An Improved Data Flushing MAC Protocol for IEEE 802.11 Wireless Ad Hoc Network

Shiann-Tsong Sheu†, Tobias Chen †, Jenhui Chen‡, and Fun Ye†
†Department of Electrical Engineering,
‡Department of Computer Science and Information Engineering,
Tamkang University, Tamsui, Taipei, Taiwan, R.O.C.
E-mail: stsheu@ee.tku.edu.tw; tchen@issey.net; jenhui@cs.tku.edu.tw

***Abstract*—In this paper, we propose a data flushing data transfer (DFDT) protocol. The distributed coordinate function (DCF) of IEEE 802.11 supports data transmissions using the data-ACK method and the request-to-send/clear-to-send (RTS/CTS) method. The data-ACK method has low protocol overhead, however, the transmissions are prone to collision. Although the RTS/CTS mechanism reduces the probability of collisions of data packets, the handshaking generates extensive overhead. Another issue with the IEEE 802.11 DCF is the contention for channel access; much bandwidth is wasted with the contention, especially when the mean data length is short. DFDT is capable of sending out multiple data packets from the upper layer, after acquiring channel access by a successful contention, within one frame which we call compiled MPDU (cMPDU). Right after the transmission of the data frame, the destination nodes will reply an positive/nagative acknowledgement in a consecutive manner. By using this method, the protocol overhead is relatively lowered while retaining service quality and the waste of bandwidth for contention is also reduced. Simulation results show that DFDT can handle higher traffic load and has better throughput then the IEEE 802.11 MAC protocol.**

***Index Terms*— ad hoc, IEEE 802.11, LAN, MAC, RTS/CTS, Wireless**

I. Introduction

In recent years, mobile computing devices such as PDAs (personal digital assistants) and portable computers have become essential in our daily activities. Now that mobile computing is very common, the demand for mobile data exchange is also swiftly growing. As a result, wireless communication is rapidly emerging, providing users with network connectivity without being restricted by a wired network. Data exchange is often desired by a group of people within a limited area, such as information exchange between members in a working group, emergency rescue team, battle field, etc. Such a group of people form an ad hoc wireless local area network (WLAN), which is a collection of mobile hosts that can rapidly build up communication networks without the aid of any pre-established infrastructure or centralized administration. Ad hoc WLANs provide a convenient solution for data exchange in small areas where wiring for conventional networks is difficult or not economic. Since any transmission in a WLAN relies on a common and open radio medium, the medium access control (MAC) protocol in WLAN would be more important than in conventional wired networks. Many researches and proposals concerning the MAC protocol for WLAN have been made, some of them are [1], [7], [8], [10], [11], [13], [14], [15], [16], [18].

In [9] the IEEE 802.11 working group presents a very well known and widely used WLAN protocol which comprises specifications of a physical (PHY) layer and a MAC layer. The PHY layer specification includes three independent systems: the frequency-hopping spread spectrum (FHSS) system, the direct sequence spread spectrum (DSSS) system, and the infrared (IR) system, in which the DSSS system is the most popular. The MAC layer includes a basic distributed coordination function (DCF) and an optional point coordination function (PCF). It is also possible to have a mixed configuration with both DCF and PCF operating simultaneously within the same basic service set (BSS). The DCF, used as the basic channel access protocol to transmit asynchronous data in the contention period, is known as carrier sense multiple access with collision avoidance (CSMA/CA [6]) including an optional transmission mode based on RTS/CTS handshaking (DFWMAC [4]). The PCF is a centralized MAC protocol only used with infrastructure network configuration that supports collision free and time bounded services.

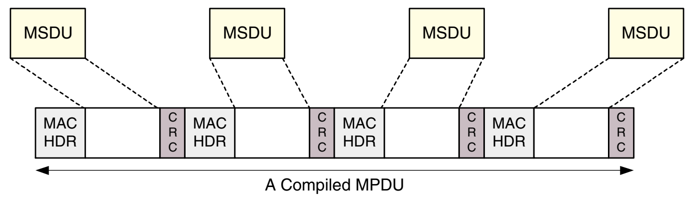

Fig. 1. The compilation process.

Previous researches on packet-radio networks enhanced the performance by increasing protocol overhead which was intended to eliminate or lower the probability of collisions (two or more stations transmitting at the same time jamming each others signals). In a packet-radio network the protocol overhead is essential for providing a stable link between different wireless stations, but the overhead also decreases the theoretical throughput of the network. The data flushing data transfer (DFDT) protocol not only concentrates on the stability of the network by using extra overhead but also considers other factors such as: contention, packet length, and packet arrival rate. These factors also affect the stability and throughput of the network.

