the nodes in network. One of the most important functions of control packets is to exchange each other's information among transmitters and receivers to get agreement on transmissions. Especially, for multiple concurrent data transmissions, there are extra information needed, e.g., for virtual MIMO-based concurrent transmissions, antenna and pilot information is needed. Therefore, in RTS packet a transmitter needs to include its antenna information for each of its receivers (one or two receivers). Based on those information, receivers can decide antenna weights for data transfer which should also be included in CTS packets. In addition, since we plan to use frame aggregation scheme to increase network throughput, transmitters need to indicate in RTS packets for each receiver how many packets will be aggregated and sent; accordingly receivers have to include the permitted packet number in CTS packets. The maximum number of packets which could be aggregated for one time data transmission could be predefined. In summary, the control packet formats of RTS and CTS for supporting multiple concurrent transmissions are respectively shown in Fig. 4 and Fig. 5.

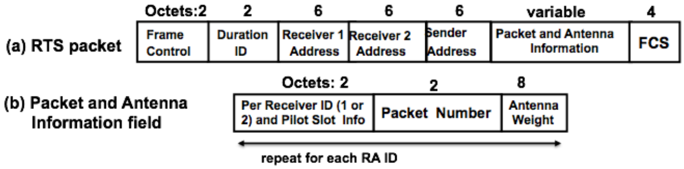


Fig. 4. Format of RTS packet

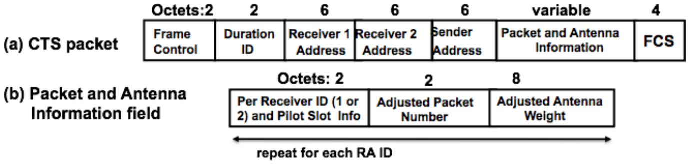


Fig. 5. Format of CTS packet

Note that in MPR case, the RTR packet is broadcasted to all its neighbors by the receiver to announce that more transmissions to it are needed. Therefore, the number of more permitted transmissions should be included in the packet, and in our example with the maximum number of concurrent transmissions $M = 2$, one more transmission to the receiver should be triggered, i.e., the number included in the RTR packet should be one. The node which wins the right for the one more transmission sends a unicast RRTR packet to the receiver to declare that it is ready for transmission, and similarly its antenna and packet information should also be included. In summary, the packet formats of RTR and RRTR are respectively shown in Fig. 6 and Fig. 7.

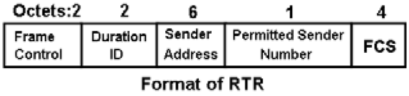


Fig. 6. Format of RTR packet

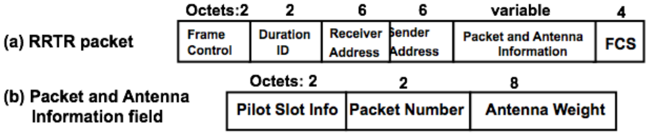


Fig. 7. Format of RRTR packet

2.4 Data Transfer

As we have presented in subsection 2.3, every node has two separate queues for storing packets aiming to two different neighbor nodes. Since every node in the queue has the same next hop destination, it provides the possibility to send them all together to increase network throughput. Therefore, we include in the RTS or RRTR packets the number of packets transmitters want to transfer, which is determined by the number of packets in the queue and the maximum number of packets permitted in one transmission (predefined in the network settings). The receivers will return back the number of permitted packets in transmissions based on the situations of transmitters and receivers, and the transmission conditions, such as the number of concurrent transmissions, the number of packets to be transmitted in each concurrent transmission, channel conditions, and so on.

After handshake is done, all or part of the packets in those queues could be sent out together. To do that, in our AMPTR protocol multiple MAC layer packets are designed to be concatenated into a large MAC layer packet, i.e., frame aggregation. For each MAC layer packet, it contains a MAC header and MAC data unit coming from network layer. When it enters into PHY layer for packet transmission, the PHY header and the frame check sequence (FCS) are then appended. In a large aggregated MAC packet, the multiple MAC packets which come from network layer and are attached with their own MAC headers are combined together to form the MAC data unit; each of the multiple MAC packets is assigned a fixed size space to be easily accessed by receivers for later packet retrieve. This aggregated packet format is shown in Fig. 8, and the case that there is only one data packet for transmission could be regarded as a special case with the number of aggregated packet as one.

