Wireless Netw (2009) 15:805–820
DOI 10.1007/s11276-007-0076-z

SDMAC: Selectively Directional MAC protocol for wireless mobile ad hoc networks

Pan Li · Hongqiang Zhai · Yuguang Fang

Abstract Using directional antennas in wireless mobile ad hoc networks can greatly improve the transmission range as well as the spatial reuse. However, it will also cause some problems such as deafness problem and hidden terminal problem, which greatly impair the network performance. This paper first proposes a MAC protocol called Selectively Directional MAC (SDMAC) that can effectively address these problems and significantly improve the network throughput. Then two improvements on SDMAC are proposed. The first one is to improve the network throughput by scheduling the packets in the queue (a scheme called Q-SDMAC), thus the head-of-line (HOL) blocking problem can be addressed. The second one is to relax the assumption that each node knows the relative directions of its neighboring nodes and use caches to buffer those relative directions (a scheme named Q-SDMAC using cache). Extensive simulations show that: (1) SDMAC can achieve much better performance than the existing MAC protocols using directional antennas; (2) The network throughput can be significantly improved by scheduling the packets in the queue; (3) Using caches can still achieve high network throughput when nodes are moving; and (4) Network throughput decreases when directional antennas have side lobe gain.

Keywords Wireless mobile ad hoc networks · MAC protocol · Directional antennas

1 Introduction

A wireless mobile ad hoc network is a network where users can communicate with each other without the support of infrastructure. It can be set up easily and quickly with low cost. As a result, wireless mobile ad hoc networks have many applications for commercial and military purposes.

Since the wireless channel is shared by all the users in the network, a medium access control (MAC) protocol is needed to coordinate the transmissions. The IEEE 802.11 DCF (Distributed Coordination Function) is such a protocol, known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with an optional use of RTS/CTS [1]. This protocol has been widely used in wireless ad hoc networks and our study here is based on this protocol architecture.

IEEE 802.11 assumes omnidirectional antennas for the users in the network. During a transmission, all users in the neighborhood of a sender or a receiver are expected to keep silent to avoid collision or interference with the ongoing transmission, leading to low spatial reuse. However, when directional antennas are used, some transmissions can be carried out at the same time without interfering with each other. For example, in the scenario 1 shown in Fig. 1(a), by using directional antennas we can allow the transmission between A and B, and the transmission between C and D at the same time. Thus, the spatial reuse can be improved.

P. Li · Y. Fang (✉)
Department of Electrical and Computer Engineering,
University of Florida, Gainesville, FL 32611-6130, USA
e-mail: fang@ece.ufl.edu

P. Li
e-mail: lipanleo@ufl.edu

H. Zhai
Philips Research North America, 345 Scarborough Rd,
Briarcliff Manor, NY 10510, USA
e-mail: hong.zhai@philips.com

Y. Fang
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, ROC

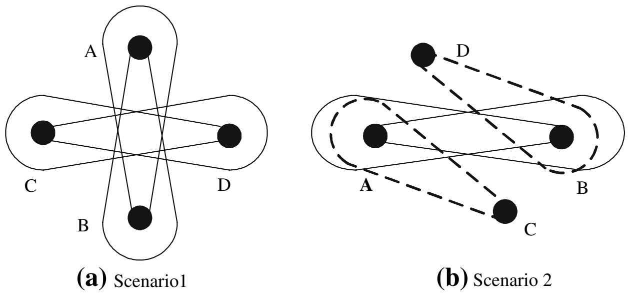


Fig. 1 Two scenarios where directional antennas are used

The transmission range can also be increased because of larger directional antenna gain and less interference, which may in turn reduce the number of end-to-end hops. Unfortunately, some problems also come along with the use of directional antennas, such as the deafness problem [2–4], the new hidden terminal problem [5], and the head-of-line (HOL) blocking problem [6]:

Deafness problem: This happens when a node sends out a RTS to the intended receiver but gets no response. Then the sender will double its contention window and then backoff. If the intended receiver is transmitting or receiving a large data packet, the sender will fail to get CTS for several times. So after the receiver finishes its transmission and becomes idle, the sender will have a large contention window and may probably have chosen a very long backoff period. Then the channel will be idle for a long time. What is worse, the receiver may want to initialize a new transmission with other nodes. It will choose a backoff interval according to a much smaller contention window than that of the sender. As a result, the receiver will likely be able to start another transmission before the sender sends out its RTS. Thus, the sender will keep deaf for a very long time. It may even drop the packet after it exceeds the maximum number of unsuccessful attempts. Figure 1(b) shows a scenario of the deafness problem. In this case, there is a transmission between node A and node B. During the transmission, A will not be able to receive the RTS from C because it is beamforming in a different direction. So C will not get any response from A. Similar to that, D will get no response from B if it sends a RTS to B. Thus, both C and D suffer from the deafness problem.

New hidden terminal problem: Hidden terminal problems are well studied for IEEE 802.11 MAC protocol when omnidirectional antennas are used. It can be alleviated by exchanging RTS/CTS prior to the data transmission. However, when we use directional antennas, the hidden terminal problem could become more serious.

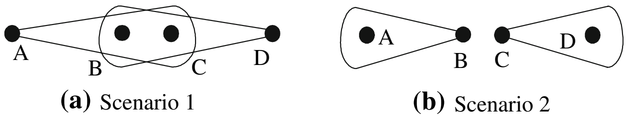


Fig. 2 New hidden terminal problems

1. *Hidden terminals caused by asymmetry in gain*: Consider the scenario shown in Fig. 2(a). There is a directional transmission from node A to node B. When node D is in omni-mode with antenna gain G^o, it is out of the transmission range of node A. But when node D is in directional mode with antenna gain G^d, it is within the transmission range of node A. So when D is performing carrier sensing omnidirectionally, it will not be able to detect the directional transmission between A and B. If D has a packet to A, or B, or C, then this transmission can be carried out, which will interfere with the reception of ACK at A.
2. *Hidden terminals caused by unheard RTS/CTS*: In the scenario shown in Fig. 2(b), there are two simultaneous directional transmissions, one from node B to node A and the other from node C to node D. Node C is within the transmission range of node A and node B when it is in omni-mode. However, it still cannot receive the RTS/CTS from node B or A because it is beamforming in a different direction. If the transmission from C to D finishes earlier, and C has packets for A or B, then this transmission can be allowed and it will interfere with the ongoing transmission from B to A.

HOL blocking problem: Many MAC protocols using directional antennas have a common problem which leads to a low network throughput. That is, a node always chooses the first packet in the queue to send it out. When the channel in the outgoing direction of that packet is busy, the transmission will be suspended until the channel becomes idle. In fact, at this time there may be some packets in the queue in whose outgoing direction the channel is idle. If this is the case, the node can instead transmit. Thus the channel can be used more efficiently and the network throughput will be improved.

This paper proposes several MAC protocols that can effectively address these problems and significantly improve the network throughput. We first propose a new MAC protocol called SDMAC, where the communication between a transmitter and a receiver is composed of three processes: Type I DRTS/DCTS Exchange process, Type II DRTS/DCTS Notification process, and DDATA/DACK Transmission process. In the first process, a transmitter and a receiver set up a communication by exchanging Type I DRTS/DCTS (directional RTS/directional CTS), which contain information on their own directional Network Allocation Vectors (DNAV) indicating the channel conditions in different directions. Then, based on this information, by employing a distributed scheduling algorithm, the transmitter and the receiver can negotiate on a short time scale to send out Type II DRTS/DCTS to notify their neighbors of the impending transmission, only in all those directions with expired DNAV. This algorithm works more efficiently when many DNAVs of the sender and the receiver have not expired. It can also ensure the synchronization between the sender and the receiver so that they can beamform to each other to carry out the DATA transmission. At last, DATA and ACK are directionally transmitted, which are denoted by DDATA and DACK, respectively. In SDMAC, packets are transmitted directionally, thus both the spatial reuse and the transmission range can be improved.

In addition to taking advantage of the benefits brought by directional antennas, SDMAC can also address many problems caused by them. In this protocol, every node keeps a deafness table, which contains the deaf nodes and their corresponding periods for being deaf. By putting the sender and the receiver into the deafness table after the reception of Type II DRTS/DCTS, SDMAC can effectively address the deafness problem and new hidden terminal problem. Besides, by putting into the deafness table the sender of Type I DRTS, as well as both the sender and the receiver of Type I DCTS, Type II DRTS/DCTS, DDATA, and DACK, SDMAC can also address the deafness problem due to node mobility. This problem exists because when a node moves around, those which were not deaf to it could become its deaf nodes. Moreover, in SDMAC, nodes set DNAV in a different way so that the spatial reuse can be increased and the interference to the ongoing transmission can be greatly reduced.