This paper is organized as follows. Section II introduces the DFDT protocol. A series of illustration of the operation of DFDT is given in III. The simulation environment and the simulation results are given in section IV. Finally, some conclusions are made in Section V.

0-7803-7467-3/02/$17.00 ©2002 IEEE.

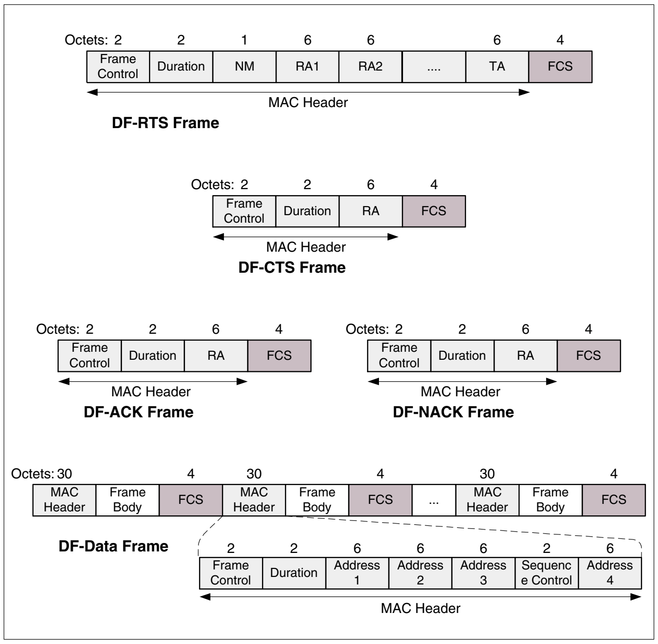

Fig. 2. The frame format of DFDT.

II. The Data Flushing Data Transfer Protocol

The data flushing data transfer (DFDT) protocol is an enhancement for the DCF of IEEE 802.11 or other similar wireless ad hoc networks. The Load of the network is determined by $L = N \times \lambda \times l$, where L is the load of the network, λ is the packet arrival rate of a single node, and l is the mean length of the packet. We are not able to change the number of stations and the total amount of data each node transfers, but as L retains the original value we can balance λ and l. We know that for a fixed L and N the throughput is higher with a low λ and a high l than a high λ and a low l. The main concept of DFDT is to increase l and decrease λ while retaining the product of them.

In Fig. 1, we see a diagram of the compilation process (CP) which is the most essential part of DFDT. First, the CP takes as many MAC service date units (MSDUs) from the Tx queue of the upper layer limited by the compilation threshold (CT). Then the CP will add a MAC header and a CRC to each MSDU generating standard MPDU packets. At last, the CP will combine all the MPDUs to form a compiled MPDU (cMPDU). CT indicates the maximum length of a cMPDU. We know that larger packets have grater risk of being corrupted, therefore the IEEE 802.11 limits the body of a MPDU to be less than the fragmentation threshold (FT) or it will be cut into multiple parts where each part will be shorter than the FT. For the same reason as implementing FT, the CT prevents CP of producing cMPDUs that are of excessive length. The main difference between a MPDU and a cMPDU is that a MPDU carries only one or one part of a MSDU whereas a cMPDU could carry multiple MSDUs with different destinations in one pack.

Listed below are the control and date frame names of DFDT, and the frame formats are illustrated in Fig. 2.

- data-flushing-request-to-send (DF-RTS)
- data-flushing-clear-to-send (DF-CTS)
- data-flushing-acknowledgment (DF-ACK)
- data-flushing-negative-acknowledgment (DF-NACK)
- data-flushing-data (DF-Data)

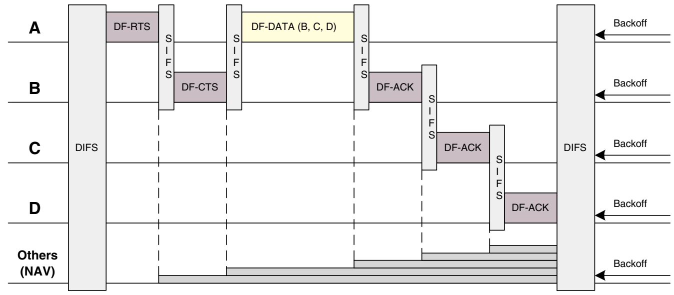

Fig. 3. The basic mechanism of DFDT (In this diagram the data frame is compiled from three data packets).

