Doubling Capacities by a Link-directionality-based Dual Channel MAC Protocol for IEEE 802.11 Ad-hoc Networks

Ping Chung Ng
Department of Engineering Science,
University of Oxford
ping.ng@eng.ox.ac.uk

David J. Edwards
Department of Engineering Science,
University of Oxford
david.edwards@eng.ox.ac.uk

Soung Chang Liew
Dept. of Information Engineering,
The Chinese University of Hong Kong
soung@ie.cuhk.edu.hk

Abstract — **In this paper, we propose a link-directionality-based dual channel MAC protocol in an attempt to double the capacities of networks using the single-channel IEEE 802.11 protocol. When an IEEE 802.11 ad-hoc network achieves capacity *C* by using a single channel, the targeted capacity by using two channels should be $2 \cdot C$. However, most of the multi-channel 802.11 protocols proposed in the literature only appear to be able to achieve less than 60% of the $2 \cdot C$ targeted capacity. Simulations show that our proposed scheme can achieve more than 106% of our targeted capacities, 1.06*$2 \cdot C = 2.12 \cdot C$. We believe this is a first paper in the literature to propose a MAC protocol to transmit RTS/DATA and CTS/ACK of a link on different channels, a key step that yields significant potential for *multiplying* the network capacities of ad-hoc networks.**

Keywords — Wireless Networks, Ad hoc Networks, IEEE 802.11, Capacity, Scalable Performance.

I. INTRODUCTION

When a wireless network uses more channel resources, it should be expected to achieve a proportionally higher network capacity. If an IEEE 802.11 ad-hoc network achieves capacity *C* by using a single channel, the targeted capacity by using *n* channels should therefore be $n \cdot C$. However, most of the multi-channel 802.11 protocols proposed in the literature simply compared their performance with the original single-channel 802.11 protocol, without considering the additional channel resources used. In fact, most of them (e.g., [1],[2]) only appear to be able to achieve less than 60% of the targeted capacities $n \cdot C$. This inefficiency can be attributed to three reasons: i) an additional control channel is used to allocate transmission channels; ii) the overhead incurred by the information added to the packet headers; and iii) the transmissions of RTS/DATA and the receptions of CTS/ACK by a node are assigned to the same channel which limit the potential for simultaneous transmissions (details will be explained in Sections III and IV). In this paper, we attempt to achieve the targeted capacity $2 \cdot C$ of a two-channel system. Our proposed protocol presented does not require i) and ii). In addition, we propose to transmit RTS/DATA and CTS/ACK of a link in separated channels to *scale* the capacity better.

II. RELATED WORK

There are many proposed multi-channel protocols for 802.11 ad-hoc networks in the literature. Reference [3] compared these protocols and classified them into four categories: 1) dedicated control channel, 2) common hopping, 3) split phase, and 4) multiple rendezvous. Our proposed protocol does not belong to these categories. Instead, our scheme assigns transmission channels based on link-directionalities.

References [4],[5] proposed to use a control channel to exchange RTS/CTS packets which contain the channel information. Then, nodes use the agreed data channels to send DATA and ACK packets. These protocols require a separate control channel which does not carry data packets. This significantly increases the overhead incurred by the protocol. For example, if three channels are used, the targeted capacity would be $3 \cdot C$. One of the three channels, however, is assigned as the control channel and this wastes one-third of the data transmission capacity.

References [6],[7] proposed to split the transmission time into two phases: i) control phase and ii) data phase. During the control phase, all nodes switch to the control channel and allocate the transmission channels for the next data phase. These protocols require synchronizations between nodes which are difficult to achieve in distributed ad-hoc networks. In addition, during the control phase, no data can be transmitted in other data channels. This, again, wastes the communications resource.

Another approach is to use frequency hopping [8]. Nodes use pre-assigned hopping patterns to switch channels for transmitting RTS/CTS packets until agreements are made between nodes. Then, they will use the concurred channels for data transmissions. As mentioned in [3], these protocols may incur significant overheads due to the frequent channel switching.

Compared to the above protocols, our proposed scheme does not need i) a dedicated control channel, ii) synchronizations between distributed nodes, and iii) the channel hopping of radio transceivers. This helps to minimize the overhead incurred by the protocol.