We also present two improvements for SDMAC. The first one is to schedule the packets in the queue to avoid the HOL blocking problem. We use a different algorithm from that in [6]. This new scheme is called Queue-SDMAC (Q-SDMAC). Simulation results show that Q-SDMAC can improve the performance of SDMAC significantly. The second one is to relax the assumption in SDMAC and Q-SDMAC: every node knows in which direction to send the packets to their neighboring nodes. Instead, every node caches the neighboring node ID and the incoming direction of packets from that node. This scheme is called Q-SDMAC with cache, where nodes update their caches every time they receive packets from the neighboring nodes. Simulation results show that the performance of Q-SDMAC does not degrade much when using cache to predict the directions of intended receivers in a mobile multi-hop scenario.

The rest of this paper is organized as follows. We present the related work in the next section. In Sect. 3 we introduce our directional antenna model. Section 4 details our proposed protocol SDMAC, and Sect. 5 shows two improvements on SDMAC. The simulation results are shown in Sect. 6. We finally conclude this paper in Sect. 7.

2 Related work

Many MAC protocols for wireless ad hoc networks using directional antennas have been proposed in the past. Ramanathan analyzes in [7] the performance of aggressive and conservative collision avoidance models, with power control and neighbor discovery. In [8], Takai et al. use the directional virtual carrier sensing combined with a DNAV table to increase the spatial reuse of the network. Choudhury et al. proposed a MAC protocol in [3] using multi-hop RTSs to establish links between distant nodes, and then transmitting CTS, DATA, and ACK over a single hop. In these papers, the main objective is to improve the network throughput by increasing spatial reuse of the network. They all have the deafness problem and the new hidden terminal problem.

Ko and Vaidya propose DMAC in [9]. They use a directional RTS (scheme 1) or an omnidirectional RTS if all antennas sense an idle channel (scheme 2). The CTS frames are always sent omnidirectionally. It is assumed in the protocol that each node knows the exact locations of other nodes and each node transmits signals based on the known physical positions of the intended receiver. In DMAC, the neighboring nodes of senders suffer from the deafness problem. Since the CTS is sent out omnidirectionally, the new hidden terminal problem cannot be well addressed.

Nasipuri et al. propose in [10] a MAC protocol using omnidirectional RTS/CTS proceeding the directional DATA transmission. The nodes in the network do not need to know the physical locations of their neighboring nodes. This protocol does not fully utilize the advantages of directional antennas because the spatial reuse is still low and the transmission range is not improved. There is no deafness problem, but the new hidden terminal problem is not well addressed.

Korakis et al. propose Circular DMAC in [11] to address the deafness problem, which has significant constant overhead and the neighboring nodes of the receiver still suffer from deafness problem. Besides, the CTS may not be received after the circular transmission of RTS, while the neighboring nodes of the transmitter still keep silent for a long period. This results in a low channel efficiency. Moreover, the new hidden terminal problem due to the use of directional antennas is not addressed in this protocol.

Choudhury and Vaidya study the deafness problem in [2] and propose a tone-based solution. They split the channel into two sub-channels. One channel is used to transmit RTS/CTS/DATA/ACK and the other one is used to transmit busy tones. This protocol can achieve better performance at the cost of an increased complexity of the protocol.

Recently, Li and Safwat propose DMAC-DA in [12] to address the deafness problem. However, it still has a large constant overhead and there could be many interferences to the ongoing transmission. Gossain et al. propose MDA in [13]. In this protocol, the sweeping RTS/CTS do not cover all neighborhoods of the sender or receiver, so the deafness problem is not effectively addressed. Besides, in order to achieve synchronization between a sender and a receiver to carry out a DATA transmission, the sweeping process has to incur significant overhead.

Our proposed protocol SDMAC implements a novel distributed algorithm to control the transmission of Type II DRTS/DCTS. In this algorithm, Type II DRTS/DCTS can cover all the neighborhoods of a sender and a receiver, and can be sent out simultaneously with no overlapping transmission areas. The overhead can be greatly reduced when the DNAVs in some directions other than that of ongoing transmission have not expired. To provide a more complete solution, SDMAC can effectively address the deafness problem and the new hidden terminal problem with low overhead. SDMAC also addresses the deafness problem due to node mobility, which has not been well addressed in the past.

Moreover, SDMAC uses a method different from all the related work above to set the DNAV: nodes receiving Type I DRTS/DCTS set DNAV in the direction through which they receive those packets; nodes receiving Type II DRTS/DCTS set DNAV in the same direction as that of the ongoing transmission. Thus the interference to the ongoing transmission can be reduced. In SDMAC, the transmission range of omnidirectional packets are set to the same as that of directional packets so that the routing protocols such as AODV and DSR can work more efficiently. All these lead to improved network throughput.

In addition, HOL blocking problem may pose a severe problem with directional antennas. Kolar et al. propose an algorithm in [6] to avoid this problem. In their algorithm, the MAC layer always chooses from the queue the packet with the least wait time for transmission. Since the omnidirectional packets have the longest waiting time, they have to be sent out with lowest priority. In this paper, we address HOL blocking problem by proposing Q-SDMAC, which is an improved version of SDMAC. In Q-SDMAC, when the HOL blocking happens, we choose the first nonblocking unicast packet to send. If there are some broadcasting packets before the first nonblocking unicast packet, we send the first broadcasting packet instead. We assign a higher priority to the broadcasting packets because they are usually routing control packets like RREQ. Thus Q-SDMAC can adapt to the topology change more quickly than the algorithm in [6]. Moreover, the computational load of Q-SDMAC is merely about half of that in [6].

Finally, in SDMAC and Q-SDMAC, we assume that the relative directions of neighboring nodes are known as many other directional MAC protocols such as [9, 12]. We relax this assumption by proposing Q-SDMAC with cache (an improved version of Q-SDMAC), where every node caches the IDs and the corresponding directions of the neighboring nodes. Simulation results show that the performance of Q-SDMAC does not degrade much when using cache in a mobile multi-hop scenario.

3 Directional antenna model

In this section we present the directional antenna model we will use for our study. The gain of an antenna in the direction $\vec{d} = (\theta, \phi)$ is given by [14]

$$G(\vec{d}) = \eta \cdot \frac{U(\vec{d})}{U_{ave}}, \tag{1}$$

where $U(\vec{d})$ is the power density in the direction $\vec{d}$, U_{ave} is the average power density over all directions, and η is the efficiency of the antenna which accounts for losses. Clearly, we can see that an omnidirectional antenna has a gain of 0 dBi and a directional antenna has a higher gain than that. Due to the higher gain and less interference, when it is beamforming in a specific direction, a directional antenna can give a longer transmission range than omnidirectional antennas.

Let P be the transmitting power, and S the surface area of the sphere with center at the transmitter and radius R. Then the surface area A on the sphere for a beamwidth of α is $2\pi\ r\ h$, where r is $R \sin\frac{\alpha}{2}$, and h is $R(1-\cos\frac{\alpha}{2})$. By the definition of antenna gain, when we neglect the side lobe directional antenna gain, we have [7]

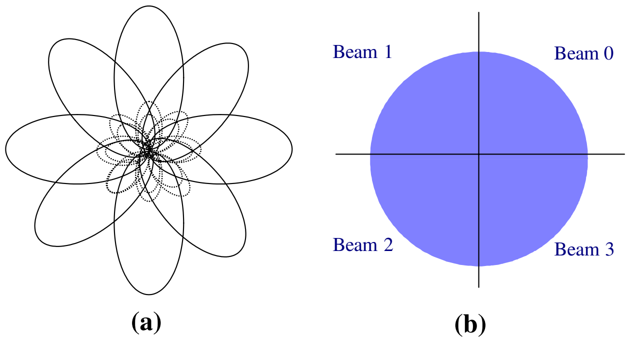
$$G_m = \frac{P/A}{P/S} = \frac{4\pi R^2}{2\pi R^2 \sin\frac{\alpha}{2}(1-\cos\frac{\alpha}{2})} = \frac{2}{\sin\frac{\alpha}{2}(1-\cos\frac{\alpha}{2})} \tag{2}$$

and when we consider the side lobe gain, we have

$$G_m \cdot U_{ave} \cdot A + G_s \cdot U_{ave} \cdot (S - A) = \eta \cdot P \tag{3}$$

where G_m and G_s are the main lobe directional antenna gain and the side lobe directional antenna gain, respectively.

There are three primary types of directional antenna systems—switched beam antenna system, steered beam antenna system, and adaptive antenna system [15]. In this study, we use the switched beam antenna system, which consists of several highly directive, fixed, pre-defined beams and each transmission uses only one of the beams. One such antenna with eight beam directions is shown in



(a) (b)

Fig. 3 (**a**) Switched beam antenna system. (**b**) Our switched antenna model

Fig. 3(a). This system detects the received signal strength and chooses from one of the beams that gives the highest received power or SINR (Signal-to-Interference and Noise Ratio). Thus, we can easily get the beam direction in which we receive the signal.