When receivers successfully receive the aggregated data packet, with the fixed space size for each aggregated packet they could extract each MAC packet out. By checking parameters of those extracted packets (such as traffic ID, packet ID,

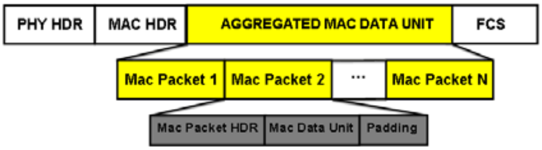


Fig. 8. Format of aggregated MAC packet

packet sequence number and so on), we could record which packets have been successfully received.

2.5 Block Acknowledgement

As shown in Fig. 1, after those aggregated packets are transmitted to receivers, transmitters expect to receive positive acknowledgement (ACK) packets from receivers. To provide efficient acknowledgements for multiple aggregated packets, we choose to add some bytes (namely, Block ACK Bitmap) in the ACK packet and use each bit of it to map the reception status of corresponding aggregated packet, namely aggregated block acknowledgement, i.e., BACK.

Our AMPTR scheme aims to support acknowledgements for all kinds of transmissions dynamically (MPT, MPR and normal point-to-point transmission). In MPT transmission process, as shown in Fig. 2, the transmitter S is expected to receive two BACKs from both node A and B. Since after data transmission transmitters and receivers know each other well to have concurrent transmissions, those two BACKs from node A and B could be concurrently transmitted to node S and then could be successfully received. In MPR transmission process, as shown in Fig. 3, both transmitters S and U are waiting for acknowledgements from node A. That is, after a time interval of Short Inter Frame Space (SIFS), node A will send one BACK aiming to both S and U. For this case, we propose that node A can include all the acknowledgements to two transmitters in one packet, and receiver ID could be included to differentiate them. Compared to the aggregated data packet the BACK packet is still pretty small even for two aggregated data packets.

To satisfy our special and dynamic acknowledgement requirement, the BACK packet should include two address spaces for possible two receivers. Especially, the transmitter of the BACK packet needs to indicate the reception status of any aggregated and transmitted packet coming from certain traffic flow of certain receiver, i.e., traffic ID (TID), packet ID, receiver ID and packet sequence number should be included in the BACK packet. To achieve above functions, the format of BACK packet is designed as Fig. 9, where BA represents a block acknowledgement. When the BACK packet only has one receiver, the left address space can then be assigned a predefined number known as invalid address.

If a BACK packet is not received within an interval of two round trip packet transmission, the sender assumes that the data packet transmission has experienced a transmission failure, and then contends channel access for retransmission

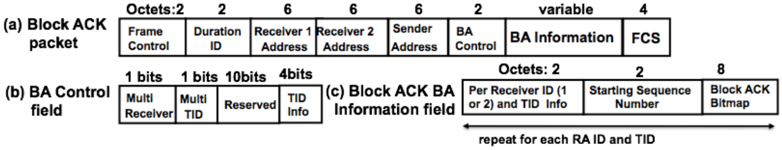


Fig. 9. Format of the Block ACK frame

with a random backoff timer with a doubled contention window size with the increment of Long Retransmission Counter (LRC) by one. If the sender can successfully receive a BACK packet, and extract for itself which aggregated packets are correctly transmitted. Those correctly received data packets will be removed from the sending queue and check transmission queue for next round of channel access contention.

3 Performance Evaluation

We evaluate the performance of AMPTR with and without frame aggregation scheme against that of IEEE 802.11 DCF through discrete-event simulations conducted in Qualnet v4.0 [11]. For our simulations, we used two network scenarios, namely a connected mesh and multi-hop topologies. In connected mesh scenarios, nodes are divided into clusters within which every node is within transmission range of one another, and traffics are generated within each cluster, i.e., destinations can be reached within one hop. In multi-hop scenarios, mobile nodes are randomly deployed and a packet may travel several hops before it reaches the destination.