III. THE CONCEPT

To avoid simultaneous transmissions that may lead to collisions, the 802.11 protocol uses short request-to-send (RTS) and clear-to-send (CTS) messages to notify other nodes within the virtual carrier-sensing range (*VCSRange*) to update their Network Allocation Vectors (NAV). The NAV includes the duration time of the ongoing transmission. Thus, no other nodes within the *VCSRange* can begin transmissions before the NAV expires. Figure 1a shows an example. Under the 802.11 protocol with RTS/CTS access mode, none of the links B, C or D can transmit at the same time with link A. As a result, only one link inside the *VCSRange* region can transmit at one time. This is because R_A has to receive the DATA packet from T_A while T_A has to wait for the ACK from R_A. Any other simultaneous transmissions within the *VCSRange* region of R_A and T_A in the same channel will lead to collision of the transmission between R_A and T_A. To avoid such collision, the RTS of T_A and the CTS of R_A forewarn links B, C, and D not to transmit before link A transmits its DATA frame.

To overcome the above situation, we can split the transmissions between two nodes of a link into two channels based on their directionalities. Let us consider the case where there are two channels, *s* and *t*. Nodes transmit RTS and DATA in one channel (e.g., channel *s*) as they are in the same direction (from T_A to R_A) while CTS and ACK are transmitted in another channel (e.g. channel *t*). The channels are assigned dynamically based on the directionality, network topology and who else are transmitting the neighborhood. RTS and DATA can be transmitted in either channel *s* or *t*, and thus CTS and ACK will be sent in the other channel (*t* or *s*). The main idea is to allow the simultaneous transmission of another link *i* within the *VCSRange* region of R_A and T_A provided that the transmissions of link *i* do not interfere with the receptions of the ACK on T_A or the DATA on R_A. There are two possible cases:

Case 1: the transmissions of link *i* within the *VCSRange* use a different channel, and thus these do not affect the reception of R_A (T_A) in another channel.

Case 2: the transmissions of link *i* use the same channel as the reception of R_A (T_A) but those transmissions are far enough from R_A (T_A).

For Case 2, let d_{TA-RA} be the distance between T_A and R_A, d_{TB-RA} be the distance between T_B and R_A, and assume the capture threshold (*CPThreshold*) is set to be 10dB. From [9], in a two-ray propagation model, assuming noise is negligible, if the signal-to-interference ratio at R_A is larger than the *CPThreshold*, R_A can capture the signal from T_A when T_B is transmitting. That is,

$$SIR = (d_{TB-RA} / d_{TA-RA})^4 > CPThreshold$$

$$d_{TB-RA} > 1.78 * d_{TA-RA} \qquad (1)$$

In the worst case that T_A and R_A are separated by the maximum transmission range (250m), R_A can capture the signal from T_A if T_B is located at more than 1.78*250m=445m away from R_A. In our simulation, the *VCSRange* is set to be 550m. If T_B can not receive the CTS from R_A, T_B must be far enough so that its signal can not interfere with the reception of signal from T_A at R_A. In next section, our proposed MAC protocol will utilize this property to assign transmission channels for links.

Figure 1b shows the same scenario as Fig. 1a with the channel assignments based on the Cases 1 and 2. Assuming link A is using channel 1 to transmit RTS and DATA from T_A to R_A and another independent channel 2 to send CTS and ACK from R_A to T_A. For link B to transmit simultaneously with link A, we can assign channel 1 for the transmission of RTS and DATA from T_B to R_B. This will not lead to collisions on link A because the signal from T_B is much weaker than the signal from T_A when they reach R_A (Case 2). R_B can then use channel 2 to transmit CTS and ACK. This, again, will not incur collisions on R_A because R_B is using another independent channel for transmissions (Case 1). Similarly, for link C, T_C can use channel 2 to transmit RTS and DATA while R_C can use channel 1 to reply CTS and ACK. For link D, since both T_D and R_D are within the *VCSRange* of link A, link A and D can not transmit at the same time and thus they have to take turns to transmit.