Our study assumes that there are N beams exclusively and collectively covering all directions in a switched beam antenna system. We also assume that when a directional antenna is engaged in transmission in one direction, the signal arriving in other directions will cause little interference to the ongoing transmission. Such an antenna model with four beams is shown in Fig. 3(b). We obtain from Eqs. 2 and 3 that the maximum main lobe directional gain G_m is approximately 68.66 (18.4 dBi) with eight beams, and 9.65 (9.8 dBi) with four beams.

4 The proposed protocol: SDMAC

4.1 Protocol description

This section details the proposed protocol: Selectively Directional MAC (SDMAC). In this protocol, every node keeps two tables: a deafness table and a DNAV table. All nodes send and receive non-broadcasting packets directionally when they are carrying out transmissions and listen to the channel omnidirectionally when they are not. We assume every node knows the relative directions of its neighboring nodes. This kind of information can be achieved through the GPS system or by some neighbor discovery process [7, 16, 17].

The timeline of SDMAC operation is shown in Fig. 4. The communication between a transmitter and a receiver has three processes: Type I DRTS/DCTS Exchange process, Type II DRTS/DCTS Notification process, and DDATA/DACK Transmission process, which are described in details in the following.

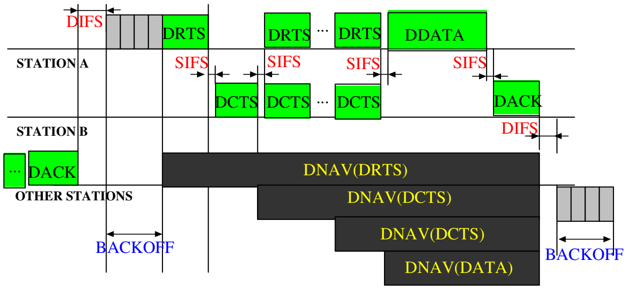


Fig. 4 Timeline of SDMAC operation

4.1.1 *Type I DRTS/DCTS Exchange*

The sender first sends Type I DRTS directly to the receiver. As shown in Table 1, Type I DRTS frame has two more fields than the RTS frame defined in IEEE 802.11 MAC protocol. One field called “Outgoing Beam” contains the outgoing beam number which is one byte long. It indicates the beam that the sender uses to transmit Type I DRTS to the receiver. The other field called “Beam Status” describes the traffic status of the sender in all the beam directions, which refer to the actual directions with respect to a common reference direction for all nodes. One bit stands for one direction. For example, the first bit stands for direction 0, and the Nth bit stands for direction $N - 1$. For the nth bit where $1 \leq n \leq N$, 1 means that DNAV[n] has not expired and the traffic is busy in direction n, and 0 means that DNAV[n] has expired and there is no traffic in direction n. In this scheme, this field takes one byte, which can be adjusted according to the number of beams each node has. The duration field of Type I DRTS is set according to (4).

$$Duration_{rts1} = 3 * SIFS + T_{cts1} + T_{data} + T_{ack} \tag{4}$$

where T_{cts1}, T_{data}, T_{ack} represent the transmission times of Type I DCTS, DATA, and ACK, respectively.

The receiver then responses with Type I DCTS using the beam on which it receives Type I DRTS destined to itself. Type I DCTS frame has the same format as Type I DRTS frame (Table 1). The “Outgoing Beam” field of Type I DCTS indicates the beam on which Type I DCTS is transmitted. The “Beam Status” field describes the traffic status of the receiver in all the beam directions.

The duration field of Type I DCTS is set according to (5).

$$Duration_{cts1} = Duration_{rts1} - T_{cts1} + M * SIFS + M * T_{rts2} \tag{5}$$

where M is the number of Type II DRTS/DCTS that will be transmitted according to the distributed algorithm, i.e., *TBeam* in Fig. 6. The Type I DRTS/DCTS frame format is shown in Table 1.

Table 1 Type I DRTS/DCTS frame format

Frame Control	Duration	Receiver Address	Transmitter Address	Outgoing Beam	Beam Status	Frame Check

4.1.2 *Type II DRTS/DCTS Notification*

After the Type I DRTS/DCTS exchange process, both the sender and the receiver will know each other's beam status. Based on this information, the sender and the receiver make their own decisions on the schedule of sending Type II DRTS/DCTS without collision, respectively and simultaneously. Here, no collision means that the other nodes will not receive Type II DRTS and DCTS at the same time so that each time they are able to receive Type II DRTS or Type II DCTS successfully. Then, according to the schedule, the sender and the receiver send Type II DRTS/DCTS, respectively, counterclockwise in the directions where the DNAV has expired. In those directions where the channel is busy, the sender and the receiver do not spend any time and they just skip those directions. A distributed scheduling algorithm is implemented to make the sender and the receiver spend a short time on this notification process. The details of this algorithm will be discussed later.

As shown in Table 2, Type II DRTS and DCTS frames have the same format, so they have the same transmission time. Type II DRTS/DCTS frames have one more field than the RTS frames defined in IEEE 802.11 MAC protocol, which is called "Outgoing Beam". It indicates the beam on which Type II DRTS or DCTS is transmitted. The duration field of the *k*th DRTS/DCTS frame is set according to (6) and (7).

$$\begin{aligned} Duration_{rts2} =& (M-k+2)*SIFS+(M-k)*T_{rts2} \\ &+T_{data}+T_{ack} \end{aligned} \tag{6}$$

$$\begin{aligned} Duration_{cts2} =& (M-k-1)*SIFS+(M-k)*T_{cts2} \\ &+Duration_{rts1}-T_{cts1} \end{aligned} \tag{7}$$

where M is equal to *TBeam* in our distributed algorithm shown in Fig. 8, $1 \le k \le M$, and N is the number of beams. T_{rts2} and T_{cts2} represent the transmission time of Type II DRTS/DCTS, respectively.

4.1.3 *DDATA/DACK Transmission*

The distributed scheduling algorithm mentioned above can ensure that the sender and the receiver can beamform toward each other at the same time to prepare for the data transmission. The transmission ends when the sender receives directional ACK (DACK) from the receiver. The duration field of DDATA frame is according to (8).

$$Duration_{data} = T_{ack} + SIFS \tag{8}$$

4.2 Key techniques

In this subsection we detail some key techniques used in the protocol SDMAC.

4.2.1 *Differentiation of two kinds of DRTS/DCTS*

As mentioned before, in this protocol, there are two types of DRTS/DCTS: Type I DRTS/DCTS is used to initiate the transmission between the sender and the receiver, and Type II DRTS/DCTS is used to notify their neighboring nodes of the forthcoming data transmission. For Type I and Type II DRTS/DCTS frames, we set the "Receiver Address" field and the "Transmitter Address" field to the MAC address of the frame receiver and that of the frame sender, respectively. We need to differentiate two kinds of DRTS/DCTS because nodes act differently after receiving these two kinds of frames.

4.2.2 *Transmitting Type II DRTS/DCTS simultaneously without collision*

Assume node A and node B use beam X and beam Y, respectively, to exchange the Type I DRTS/DCTS successfully, and then, A and B use beam X' and Y', respectively, to send Type II DRTS/DCTS to notify their neighbors of the forthcoming transmission. We say beam X' of node A and beam Y' of node B collide if A's transmission area using beam X' and B's transmission area using beam Y' overlaps.

Consider the example shown in Fig. 5, where A is the sender and B is the receiver. If A transmits Type II DRTS on beam X' and B transmits Type II DCTS on beam Y' at the same time, node C will receive both packets because it

Table 2 Type II DRTS/DCTS frame format

Frame Control	Duration	Receiver Address	Transmitter Address	Outgoing Beam	Frame Check

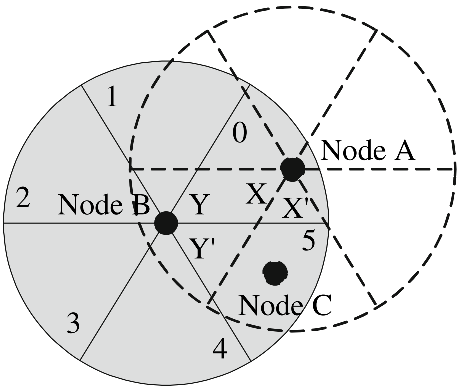


Fig. 5 An example for transmitting Type II DRTS/DCTS simultaneously without collision

is listening to the channel omnidirectionally. In this situation, node C cannot receive any packet successfully and we say beam X' of node A and beam Y' of node B collide. Since node C does not know the impending transmission between node A and node B, it will be able to send packets to these two nodes, and then the deafness problem arises. As a result, this kind of collision should be avoided to ensure that the neighboring nodes can receive Type II DRTS or DCTS successfully.