As performance metrics, we used aggregate throughput for the whole network, average packet delivery ratio and average packet delivery delay, and MPTR/MPT ratio. In the connected mesh (single hop) scenario, packet delivery is carried out in one hop and packet delivery delay is calculated as the difference between the time a packet arrives at the queue and the time a packet gets transmitted successfully. In multi-hop scenarios, the evaluation metric of aggregate throughput is defined as aggregated traffic “end-to-end” throughput, that is, in the whole network how many bits per second are successfully sent out to reach destinations which may be multiple hops away from sources; similarly, end-to-end multi-hop packet delivery ratio and multi-hop end-to-end packet delivery delay are used for evaluation. In AMPTR protocol, when traffic flows are not high enough to allow MPT or MPR handshake to success, the conventional point-to-point transmission will be carried out. The MPTR ratio is the percentage of MPT and MPR transmissions occupied over all the data transmissions through the whole simulation. The MPT ratio plus MPR ratio equals to the MPTR ratio.

3.1 Simulation Setup

In our simulations, 36 nodes are deployed over a square area with $1000 \times 1000\, m^2$, the data rate for channel is 54 $Mbps$ and the transmission range is set to 250 m.

Fix-size (1024 bytes) data packets generated from CBR sources are continuously sent out. We use omni-directional antennas, and TwoRayGround propagation model. Each data point in graphs is an average of 10 different simulation runs. Each run is conducted with a random seed with a time duration of 10 minutes.

In connected mesh (single hop) scenarios, nodes are static, and placed in scheduled positions to form clusters. 36, 72, 108, 144 or 180 concurrent flows are simulated, one node can simultaneously generate several traffic flows to multiple destinations and multiple sources can simultaneously generate traffic flows for one same destination. Those kinds of traffic conditions enhance the probability of MPT or MPR processes.

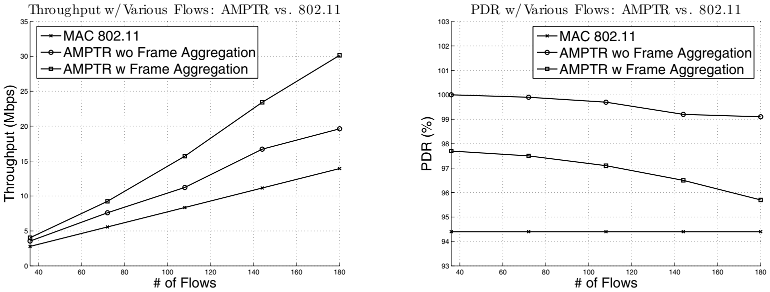
In multi-hop network scenarios, nodes are randomly placed and move according to the random way-point model with speeds varying between 0 and 5 m/s and with no pause. Multiple concurrent flows are also simulated, while sources and destinations are randomly selected such that a node may be the source for multiple destinations and a node may be a destination for multiple sources.

For both single and multiple hop scenarios, approximately, half of the nodes are sources and half are destinations, and all the traffic flows start at the same time and use the same traffic rate for one network configuration. The traffic rate varies from low to high, that is, traffic flows with 2, 5, 10, 20, 25, 33 and 50 packets per second (the inverse of traffic data rate) are individually simulated.

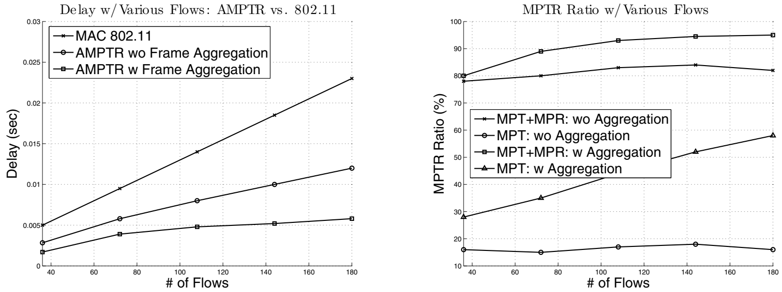
3.2 Simulation Results

We consider the advantages of our adaptive proposal for MPT and MPR in improving network capacity and reducing waiting and service time for packet transmissions, in terms of network throughput and packet delivery delay respectively, under various number of traffic flows, or various traffic data rate. We also observe the packet delivery ratio (PDR) and MPTR/MPT ratio, which reflect behaviors of MAC protocols. To be comprehensive, we make simulations in both single hop and multi-hop scenarios.