It is easy to notice that the basic frame structure is taken from the IEEE 802.11. The DF-CTS and DF-ACK are exactly the same as the clear-to-send (CTS) and acknowledgment (ACK) of the IEEE 802.11. DF-NACK indicates an unsuccessful transmission which is the opposite of DF-ACK. As IEEE 802.11 does not give acknowledgements for unsuccessful transmissions, DF-NACK is the only frame that does not have a corresponding control frame in the IEEE 802.11 . The DF-RTS is not completely the same but still similar to the request-to-send (RTS) of the IEEE 802.11. The major difference between DF-RTS and RTS of IEEE 802.11 is that DF-RTS has a NM field which indicates the number of receiver addresses (RAs) fields of the DF-RTS frame. Multiple RA fields are needed to notify multiple destinations of the incidence of an upcoming DF-Data frame. This means that DF-RTS is of dynamic length and depends on the number of MPDUs compiled into the cMPDU the DF-RTS is representing. Note that the length of a DF-RTS frame influences the probability of success of the DF-RTS/DF-CTS dialogue, and it might be desired to have a threshold value for the maximum length of a DF-RTS. The DF-Data frame is a cluster of one or more sub-frames where each sub-frame is identical to a IEEE 802.11 MPDU. Having DF-Data sub-frames identical to IEEE 802.11 MPDUs makes it easier to integrate DFDT with IEEE 802.11 architecture and more robust than having a single MAC header carrying all information in the front.

The mechanism of DFDT is much like the mechanism of IEEE 802.11 including two methods for transmission: a) direct DF-Data/DF-ACK and b) DF-RTS/DF-CTS/DF-Data/DF-ACK. The RTS threshold (RT) is the switch to decide which method to use; method (a) is used if the length of the DF-Data frame is shorter than RT, and if vice versa then method (b) is used. For simplicity reasons, in the rest of this article the RT is set to zero which means that only method (b) is used. An example of the mechanism of DFDT where the DF-Data frame is compiled of three MSDUs is shown in Fig. 3. The source sends out DF-RTS when the DF-Data frame is ready for transmission. Then the first destination node (RA1) will reply DF-CTS to the source if the DF-RTS was correctly received. DFDT requires only RA1 to reply DF-CTS, because one DF-CTS is enough to reserve the radio channel in an ad hoc environment where all nodes can hear each other. After the DF-RTS/DF-CTS hand-

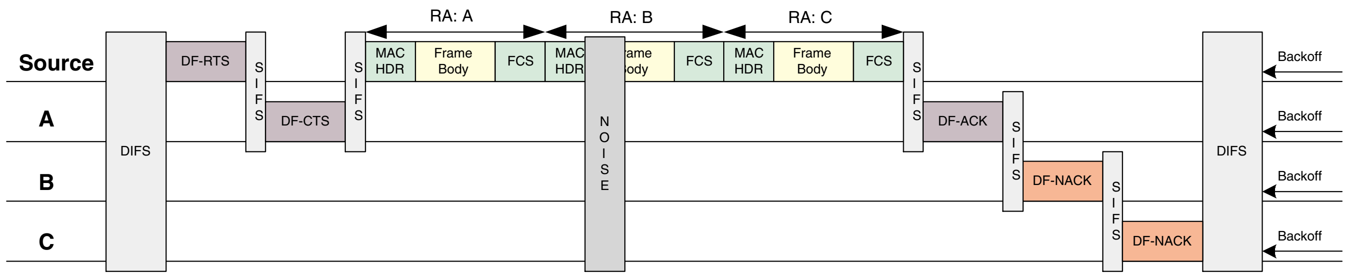

Fig. 4. Example of noise interfering all receivers during the transmission of the DF-Data frame.