Design goal: We need to design a distributed algorithm to make the sender and the receiver send Type II DRTS/DCTS simultaneously without collision. The overhead due to transmitting Type II DRTS/DCTS should be small. The transmission of Type II DRTS/DCTS should cover all the neighborhood of both the sender and the receiver. The sender and the receiver should beamform to each other for DATA transmission after the transmission of Type II DRTS/DCTS.

Observation: If node A and B use beam X and beam Y, respectively, to exchange the Type I DRTS/DCTS, then node A can conclude that beam Y' of node B and beam X' of node A collide if $(Y'-X')(Y'-Y)\leq 0$, where $X, X', Y, Y' \in [0, N-1]$. Similarly, node B can conclude that beam X' of node A and beam Y' of node B collide if $(X'-Y')(X'-X)\leq 0$. This can be seen clearly from Fig. 5.

A distributed scheduling algorithm: As mentioned before, in Type I DRTS/DCTS Exchange process, the sender and the receiver check the traffic status in all directions of their own neighborhoods, put the information in “Beam Status” field of Type I DRTS/DCTS, and then exchange the packets. Thus both the sender and the receiver know the traffic status in all directions of each other’s neighborhood. In Type II DRTS/DCTS Notification process, the sender and the receiver start to transmit Type II DRTS/DCTS, respectively, counterclockwise beginning from the direction next to the one used to exchange Type I DRTS/DCTS. Their transmissions of Type II DRTS/DCTS are based on the following decision rule:

1. When one node senses there is no traffic in beam direction X' and the other node senses there is no traffic in beam direction Y', they transmit simultaneously if beam X' and beam Y' do not collide. But if beam X' and beam Y' collide, the node that has swept fewer directions sends first while the other node does not transmit at the same time. We should notice that there is no possibility that two nodes have swept the same number of beams when the two beams they are checking collide.
2. When one node has no traffic in beam direction X' while the other node has swept $N-2$ beam directions, and is trying to transmit on the last beam direction but finds the channel is busy, then the first node transmits in beam direction X' and the second node waits on the last beam direction.
3. When both nodes have finished sweeping all the other $N-1$ beam directions except for the one used to exchange Type I DRTS/DCTS, Type II DRTS/DCTS Notification process terminates and the DDATA/DACK Transmission process follows.

The details of the algorithm is shown in Fig. 6. We further explain the algorithm by using a simple example. In Fig. 5, assume each node has six beam directions, and node A and B use beam 3 and beam 0, respectively to communicate with each other. We also assume the beam status of A is 011000, and that of B is 011100, both from direction 0 to direction 5. So according to our algorithm, in time slot 1 A will transmit on beam 4 while B does not transmit. In time slot 2, A transmits on beam 5 while B transmit on beam 4. In time slot 3, A transmits on beam 0 while B transmits on beam 5. Then Type II DRTS/DCTS Notification process terminates, A beamforms to direction 3, B beamforms to direction 0, and DDATA/DACK transmission process begins. Here, one time slot stands for the transmission time of one Type II DRTS/DCTS frame.

The comparison with other algorithms: We use the same example to compare our algorithm with other algorithms. In the algorithm in [7, 12], A and B will spend five time slots on the transmission of RTS/CTS, which is 66.7% more than the time spent in our algorithm. In the algorithm in [13], the process of RTS/CTS transmission also takes three time slots: in time slot 1, A transmits on beam 4 while B does not transmit; in time slot 2, A transmits on beam 5 while B does not transmit; and in time slot 3, A transmits on beam 0 while B does not transmit. However, the deafness problem still exists since B does not transmit on beam 4 and beam 5 according to the algorithm in [13].

From the above we can see by transmitting Type II DRTS/DCTS, our distributed scheduling algorithm can

Fig. 6 Pseudo-code of the distributed scheduling algorithm

```
INPUTS  N, SBeams[], Sout, RBeams[], Rout
OUTPUTS: TBeam
% 'N' is the total number of beams each node has. 'SBeams' and 'RBeams'
are the beam status of the sender and the receiver respectively, 1
for busy and 0 for idle in that direction. 'Sout' and 'Rout' are the
beams sender and receiver use to transmit Type I DRTS/DCTS respectively.
TBeam is the total number of time slots. Here, one time slot is actually
the transmission time for one DRTS/DCTS.

begin
  SBeams[Sout] = 1, RBeams[Rout] = 1, Scount = 0, Rcount = 0, TBeam
= 0
  sb = (Sout + 1)%N, rb = (Rin + 1)%N
L: while ((rb != Rin or sb != Sout) and (Rcount < N or Scount < N))
    if ((RBeams[rb] == 1) and (rb != Rout))
      rb = (rb + 1)%N, Rcount++
      goto L
    end if
    if ((SBeams[sb] == 1) and (sb != Sout))
      sb = (sb + 1)%N, Scount++;
      goto L
    end if
    if ((SBeams[sb] == 0) and (RBeams[rb] == 0))
      if (sb and rb collide)
        if (Scount < Rcount)
          In time slot TBeam  sender transmits on beam sb while
          receiver does not transmit.
          TBeam++, Scount++, sb = (sb + 1)%N
          goto L
        else
          In time slot TBeam  sender does not transmit while
          receiver transmits on beam rb.
          TBeam++, Rcount++, rb = (rb + 1)%N
          goto L
        end if
      else
        In time slot TBeam, sender transmits on beam sb while
        Receiver transmits on beam rb.
        TBeam++, Scount++, Rcount++
        sb = (sb + 1)%N, rb = (rb + 1)%N
        goto L
      end if
    end if
    if ((SBeam[sb] == 0) and (rb == Rout))
      In time slot TBeam  sender transmits on beam sb while
      receiver does not transmit.
      TBeam++, Scount++, sb = (sb + 1)%N
      goto L
    end if
    if((RBeams[rb] == 0) and (sb = Sout))
      In time slot TBeam  sender does not transmit while
      receiver transmits on beam rb.
      TBeam++, Rcount++, rb = (rb + 1)%N
      goto L
    end if
  endwhile
end begin
```

cover all the areas where deafness problem may exist. More importantly, it can also greatly reduce the overhead due to the use of Type II DRTS/DCTS.

4.2.3 *Deafness avoidance*

Every node keeps a deafness table (Table 3) containing deaf nodes and their corresponding periods for being deaf.

1. When a node receives Type II DRTS/DCTS, it puts both the sender and the receiver of the frame into Table 3 (deafness table) because the communication between the sender and the receiver has been set up. For example, assume that there is a transmission between node A and node B in Fig. 7. If node C is in position C_2 or C_3 initially, after it receives Type II DRTS/DCTS, it will put both A and B into Table 3 to temporarily block the transmission to these two deaf nodes. The deafness periods upon receiving Type II DRTS/DCTS are set according to the duration field of the frame.
2. When a node receives Type I DRTS, we contend that it is not enough for the node to only set DNAV in the direction where it receives the packet, which is what the previous works do. In the example discussed above, assume node C is in position C_1 initially, and then it moves to position C_2 or C_3 after receiving Type I DRTS from node A. In this case, setting DNAV after the reception of Type I DRTS/DCTS cannot prevent

Table 3 Deafness table

Node ID	Deafness period

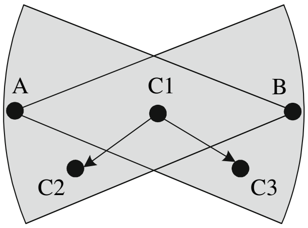


Fig. 7 An example for illustrating deafness avoidance when nodes are moving

node C from sending packets to node A, which is in a new beam direction of node C. So it is necessary for C to consider both the sender and the receiver of Type I DRTS to be possible deaf nodes. In fact, it should put only the sender of Type I DRTS instead of both the sender and the receiver of Type I DRTS into Table 3 (deafness table). This is because the reception of Type I DCTS, instead of the reception of Type I DRTS, means the handshake of Type I DRTS/DCTS between sender and receiver is successful. If a node receives Type I DRTS from the sender, it may not be able to receive Type I DCTS from the destined receiver due to DRTS collision there. So it should not put the receiver in the deafness table. On the other hand, nodes receiving Type I DCTS should put both the sender and the receiver into Table 3 (deafness table). The deafness periods upon receiving Type I DRTS/DCTS is set as follows:

$$Deafness_{rts1} = T_{cts1} + N * T_{rts2} + (N + 1) * SIFS \tag{9}$$

$$Deafness_{cts1} = N * T_{cts2} + N * SIFS \tag{10}$$

where N is the number of beams. So, if a node A transmits Type I DRTS to node B and B transmits Type I DCTS in response, a node C receiving Type I DRTS and DCTS will consider A and B to be deaf nodes until they finish transmitting Type II DRTS/DCTS, which takes a period less than that for transmitting N Type II DRTS/DCTS according to our algorithm. Then the transmission of DDATA from A to B will make C consider them to be deaf nodes again. Thus the nodes receiving Type I DRTS/DCTS will not transmit to A and B during the transmission between them. In the other case, i.e., if A transmits Type I DRTS to node B and B has no response, C will consider only A to be a deaf node because actually B may not be a deaf node to C. If C has packets for A, it will just wait for a period of $Deafness_{rts1}$. In this case, The channel is not wasted much.