Fig. 10 (a) presents a comparison of simulation results on aggregate network throughput for single hop scenarios under various number of traffic flows. AMPTR with frame aggregation performs significantly better than IEEE 802.11 DCF and yields almost double times the throughput of 802.11 DCF, and even AMPTR without frame aggregation can still achieve around 30 percent higher throughput than that of 802.11 DCF. The performance gain of our AMPTR over 802.11 DCF is mainly due to two aspects: the ability of nodes in AMPTR transmitting or receiving multiple data streams concurrently, and the ability of nodes aggregating multiple packets for the same destination and sending them altogether in one time. Since in our simulation each node could concurrently transmit or receive two different data streams, the network throughput of AMPTR without frame aggregation is expected two times that of 802.11 DCF. The reason why the practical results are not as good as what we have expected is as follows. First, the handshake phase of AMPTR takes place among multiple nodes (may be more than two) and it takes more time and control messages to complete than that of 802.11 DCF. Secondly, the handshake for MPT or MPR may not



(a) Throughput (b) PDR



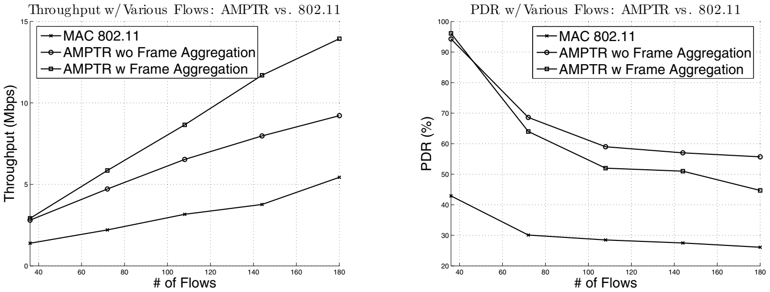
(c) Delay (d) MPTR/MPT Ratio

Fig. 10. Protocol Performance vs Traffic Flow Number in Single Hop Scenarios

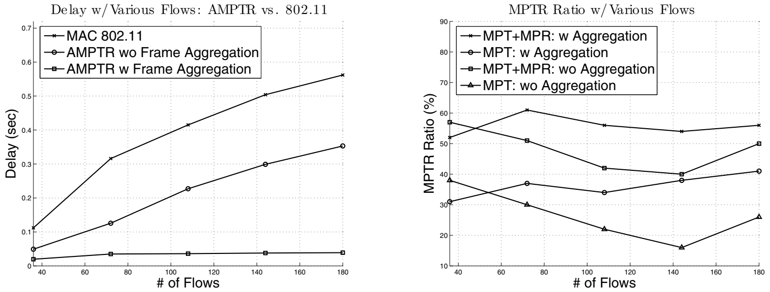
success, and at that time normal point-to-point will take place, which has spent longer time to achieve transmission agreement than 802.11 DCF does. Thirdly, the number of packets for the same destination varies and depends on traffic conditions, and sometimes frame aggregation may not be carried out when there is only one packet ready for transmission at that time.

From Fig. 11 (a) we observe that in multi-hop scenarios AMPTR performs slightly worse than it in single hop scenarios. That is because in multi-hop scenarios the aggregate throughput is for traffics which may need multiple hops to reach destinations. Also because the traffic packet delivery should overcome more challenges to reach destinations multiple hops away, the PDR in multi-hop scenarios (Fig. 11 (b)) is much less than that in single hop scenarios (Fig. 10 (b)). The MPTR/MPT ratios for both scenarios are presented in Fig. 10 (d) and Fig. 11 (d), respectively. We observe that the MPTR ratio in multi-hop scenarios is smaller than that in single hop scenarios, and accordingly its throughput is also lower than that in single hop scenarios.

Similarly for packet delivery delay, since each node could concurrently transmit two different data streams, and especially multiple packets could be aggregated and sent out in one time, the packet waiting for service time could be greatly reduced which in turn causes the great reduction in packet delivery delay. Our expectation has been validated in Figs. 10 (c) and 11 (c) for both single and multi-hop scenarios. In single hop scenarios, from low to high traffic flows