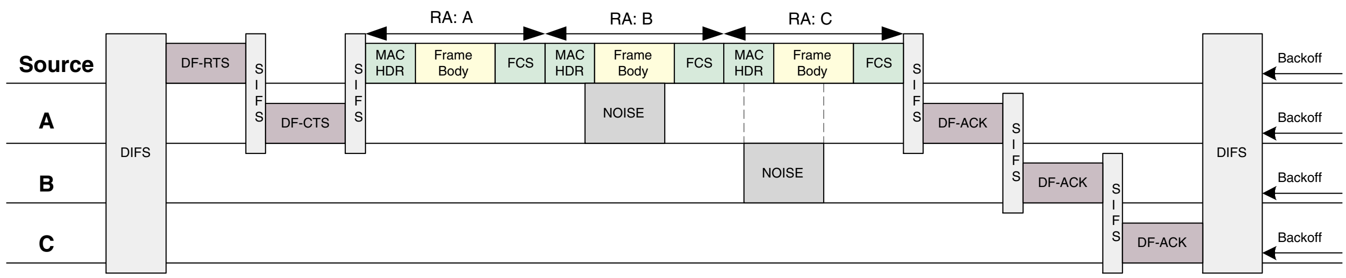

Fig. 5. Example of noises interfering individual receivers during the transmission of the DF-Data frame.

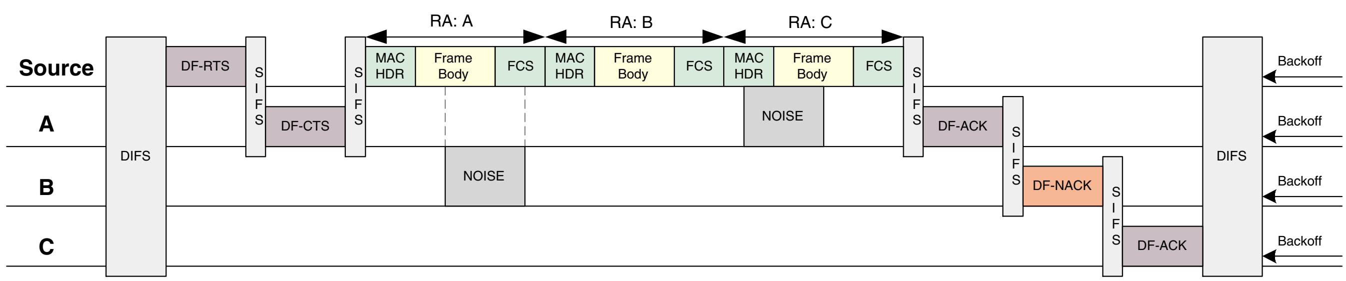

Fig. 6. Another example of noises interfering individual receivers during the transmission of the DF-Data frame.

shake, the transmission of the DF-Data begins. If the source did not hear a DF-CTS, it will retry transmitting DF-RTS. Following the DF-Data transmission, the destination nodes will confirm the status of their reception by replying: DF-ACK indicating a successful transmission, or DF-NACK suggesting that the transmission failed. The DF-ACK/DF-NACK replies are separated by a SIFS and ordered according to the data sequence in the DF-Data frame.

III. Illustrations of the Mechanism of DFDT

The basic mechanism of DFDT working in a error free environment was introduced in the preceding section, and here we will present some examples of DFDT working in a faulty environment. Errors occurring during the DF-RTS/DF-CTS handshake will not be discussed since the behavior of DF-RTS/DF-CTS handshake is completely the same as the RTS/CTS handshake of IEEE 802.11.

In Fig 4 some noise jammed all receivers for some time while the DF-Data frame was on air. Because the jam occurred in the second sub-frame, the first sub-frame could therefor still be successfully received by node A. It is clear that the second sub-frame destined for node B arrived broken. The third sub-frame could not be distinguished by node C since the information on where the third sub-frame started was destroyed as the noise jammed the second sub-frame (The duration field in the MAC header of a sub-frame, except for the last sub-frame, is used to determine the length of the particular sub-frame). After the DF-Data frame, node A replied a DF-ACK indicating a successful transmission while B and C replied DF-NACK indicating a failure. The DF-NACK is needed to keep the channel busy; preventing other stations from accessing the channel in the duration where the receiver nodes are scheduled to reply DF-ACK for a successful transmission.