3. When a node receives DDATA or DACK, it puts both the sender and the receiver into the deafness table in addition to setting DNAV in the direction where it receives the packet. Because it may move into other areas during the transmission, where the setting of DNAV does not prevent it from transmitting to the sender or the receiver. The deafness periods upon receiving DATA or ACK are set according to the duration field of the frame.

Therefore, our protocol can greatly alleviate the deafness problem and can also address the deafness problem caused by node mobility.

4.2.4 *A new way of setting DNAV*

Every node also keeps a DNAV table, as shown in Table 4. When a node correctly receives Type I DRTS/DCTS, DDATA or DACK, it sets the DNAV in the direction in which it receives the packet. When a node correctly receives Type II DRTS/DCTS, it sets the DNAV in the same direction as that indicated by the “Outgoing Beam” field of the frame. This is because we want to block the transmission in the same direction as that of the DDATA or DACK transmission to avoid the possible collision at the receiver or the sender. The hidden terminal problem due to the asymmetry in gain can also be addressed by using this method.

When a node receives a packet with collision or error, it then sets the DNAV in the way defined in IEEE 802.11. The values of DNAV, upon correctly receiving Type II DRTS/DCTS, DATA and ACK, are set according to the duration field of the frame. Different from that, the values of DNAV, upon receiving Type I DRTS/DCTS, are set as follows:

$$DNAV_{rts1} = T_{cts1} + N * T_{rts2} + (N + 1) * SIFS \tag{11}$$

$$DNAV_{cts1} = N * T_{cts2} + N * SIFS \tag{12}$$

where N is the number of beams.

When a node wants to send packets using beam M, it first checks whether DNAV[M] in Table 4 has expired. If not, the node cannot transmit in that direction. Otherwise, it checks Table 3 to see whether the intended destination node is in the table. If so, it will not transmit. Otherwise, the node can transmit using beam M.

Table 4 DNAV table

DNAV[1]	DNAV[2]	DNAV[3]	...	DNAV[N]

4.2.5 Tuning the power

In this protocol, we use an enhanced antenna gain for directional transmissions in order to have a larger directional transmission range. In this way, the average number of end-to-end hops can be reduced and the end-to-end throughput can be increased. Moreover, the routing protocols such as Ad-hoc On-demand Distance Vector Routing (AODV) and Dynamic Source Routing (DSR) can be used to find a path between two nodes by broadcasting Route Request Packets (RREQ). Since we use a larger antenna gain for directional transmission and a smaller antenna gain for omnidirectional transmission, the transmission range of broadcasting packets will be smaller than that of data packets. Then the paths found by these routing protocols may not be the shortest paths. As a result, in the protocol we increase the transmitting power for omnidirectional transmissions so that they have the same transmission range as that of directional transmissions.

5 Two improvements on SDMAC

5.1 Scheduling the packets in the queue

As mentioned in [18, 19], HOL blocking problem together with the random nature of the contention-based MAC protocols may result in serious instability and unfairness problem in the conventional wireless networks using omnidirectional antennas. Moreover, it is worthwhile to notice that it is also a serious problem with the use of directional antennas.

Kolar et al. propose an algorithm in [6] to avoid this problem. In their algorithm, the MAC layer always chooses from the queue the packet with the least wait time for transmission. The omnidirectional packets have the largest wait time because they must wait for all the directions to be clear. Thus, directional packets have a higher priority to be transmitted. In this paper, we address this problem by proposing Q-SDMAC, which is an improved protocol based on SDMAC. In Q-SDMAC, when the HOL blocking happens, we choose the first nonblocking unicast packet to send. If there are some broadcasting packets before the first nonblocking unicast packet, we send the first broadcasting packet instead. We assign a higher priority to the broadcasting packets because they are usually routing control packets like RREQ. Thus, Q-SDMAC can adapt to the topology change more quickly than the algorithm in [6]. Besides, if the channel is busy in all directions, or if there is no nonblocking unicast packet in the packet queue, we send out the first packet in the queue after the channel that packet needs to use is idle. In our protocol, we choose not to send out the packet with least waiting time since that may lead to the unfairness problem. Moreover, our algorithm is more computational efficient. Assume there are n packets in the queue. According to the algorithm in [6], every node has to compare the wait time of the packets in the queue for $n-1$ times to choose a packet to transmit. However, in Q-SDMAC, on average each node only needs to do the comparison for $(1+2+\cdots+n-1)/(n-1)$, i.e., $n/2$ times, which is merely about half of that in [6]. The flow diagram of the algorithm is shown in Fig. 8.

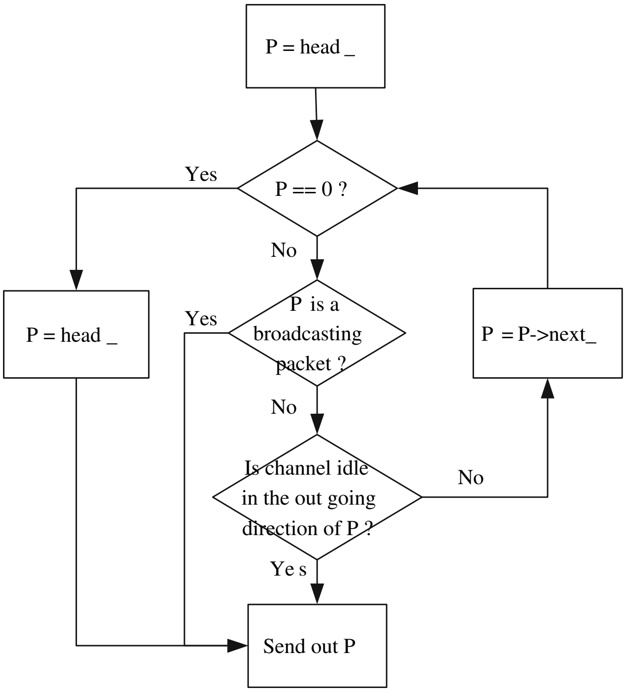


Fig. 8 Flow diagram of the algorithm to address the HOL problem. head_ is the pointer pointing to the first packet in the queue. Every packet has a pointer next_ pointing to the next packet in the queue

5.2 Relaxing the assumption in SDMAC

In SDMAC and Q-SDMAC, we assume that the relative directions of neighboring nodes are known. Many other directional MAC protocols such as [9, 11, 12] also have this assumption. In this paper, we propose Q-SDMAC with cache (a scheme improved on Q-SDMAC) to relax this assumption, where every node caches the neighboring node ID and the incoming direction of packets from that node, as shown in Table 5. The cache is updated every time the node receives a packet from its neighboring nodes. Before a node sends a packet, it checks its cache to see whether there is a record for the intended receiver. If so, the sender can directly send the packet in that direction. If not, the sender transmits RTS omnidirectionally and waits for the directional CTS from the receiver to determine which direction to use to send the DATA.

Table 5 Cache components

Node ID	Incoming Beam number

6 Performance evaluation

6.1 One-hop scenarios

We first use NS2 to simulate SDMAC and compare its performance with DMAC [9], Circular-DMAC [11] and DMAC-DA [12] in one-hop scenarios. Each node is equipped with a switched directional antenna with four beam directions whose height is set to 1.5 m. The main lobe antenna gain and the side lobe antenna gain are set to 9.5 dBi, and $-\infty$ dBi, respectively. The path loss model is set as the two-ray model. The data rate is set to 2 Mbps and the basic data rate is set to 1 Mbps. The packet size is 512 bytes. We also use the same power level for all three protocols to ensure a fair comparison.

In Scenario 1 shown in Fig. 9, node B is in the transmission range of node A. Node C is in the transmission of B but not in the transmission range of A. It is in the sensing range of node A. There are two flows: node A to node B (Flow 1) and node B to node C (Flow 2). We choose this scenario to compare SDMAC with DMAC [9]. In this scenario, when DMAC is used, RTS is sent directionally and CTS is sent omnidirectionally. So A cannot receive either of the RTS and CTS when B is transmitting to C and it suffers from the deafness problem. However, when using our protocol, A will receive the Type II DRTS sent by B, so it will not transmit to B when it is deaf to A. Simulation results are shown in Fig. 10(a, d). From Fig. 10(a) we can see that the total throughput of SDMAC is more than that of DMAC by up to 30% when the data sending rate is less than 800 kbps. We should notice that it seems that DMAC outperforms SDMAC when data rate is higher than 800 kbps. That is because when data rate is high, Flow 1 of DMAC is severely dominated by Flow 2, which is in fact due to the unfairness of DMAC. This is shown in Fig. 10(d), from which we can also see that our protocol can achieve much better fairness between the two flows.