(a) Traffic Throughput (b) End-to-End PDR

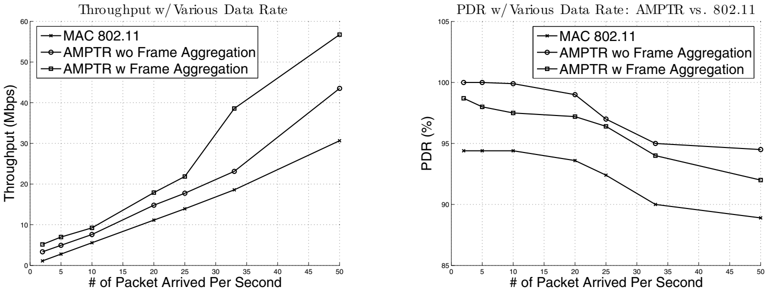


(c) End-to-End Delay (d) MPTR/MPT Ratio

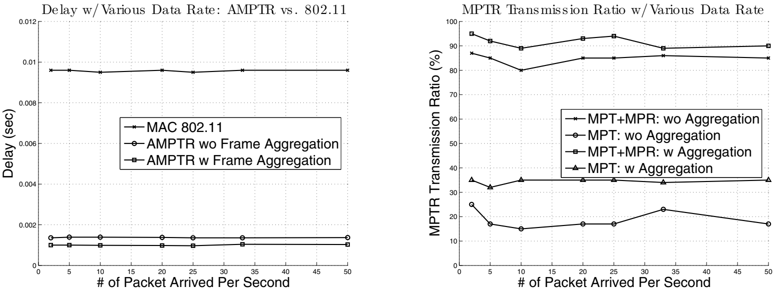
Fig. 11. Protocol Performance vs Traffic Flow Number in Multi-hop Scenarios

the delay of AMPTR with frame aggregation can be only around 1/3 to 1/4 of that of 802.11 DCF; even the performance of AMPTR without fame aggregation is worse than that of AMPTR with frame aggregation, it is only 60 percent of the delay of 802.11 DCF. In multi-hop scenarios, the delay is not for one single hop, but for multiple hops from a source to a destination, i.e., end-to-end delay; the delays for both 802.11 and AMPTR without aggregation increase by around 20 times, while that with frame aggregation only increases by around 6 times. The advantage of frame aggregation is shown as it can reduce many times handshakes into one; considering the possible retransmission for each hop, the number of handshakes reduced will be affected by the delivery hop number, the number of aggregated frames plus retransmission times. The more handshakes it can reduce, the smaller the packet delivery delay will be. However, from Figs. 10 (b) and 11 (b) we notice that the frame aggregation also has disadvantages. Because the packet size is increased, the transmission failure probability is also increased which in turn caused the lower PDR than that of AMPTR without frame aggregation, even though AMPTR with frame aggregation still has higher PDR than 802.11 DCF.

We also take a look at the protocol performance under various traffic data rate, i.e., varying the number of arrival packets per second, in Figs. 12 and 13 with the number of traffic flows of 36*2. In single hop scenarios, it is obvious and interesting that as the number of arrival packets increases, the MPTR/MPT



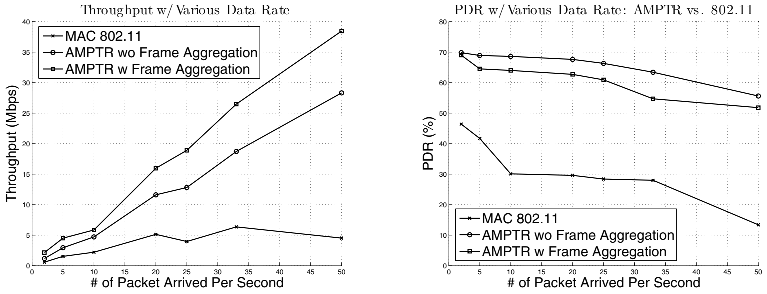
(a) Throughput (b) PDR



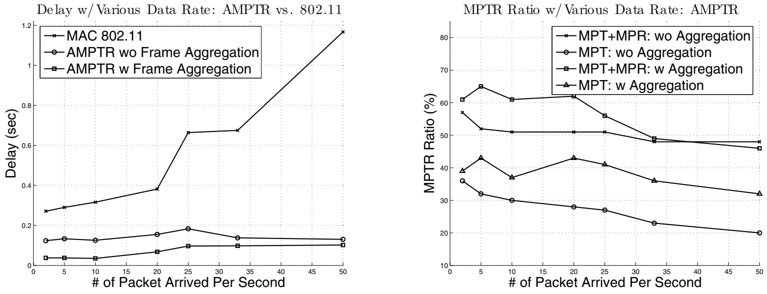
(c) Delay (d) MPTR/MPT Ratio

Fig. 12. Protocol Performance vs Arrival Packet Number Per Second in Single Hop Scenarios