If noises occur in small areas, not covering all receivers at once, transmissions might be completely unaffected, Fig. 5 shows such an example. node A certainly did not have a problem receiving the first sub-frame while node B was also able to determine the location of the second sub-frame because the first sub-frame was correctly received by it, and C was not affected by any noise so the transmission had to be successful. All three nodes replied DF-ACK consecutively following the DF-Data frame to report the success of reception to the transmitter. In Fig. 6 we see another example, this time with noise at an unhappy spot. The noise at node A has no influence on the reception of the first sub-frame which is the only concern of node A, but the noise at node B damages the first sub-frame leaving

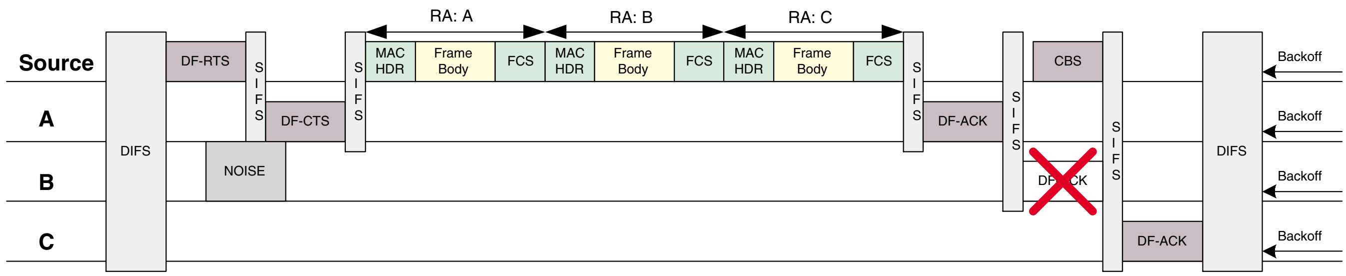

Fig. 7. Example of a noise resulting in a missing DF-ACK/DF-NACK.

the rest of the cMPDU unreadable. Although the section of the second sub-frame was correctly received, node B was unable to extract the desired data out of the DF-Data frame due to the fact that the first sub-frame was broken. A reception is successful if the cMPDU is readable from the beginning of the cMPDU to the end of the sub-frame destined to that receiver, the rest could be totally damaged.

Because the DF-Data frame is mostly followed by multiple replies from different destination nodes, there could always be such case that a certain destination node fails to send a reply. In Fig. 7, the DF-RTS/DF-CTS dialogue was not sensed by node B because of noise interference, therefore node B did not listen to the DF-Data frame and also did not give any reply after the DF-Data frame. The gap of a missing DF-ACK/DF-NACK could be interpreted as a free channel by another transmitter, and all communications would be jammed if any transmission started there. To prevent this from happening, the transmitter has to keep the channel busy if any receiver fails to reply a DF-ACK or DF-NACK. In Fig 7 the transmitter kept the channel busy by transmitting a channel busy signal (CBS).

IV. Simulation

The simulation was made by a custom program that implemented the IEEE Standard 802.11b-1999 [9] and DFDT in a synthetic environment. Direct sequence spread spectrum (DSSS) system was used for the PHY layer and at the MAC layer we used DCF as the medium access protocol with long PLCP PPDU format. The parameters used in the simulations throughout this article are listed in Table I where most of them were taken from the standard. The number of MSDU arrivals were decided by poisson distribution and the length of the MSDU was given by the exponential distribution function. Some assumptions were made to reduce the complexity of the simulation model:

- The data rate of all communications was fixed at 2Mb/s, leaving out other data rates such as 11Mb/s.
- The propagation delay was neglected.
- The channel was error-free; No noise from other devices or interference from nearby BSS.
- All stations were active; non in power-saving mode.
- The ad hoc network was perfectly fully connected, which means that there were no hidden terminals and each node had direct radio contact with other nodes.

In Fig. 8, the throughput of IEEE 802.11 and DFDT are shown for different traffic loads generated by 25 nodes with the RTS/CTS handshaking mechanism always on and the CT and FT set to the maximum value (2312 Octets). The simulation was done with five different MDL values which generated five pairs of curves. In each pair the solid curve indicates the result of the IEEE 802.11 while the result of the DFDT is indicated by the dashed curve. Inspecting the curve of IEEE 802.11 MAC with MDL = 128 Octets, as the load reaches 20 packets/sec the network saturates with the throughput only at 0.24. If the DFDT was used saturation would occur only when the load reaches 40 packets/sec with throughput up to 0.6. The results show a saturation throughput improvement of 150% and saturation load improvement of 100% for DFDT over IEEE 802.11 MAC. As for the other MDLs we can see a similar phenomenon, for MDL = 256, 512, 1k, 2k Octets the saturation throughput improvements are about 76%, 34%, 4%, 1% respectively, and saturation load improvements are about 66%, 20%, 1%, 0% respectively. From the results we can see enormous improvements archived by DFDT when the MDL is small. Although the improvements are not obvious when the MDL is close to the maximum value, DFDT will never have performance lower than the IEEE 802.11 MAC.