In Scenario 2 shown in Fig. 9, node B is in transmission range of both node A and node C. Node C is in the transmission range of B and the sensing range of A. The distance between B and C is much greater than that between B and A. There are two flows: node A to node B (Flow 1) and node C to node B (Flow 2). We choose this scenario to compare SDMAC with CDMAC [11]. In this scenario, when using CDMAC, A and C cannot receive the DRTS sent by each other, so they both try to send packets to B. Since the CTS is sent directionally, C will not be able to receive the CTS from B to A when they are transmitting, so it suffers from the deafness problem. Moreover, since the distance between B and C is much greater than that between B and A, then the signal from C will be probably ignored if the signal power is less than that of the signal from A by a threshold. Thus Flow 2 will be dominated by Flow 1 at high date rates. In our protocol, B will send Type II DCTS circularly to solve this problem. The simulation results are shown in Fig. 10(b, e). From these two figures we can see that the total throughput of SDMAC is higher than that of CDMAC by up to 15%. Our protocol also achieves better fairness than CDMAC.

In Scenario 3 shown in Fig. 9, Flow 1 (node A to node B) and Flow 2 (node C to node D) interfere with each other. We choose this scenario to compare SDMAC with DMAC-DA [12]. DMAC-DA allow these two flows going at the same time, which will definitely degrade the network performance. On the contrary, our protocol block the transmissions in parallel directions. So we can achieve better performance. The simulation results are shown in Fig. 10(c, f). From these two figures we can see that the total throughput of SDMAC is higher than that of DMAC-DA by up to 35% and SDMAC can achieve better fairness than DMAC-DA.

6.2 A multi-hop scenario

We then evaluate the performance of our MAC protocol SDMAC in multi-hop networks. We compare SDMAC with IEEE 802.11 [1], DMAC [9], Circular-DMAC [11],

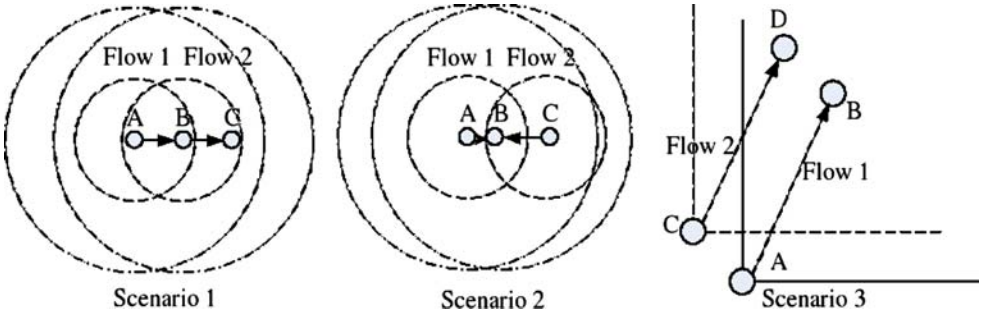
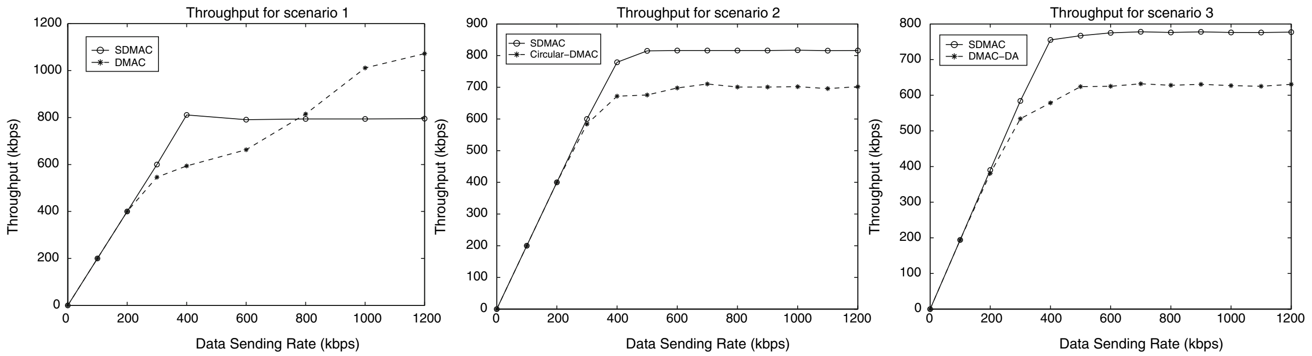


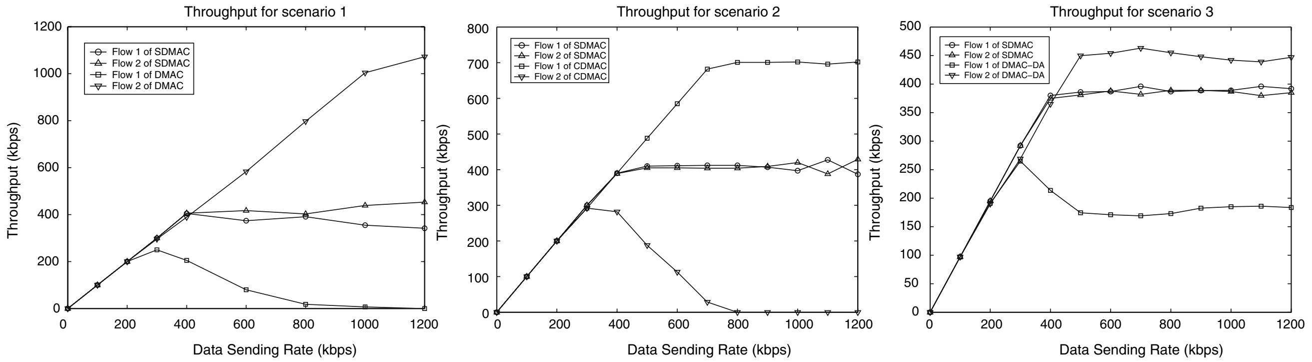
Fig. 9 Scenario 1 is for the comparison between SDMAC and DMAC; Scenario 2 is for the comparison between SDMAC and Circular-DMAC. In this two scenarios, Small circle stands for transmission range and big circle stands for the sensing range. Scenario 3 is for the comparison between SDMAC and DMAC-DA



(a) Aggregated throughput of Flow 1 and Flow 2 in Scenario 1.

(b) Aggregated throughput of Flow 1 and Flow 2 in Scenario 2.

(c) Aggregated throughput of Flow 1 and Flow 2 in Scenario 3.



(d) Throughputs of Flow 1 and Flow 2, respectively, in Scenario 1.

(e) Throughputs of Flow 1 and Flow 2, respectively, in Scenario 2.

(f) Throughputs of Flow 1 and Flow 2, respectively, in Scenario 3.

Fig. 10 Simulation results of Scenarios 1, 2, 3 shown in Fig. 9

and DMAC-DA [12]. We use a 1000 m × 1000 m 2D topology in which there are 50 nodes. Ten nodes are chosen to be CBR (Constant Bit Rate) sources and their destination nodes are randomly chosen. The network uses AODV (Ad Hoc OnDemand Distance Vector Routing) routing protocol. Some simulation parameters are shown in Table 6. Figure 11 shows the simulation result. We can see that SDMAC can achieve higher throughput than all the other MAC protocols. This is because SDMAC can address the deafness problem and the new hidden terminal problem more effectively with lower overhead.