ratio keeps the same, as shown in Fig. 12 (d). It is heuristic that data rate does not change the traffic pattern. However, since there are more packets sent out per second and for AMPTR with frame aggregation more packets for possible aggregation, the throughput for all the protocols increases and especially AMPTR with frame aggregation achieve larger increment ratio than other protocols (shown in Fig. 12 (a)). Note that in Fig. 12 (c) as data rate increases the packet delivery delay basically keeps the same, which may occur when the data rate is much smaller than the packet delivery delay. For example, the maximum data rate in our simulations is $1/50 = 0.02$, while all the packet delivery delay is less than 0.01. That is, nodes could send packets out before new packets come. While traffic data rate increases, the PDR decreases slightly (Fig. 12 (b)), which may be caused by the channel access schedule effected by increasing data rate. However, multi-hop scenarios have more complicated situations (Fig. 13). From Figs. 13 (b), 13 (a) and 13 (c), we observe that as the number of arrival packets increases, the PDR of AMPTR decreases lighter than that of 802.11 DCF, and that the throughput of AMPTR increases while that of 802.11 DCF has no obvious increment, and that the delay of 802.11 DCF increases greatly while that of AMPTR increases very slightly. All those observations demonstrate the stability and advantage of our proposal in multi-hop scenarios as traffic packet rate increases.



(a) Traffic Throughput (b) End-to-End PDR



(c) End-to-End Delay (d) MPTR/MPT Ratio

Fig. 13. Protocol Performance vs Arrival Packet Number Per Second in Multi-hop Scenarios

In summary, our AMPTR proposal can greatly improve network capacity and reduce packet transmission waiting and service time for ont only single hop networks but also random deployed mobile multi-hop topologies.

4 Conclusions

We presented an adaptive MAC protocol which exploits the advanced physical layer MPT and MPR ability, by supporting nodes which access channel to transmit multiple packets concurrently or receive multiple packets concurrently. We implemented our proposal in Qualnet and demonstrated its advantages in improving network throughput and reducing packet delivery delay by comparing its performances with those of 802.11 DCF.

References

1. Wang, Z., Ji, M., Sadjadpour, H., Garcia-Luna-Aceves, J.J.: Cooperation-Multiuser Diversity Tradeoff in Wireless Cellular Networks. In: IEEE Globecom 2009 Wireless Networking Symposium, USA, pp. 1–6. IEEE Press, Honolulu (2009)
2. Paulraj, A., Nabar, R., Gore, D.: Introduction to Space-Time Wireless Communications. Cambridge University Press, Cambridge (2003)

3. Mirkovic, J., Zhao, J., Deneneer, D.: A MAC Protocol with Multi-User MIMO Support for Ad-Hoc WLANs. In: IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5. IEEE Press, Athens (2007)
4. Sadjadpour, H., Wang, Z., Garcia-Luna-Aceves, J.J.: The Capacity of Wireless Ad Hoc Networks with Multi-Packet Reception. IEEE Transactions on Communications 58(2), 600–610 (2010)
5. Jin, H., Jung, B.C., Hwang, H.Y., Sung, D.K.: A MIMO-Based Collision Mitigation Scheme in Uplink WLANs. IEEE Communications Letters 12(6), 417–419 (2008)
6. Ke, B.W., Zhang, Y.J., Liew, S.C.: Media Access Control with Spatial Correlation for MIMO Ad Hoc Networks. In: IEEE International Conference on Communications, pp. 3660–3665. IEEE Press, Glasgow (2007)
7. Park, M., Choi, S.-H., Nettles, S.M.: Cross-layer MAC Design for Wireless Networks Using MIMO. In: IEEE Global Telecommunications Conference, pp. 2870–2874. IEEE Press, St. Louis (2005)
8. Mirkovic, J., Orfanos, G., Reumerman, H.-J., Deneneer, D.: A MAC Protocol for MIMO Based IEEE 802.11 Wireless Local Area Networks. In: IEEE Wireless Communications and Networking Conference, pp. 2131–2136. IEEE Press, Hong Kong (2007)
9. Mirkovic, J., Zhao, J., Deneneer, D.: A MAC Protocol with Multi-User MIMO Support for Ad-Hoc WLANs. In: IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5. IEEE Press, Athens (2007)
10. IEEE 802.11 Working Group: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 4: Enhancements for Higher Throughput. IEEE-SA Standards Board (2008)
11. Scalable Network Technologies: Qualnet Simulator v. 4.0