TABLE I

Parameters used throughout this section

Parameters	Values
aSlotTime	20 μs
aSIFSTime	10 μs
aDIFSTime	50 μs
aPreambleLength	144 μs
aPLCPHeaderLength	48 bits
CW_{min}	31 slots
CW_{max}	1023 slots
dot11MaxTransmitMSDULifetime	512 ms
dot11MaxReceiveLifetime	512 ms

The CT has a major influence on the saturation throughput of the DFDT protocol; using a larger CT will result in a higher saturation throughput. Note that this is only true for a error-free network (bit error rate (BER) = 0), because if we had a BER grater then zero the error penalty would be higher with a larger CT. In reality BER is always grater then zero and CT should be in inverse ratio with it. Fig. 9 shows the DFDT with different CT values and the IEEE 802.11 MAC with corresponding FT values, note that CT = FT. The different FT values actually do not effect the throughput of IEEE 802.11 much because the MDL in this simulation is quite small; we can see that the throughput curves with different FTs are all identical. For the

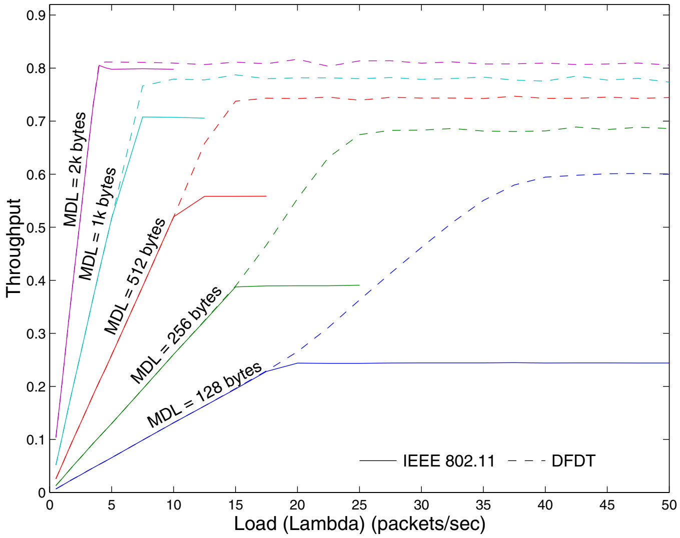

Fig. 8. Comparison of throughput of IEEE 802.11 and DFDT with different MDLs when N = 25 and CT = 2312 octets.

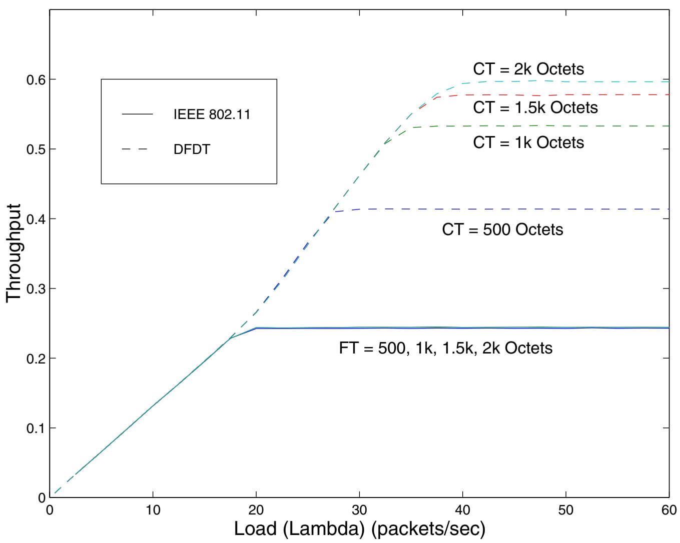

Fig. 9. Comparison of throughput of IEEE 802.11 and DFDT with different CT and FT values when N = 25 and MDL = 128 octets.

curves of DFDT with different FT values, we see a higher performance with each grater FT value. The saturation throughput improvement of DFDT over the IEEE 802.11 MAC for FT = 500, 1k, 1.5k, 2k Octets are about 85%, 118%, 136%, 144% respectively. As the CT decreases, the performance curve of DFDT and IEEE 802.11 will eventually overlap.