Table 6 Some simulation parameters

Parameters	Value
Channel frequency	2.4 GHz
Data rate	2 Mbps
Packet size	512 bytes
RTS retry limit	7
Main lobe antenna gain	12.0 dBi
Side lobe antenna gain	–∞ dBi
RX threshold	–81.0 dBi
CS threshold	–91.0 dBi
Number of beams	8

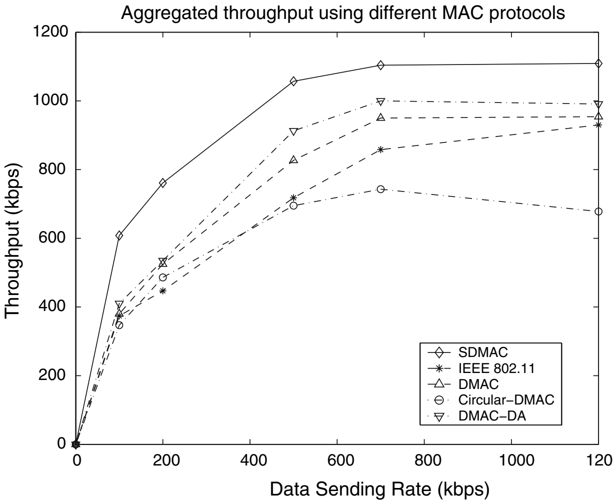


Fig. 11 Compare different MAC protocols in terms of aggregated throughput

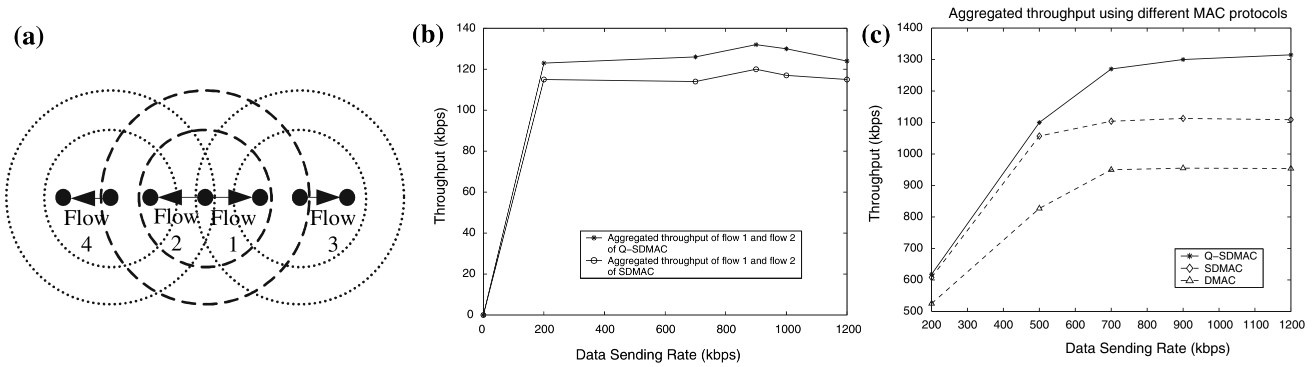


Fig. 12 (**a**) A simple scenario for Q-SDMAC. (**b**) Total throughput of Flow 1 and Flow 2 in (**a**). This is a comparison between Q-SDMAC and SDMAC. (**c**) The comparison of the aggregated throughput of Q-SDMAC, SDMAC and DMAC in a multi-hop scenario

6.3 Simulation results of Q-SDMAC

We first use a simple scenario in Fig. 12(a) to show that by exploring the information of packets in the queue, we can achieve higher throughput. Data sending rate of Flow 3 and Flow 4 are set to 1 Mbps to keep them busy. When transmission on Flow 1 needs to wait until Flow 3 is idle, Flow 2 can be checked. If Flow 4 is idle at this time, transmission on Flow 2 will be carried out first. So total throughput of Flow 1 and Flow 2 can be improved. Simulation result is shown in Fig. 12(b). We can see that the throughput of Q-SDMAC is higher than that of SDMAC by up to 10%.

We then evaluate the performance of Q-SDMAC protocol in multi-hop networks. We compare it with SDMAC and DMAC [9]. The scenario is the same as that in Sect. 6.2. Simulation parameters are the same as that shown in Table 6, except that each node has four, not eight, beam directions, and the main directional antenna gain is set to 9.5 dBi. Thus in the network many nodes will have packets for neighbor nodes in different directions. Figure 12(c) shows the simulation result. We can see that Q-SDMAC can improve the throughput by up to 60% compared to DMAC, and by up to about 15% compared to SDMAC.

6.4 Simulation results of Q-SDMAC with cache

So far the simulations are done with nodes being static. Now we show that Q-SDMAC with cache works well when the nodes in the network are mobile. We use the Random WayPoint (RWP) mobility model in the simulation. The topology is the same as that in Sect. 6.2, and simulation parameters are the same as that shown in Table 6. We compare Q-SDMAC with cache with Q-SDMAC and DMAC, which have exact position information of nodes in the network. Figure 13(a–c) shows the simulation results when the nodes are moving at maximum speed of 1, 3 and 5 m/s respectively. From the figures we can see that Q-SDMAC with cache achieves much better performance than DMAC. The throughput of Q-SDMAC with cache is just a little bit lower than that of Q-SDMAC.

6.5 Impact of the side lobe antenna gain

The simulations above are carried out with the assumption that the directional antennas do not have side lobe antenna gain. This is an ideal assumption by which two nodes are able to send packets directionally at the same time without interfering with each other at all even when they are very close. However, for the real cases, directional antennas do have side lobe antenna gain. We redo the simulation in Sect. 6.2 by changing the side lobe antenna gain to nonzero values. Figure 14(a, b) shows the results when side lobe gain is –20 dBi and –30 dBi, respectively. We can see that compared to using directional antennas without side lobe gain, using directional antennas with side lobe gain leads to degraded network throughput performance. This is because when directional antennas with side lobe gain are used, a node transmitting packets in one direction may be affected by its neighboring nodes in other directions if they are close enough. Comparing Fig. 14(a, b), we can find that a higher side lobe gain will result in a more degraded network throughput, which is consistent with our intuition.

7 Conclusion

Directional antennas can provide larger transmission range and higher spatial reuse. However, using directional antennas in wireless networks also causes many problems, such as deafness problem, new hidden terminal problem, and HOL blocking problem, which make it less effective in improving the network performance if not well addressed.

In this paper, we propose new directional MAC protocols to address these problems. Basically, deafness problem and new hidden terminal problem are incurred because

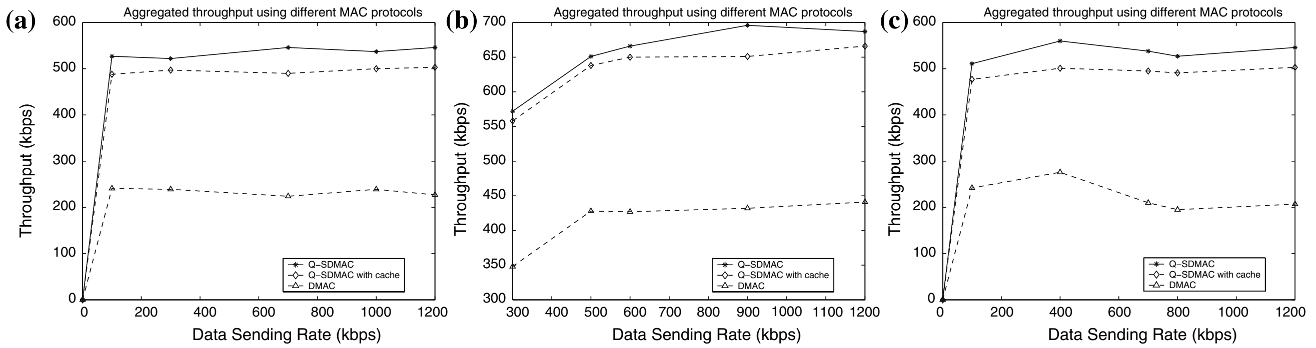
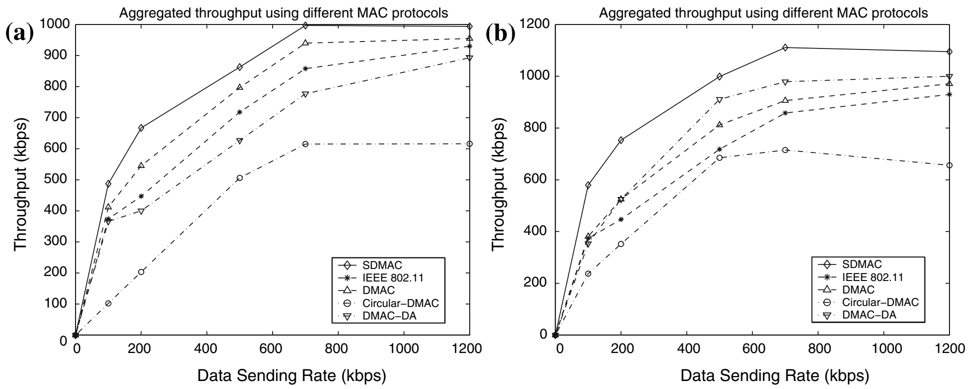


Fig. 13 The scenario is that 50 nodes are deployed randomly in a 1000 m × 1000 m area. Ten flows are randomly set up. This is the comparison of the aggregated throughput of Q-SDMAC with cache, Q-SDMAC and DMAC. (**a**), (**b**), and (**c**) are the results when the node are moving at a maximum average speed of 1, 3, and 5 m/s, respectively

Fig. 14 Compare different MAC protocols in terms of aggregated throughput. (**a**) and (**b**) are the results when side lobe antenna gain is –20 dBi and –30 dBi, respectively



directional transmission is not known to all the neighboring nodes of a transmitting and a receiving node. As a result, in our proposed protocol SDMAC, the sender and the receiver transmit newly introduced packets called Type II DRTS/DCTS to notify their neighboring nodes of the impending data transmission. Nodes receiving the notification put the sender and the receiver into their deafness tables, which can prevent them from sending packets to these two nodes. A distributed scheduling algorithm is implemented to reduce the overhead introduced by the transmission of Type II DRTS/DCTS. By doing this, the deafness and new hidden terminal problem can be effectively addressed. Besides, we also address the deafness problem due to mobility.