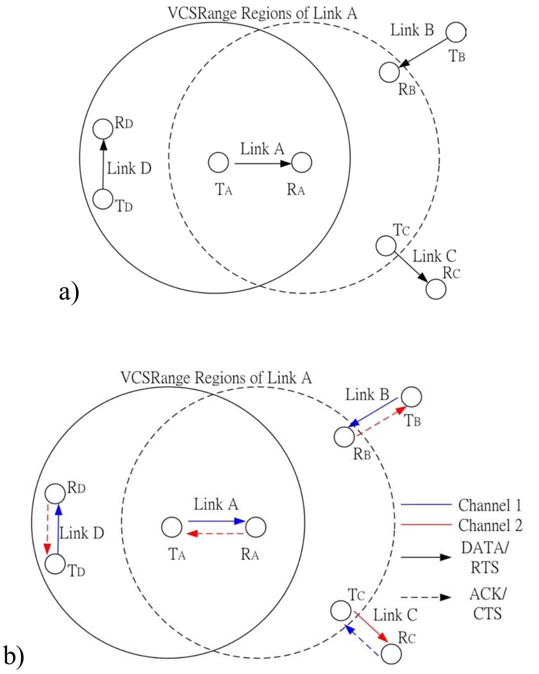


Figure 1. A network topology using a) original 802.11 and b) our proposed scheme

IV. PROPOSED MAC PROTOCOL

In this section, we describe a MAC protocol to achieve the channel assignments as explained in Section III. The protocol assigns the transmission channels of each link based on the availabilities of the receptions of RTS and CTS from other links. The protocol is modified from the original 802.11 MAC protocol and it attempts to seek opportunities for simultaneous transmissions. Assume all nodes use the same power for transmissions and each node has two half-duplex transceivers that are monitoring both channels at the same time. Consider two links, link *i* and link *j*. When a node (e.g., T_i) of link *i* receives the RTS_j but not the CTS_j of another link *j*, it will assign its RTS_i to the same channel as that of RTS_j. Thus, link $i_{T->R}$ can transmit simultaneously with $j_{T->R}$ because receiver R_j is located far enough away from the transmitter T_i (as explained in Case 2 in Section III) and T_j is receiving CTS or ACK in another channel (Case 1 in Section III). Thus, there is no collision between link *i* and link *j*. Similarly, when a node (e.g., T_i) of link *i* receives the CTS_j but not the RTS_j of another link *j*, it will assign its RTS_i to the same channel as that of CTS_j. If a node can receive both the CTS_j and RTS_j of another link *j*, it will fall back to the original 802.11 protocol and will resume transmissions only after the completion of the sensed signal. In this case, links *i* and *j* have to take turns to transmit.

Our proposed MAC protocol constructs a simultaneous transmission table (*SimTable*) in each node based on the receptions of RTS and CTS from other links. Each node attempts to seek opportunities for simultaneous transmissions according to the sensing signals and the records of its *SimTable*. Table I shows an example of the *SimTable* of node T_A of link A in Fig. 2. In the topology of Fig. 2, let us say link B begins the transmission first. T_A receives the CTS from R_B and then updates its $SimTable_{TA}$ (as shown in Table 1) with a new record (the first row). The *CTSChannel* field is set to Channel 2. Since T_A cannot receive the RTS from T_B, the *RTSChannel* field remains *null*. According to $SimTable_{TA}$, T_A

realizes that it can transmit simultaneously with R_B in Channel 2 without interfering T_B as T_B is far enough from T_A. Similarly, T_A receives the RTS from T_C and then updates its $SimTable_{TA}$ with another record (the second row). The *RTSChannel* field is set to Channel 2. Since T_A cannot receive the CTS from T_C, the *CTSChannel* field remains *null*. T_A thus recognizes that it can transmit concurrently with T_C in Channel 2 without intruding R_C. When T_A has a packet to transmit, it compares the sensing signals with its $SimTable_{TA}$. If simultaneous transmissions are allowed, T_A will transmit packets to R_A. Otherwise, it will wait for the completions of the sensing signals and then resume the transmission process.

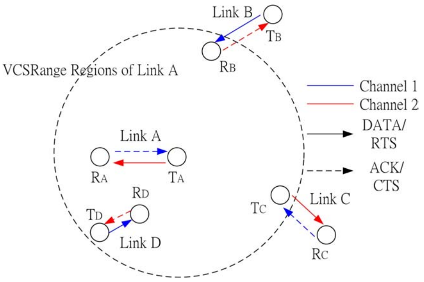


Figure 2. A network topology using our proposed channel assignment scheme

TABLE I. SIMTABLE OF NODE TA IN FIGURE 2

Index	From	To	RTSChannel	CTSChannel
1	T_B	R_B	*null*	2
2	T_C	R_C	2	*null*
3	T_D	R_D	1	2

Consider link D in Fig. 2. T_D sends a RTS to R_D and R_D replies a CTS back to T_D. T_A receives both the RTS and CTS of link D. Thus, the *RTSChannel* and *CTSChannel* fields of the record (the third row in Table I) are set to 1 and 2 respectively. So, in this case, link A cannot transmit at the same time with link D and they must take turns to transmit.

V. ALLOWING TRI-LINK SIMULTANEOUS TRANSMISSIONS

In Section III, we have considered pair-wise simultaneous transmissions. This can significantly boost the network throughput. In an earlier work [10], we showed that a dual channel MAC protocol based on pair-wise simultaneous transmissions could achieve more than 78% of our targeted capacities, *0.78*2C=1.56C*. However, pair-wise simultaneous transmissions cannot achieve the targeted capacity $2 \cdot C$ due to the overheads induced by the protocol. To further boost the capacity to reach this target, we could further improve the protocol by allowing tri-link simultaneous transmissions. In this section, we will discuss the hardware requirements for permitting tri-link simultaneous transmissions to achieve our targeted capacity.