V. Conclusions

In this paper we have introduced and investigated DFDT which serves to be an enhancement for the IEEE 802.11 or other ad hoc WLAN MAC protocols. The major innovation of DFDT is the compilation process which allows a station to combine multiple MSDUs destined for different receivers into one single cMPDU. The cMPDU are then sent out in one physical data packet after a successful contention. The DFDT improves the IEEE 802.11 MAC by the following two concepts: lowering the contention of the network, and decreasing the percentage of overhead required for a transmission. The concepts are simply achieved by the compilation process. When we have high percentage of short packets, which is often the case for wireless networks, DFDT shows enormous performance improvement over the IEEE 802.11.

Results of this paper show that DFDT outperforms the IEEE 802.11 MAC and thus serves as an excellent enhancement for to the IEEE 802.11 or other WLAN MAC protocols. Although DFDT was presented as a one hop (ad hoc) packet-radio network protocol, it should also works well in an infrastructure environment. With some modifications DFDT could also work in a multihop wireless network environment.

References

[1] N. Abramson, "The ALOHA System–Another Alternative for Computer Communications," *Proc. Fall Joint Computer Conf.*, pp. 281–285, 1970.

[2] G. Bianchi "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[3] F. Calì, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Networking*, vol. 8, no. 6, pp. 785–799, Dec. 2000.

[4] K.-C. Chen, "Medium access control of wireless LANs for mobile computing," *IEEE Network*, vol. 8, no. 5, pp. 50–63, Sept.-Oct. 1994.

[5] H.S. Chhaya and S. Gupta, "Performance Modeling of Asynchronous Data Transfer Methods in the IEEE 802.11 MAC Protocol," *ACM/Baltzer Wireless Networks*, vol. 3, no. 3, pp. 217–234, Aug. 1997.

[6] A. Colvin, "CSMA with collision avoidance," *Computer Commun.*, vol. 6, no. 5, pp. 227–235, 1983.

[7] C.L. Fullmer, and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," *Proc. ACM SIGCOMM'95*, pp.262–273, Cambridge, MA, USA, Aug.-Sep. 1995.

[8] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard and B. Ramanurthi, "Packet Reservation Multiple Access for Local Wireless Communication," *IEEE Trans. Commun.*, vol. 37, pp. 885–890, Aug. 1989.

[9] IEEE 802.11 Working Group, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ANSI/IEEE Std. 802.11, Sept. 1999.

[10] P. Karn, "MACA-A new channel access method for packet radio," *Proc. 9th ARRL/CRRL Amateur Radio Computer Networking Conf.*, pp. 134–140, Sept. 1990.

[11] C.R. Lin and M. Gerla, "Asynchronous Multimedia Multihop Wireless Network," *Proc. IEEE INFOCOM'97*, vol. 1, pp. 118–125, Apr. 1997.

[12] B. O'Hara and A. Petrick, *The IEEE 802.11 Handbook: A Designer's Companion*, IEEE press, Dec. 1999.

[13] T. Ozugur, M. Naghshineh, P. Kermani, C.M. Olsen, B. Rezvani, and J.A. Copeland, "Balanced Media Access Methods for Wireless Networks," *ACM/IEEE MobiCom'98*, pp. 21–32, Oct. 1998.

[14] S.-T. Sheu and T.-F. Sheu, "A Bandwidth Allocation/Sharing/Extension Protocol for Multimedia Over IEEE 802.11 Ad Hoc Wireless LANs," *IEEE J. Select. Areas Commun.*, vol. 19, no.10, pp. 2065–2080, Oct. 2001.

[15] G.S. Sidhu, R.F. Andrews, and A.B. Oppenheimer, *Inside AppleTalk*, 2nd ed., Addison-Wesley, 1980.

[16] Z. Tang and J.J. Garcia-Luna-Aceves, "Hop-Reservation Multiple Access (HRMA) for Ad-Hoc Networks," *Proc. IEEE INFOCOM'99*, vol. 1, pp. 194–201, New York, Mar. 1999.

[17] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu,"The Broadcast Storm Problem in a Mobile Ad Hoc Network," *ACM/Kluwer Wireless Networks*, vol. 8, no. 2–3, pp. 153–167, Mar. 2002.

[18] F.A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II–The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Trans. Commun.*, vol. COM–23, no. 12, pp. 1417–1433, Dec. 1975.