Moreover, we also propose two improvements on SDMAC. The first one is called Q-SDMAC, which addresses the HOL blocking problem by implementing a new packet scheduling algorithm. It works as follows: when the HOL blocking happens, we choose the first nonblocking unicast packet to send. If there are some broadcasting packets before the first nonblocking unicast packet, we send the first broadcasting packet instead. Broadcasting packets have a higher priority because they are usually routing control packets. Thus we can adapt to the topology change quickly. The second improvement is called Q-SDMAC with cache, which relaxes the assumption of SDMAC and Q-SDMAC that the relative directions of neighboring nodes are known by using caches.

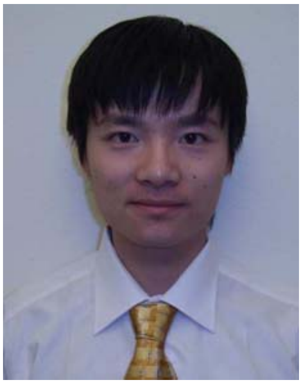
Extensive simulations are carried out. The results show that the proposed protocols can achieve much better network performance than DMAC, Circular DMAC, DMAC-DA and IEEE 802.11. Besides, in the literature, many simulations for directional antennas set the directional antenna side lobe gain to be 0, i.e., $-\infty$ dBi, which is not true in practice. In this paper, we also show some results with nonzero side lobe gain. We find that the higher the side lobe gain, the smaller the network throughput. Thus, side lobe antenna gain does affect the network performance and should be considered in practical design.

Acknowledgments The work of Li, Zhai and Fang was supported in part by U.S. National Science Foundation (NSF) under Grant CNS-0721744 and GrantDBI-0529012. The work of Fang was also supported in part by the National Science Council (NSC), ROC, under the NSC VisitingProfessorship with contract number NSC-96-2811-E-002-010.

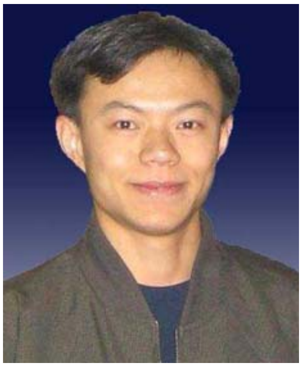
References

1. Wireless lan medium access control (mac) and physical layer (phy) specifications. IEEE Standards Working Group, (1999)
2. Choudhury, R. R., & Vaidya, N. H. (2004). Deafness: A mac problem in ad hoc networks when using directional antennas. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP)*, October 2004, pp. 283–292.
3. Choudhury, R. R., Yang, X., Vaidya, N. H., & Ramanathan, R. (2002). Using directional antennas for medium access control in ad hoc networks. In *Proceedings of the 8th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM)*, September 2002, pp. 59–70.
4. Gossain, H., Cordeiro, C. M., Cavalcanti, D., & Agrawal, D. P. (2004). The deafness problems and solutions in wireless ad hoc networks using directional antennas. In *IEEE Global Telecommunications (GlobeCom) Workshop on Wireless Ad Hoc and Sensor Networks*, November 2004, pp. 108–113.
5. Li, G., Yang, L. L., Conner, W. S., & Sadeghi, B. (2005). Opportunities and challenges in mesh networks using directional anntenas. In *Proceedings of IEEE WiMesh*, September 2005.
6. Kolar, V., Tilak, S., & Abu-Ghazaleh, N. (2004). Avoiding head of line blocking in directional antenna. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN)*, November 2004, pp. 385–392.
7. Ramanathan, R. (2001). On the performance of ad hoc networks with beamforming antennas. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, October 2001, pp. 95–105.
8. Taikai, M., Martin, J., Ren, A., & Bagrodia, R. (2002). Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002, pp. 183–193.
9. Ko, Y., & Vaidya, N. H. (2000). Medium access control protocols using directional antennas in ad hoc networks. In *Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom)*, March 2000, vol. 1, pp. 13–21.
10. Nasipuri, A., Ye, S., & Hiromoto, R. (2000). A mac protocol for mobile ad hoc networks using directional antennas. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, September 2000.
11. Korakis, T., Jakllari, G., & Tassiulas, L. (2003). A mac protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2003, pp. 98–107.
12. Li, Y., & Safwat, A. M. (2005). Efficient deafness avoidance in wireless ad hoc and sensor networks with directional antennas. In *Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Montreal, Quebec, Canada, October 2005, pp. 175–180.
13. Gossain, H., Cordeiro, C. M., Cavalcanti, D., & Agrawal, D. P. (2005). Mda: An efficient directional mac scheme for wireless ad hoc networks. In *Proceedings of IEEE Global Telecommunications (GlobeCom)*, November 2005.
14. Liberti, J., & Rappaport, T. (1999). *Smart antennas for wireless communications*. Prentice-Hall PTR.
15. Janaswamy, R. (2001). *Radiowave propagation and smart antennas for wireless communications*. Boston: Kluwer Academic Publishers.
16. Ramanathan, R., Redi, J., Santivanez, C., Wiggins, D., & Polit, S. (2005). Ad hoc networking with directional antennas: A complete system solution. *IEEE Journal on Selected Areas in Communications, 23*(3), 496–506.
17. Ueda, T., Tanaka, S., Roy, S., Saha, D., & Bandyopadhyay, S. (2004). Location-aware power-efficient directional mac protocol in ad hoc networks using directional antenna. *IEICE Transactions on Communications 2005, E88-B*(3), 1169–1181.
18. Bhagwat, P., Bhattacharya, P., Krishna, A., & Tripathi, S. K. (1996). Enhancing throughput over wireless LANs using channel state dependent packet scheduling. In *Proceedings of IEEE Infocom*, March 1996, vol. 3, pp. 1133–1140.
19. Wang, J., Zhai, H., & Fang, Y. (2004). Opportunistic packet scheduling and media access control for wireless LANs and multi-hop ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'04)*, March 2004, pp. 1234–1239.

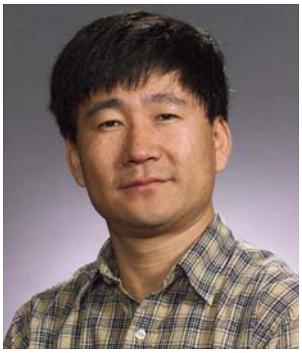
Author Biographies



Pan Li received his B.S. in Electrical Engineering from Huazhong University of Science and Technology in 2005. He is completing his Ph.D. degree in the Department of Electrical and Computer Engineering at University of Florida. His research interests include medium access control, routing algorithms, and performance analysis in wireless networks. He is a student member of the ACM and the IEEE.



Hongqiang Zhai received the Ph.D. degree in Electrical and Computer Engineering from University of Florida in August 2006 and the B.E. and M.E. degrees in Electrical Engineering from Tsinghua University, Beijing China, in July 1999 and January 2002, respectively. He worked as a research intern in Bell Labs Research China from June 2001 to December 2001, in Microsoft Research Asia from January 2002 to July 2002, and in Kiyon Inc. from September 2005 to December 2005. Currently he is a senior member of research staff in Wireless Communications and Networking Department of Philips Research North America. He is a recipient of the Best Paper Award at the 14th IEEE International Conference on Network Protocols (ICNP 2006). His research interests include performance analysis, medium access control, quality of service, fairness, congestion control, routing algorithms and cross-layer design in wireless networks. He is a member of the ACM and the IEEE.



Yuguang Fang received a Ph.D. degree in Systems Engineering from Case Western Reserve University in January 1994 and a Ph.D. degree in Electrical Engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at University of Florida in May 2000 as an assistant professor, got an early promotion to an associate professor with tenure in August 2003 and to a full professor in August 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009. He has published over 200 papers in refereed professional journals and conferences. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He has served on several editorial boards of technical journals including IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing and ACM Wireless Networks. He has also been actively participating in professional conference organizations such as serving as The Steering Committee Co-Chair for QShine 2005, the Technical Program Vice-Chair for IEEE INFOCOM 2005, Technical Program Symposium Co-Chair for IEEE Globecom 2004, and a member of Technical Program Committee for IEEE INFOCOM (1998, 2000, 2003–2007).