A. Hardware Requirements

Tri-link simultaneous transmission requires the use of the 802.11g chip set to extract packets from overlapping signals. Consider Links A, B and C in Fig. 2. Suppose link A first begins the transmission. Both links B and C can receive the RTS/CTS from link A and thus they (B & C) decide that they can transmit simultaneously with link A. Link B then starts the transmission while link A is still transmitting. From link C's point of view, signals from links A and B are overlapped and thus link C fails to extract the RTS/CTS from link B. Therefore, link C cannot decide if it can transmit simultaneously with link B. To solve this problem, a possible solution is to use the widely deployed IEEE 802.11g DSSS-OFDM chip set. Nodes use direct sequence spread spectrum (DSSS) for transmitting RTS/CTS and Orthogonal Frequency Division Multiplexing (OFDM) for sending DATA and ACK packets. Figure 3 shows parts of the DSSS-OFDM chip set [11]. Consider Fig. 2 again. When link A is sending DATA OFDM signal while link B is transmitting CTS DSSS signal, the filter (as shown in Fig. 3) of node T_C can extract the CTS signal from the DATA signal. Thus link C can decide if it can transmit with links A and B at the same time.

In Fig. 2, let d_{RA-TA} be the distance between R_A and T_A, d_{TB-TA} be the distance between T_B and T_A, and d_{RC-TA} be the distance between R_C and T_A. Again, in a two-ray propagation model, the signal-to-noise ratio at T_A when all links A, B and C are transmitting,

$$SIR = \frac{\dfrac{1}{{d_{RA-TA}}^4}}{\dfrac{1}{{d_{TB-TA}}^4}+\dfrac{1}{{d_{RC-TA}}^4}} = \frac{{d_{TB-TA}}^4 \cdot {d_{RC-TA}}^4}{{d_{RA-TA}}^4 \cdot ({d_{TB-TA}}^4 + {d_{RC-TA}}^4)} \qquad (2)$$

In the worst case that T_A and R_A are separated by the maximum transmission range (250m), and T_B and R_C are just located outside the *VCSRange* of T_A (i.e., d_{TB-TA} and d_{RC-TA} just slightly larger than 550m), *SIR=11.7dB > CPThreshold (10dB)*. Therefore, T_A can receive the signal from R_A successfully even though T_B and R_C are transmitting at the same time.

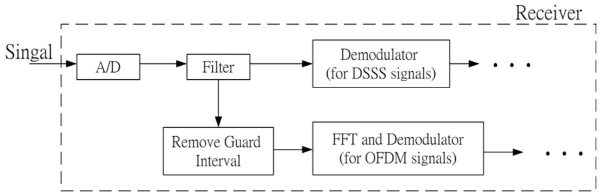


Figure 3. Parts of the IEEE 802.11g DSSS-OFDM chip set

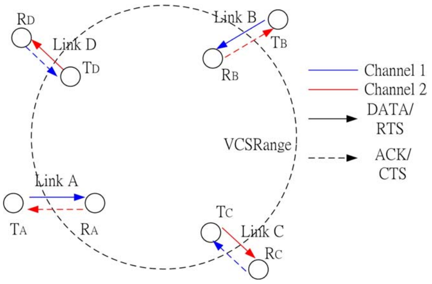


Figure 4. Multiplying the network capacity by our proposed channel assignment scheme

In addition to the use of 802.11g chip set, our MAC protocol requires two transceivers in each wireless node. Since wireless LAN products have already been widely deployed, we believe the cost for adding an additional transceiver would be acceptable, considering the growing costs of radio spectrum resources.

B. Multi-link Simultaneous Transmissions

Figure 4 shows another scenario with the channel assignments based on the directionalities of the links. Links A to D can transmit simultaneously and this can *multiply* the network capacity by using *only two channels*. In this paper, we first consider tri-link simultaneous transmissions with the attempt to double the network throughput by using two channels. Channel assignments of simultaneous transmissions for multiple links require a more complicated MAC protocol. The details will be deferred to another paper.

VI. HIDDEN NODE PROBLEM

Hidden-node problem in wireless networks has been studied extensively. Figure 5a shows a hidden-node scenario with the original 802.11 protocol. When node 3 sends a packet to node 4, node 2 senses the channel to be busy while node 1 senses the channel to be idle, since node 3 is inside the carrier-sensing range of node 2 but outside that of node 1. Once node 1 senses the channel as idle, it may count down its back-off contention window until zero and transmit a packet to node 2.

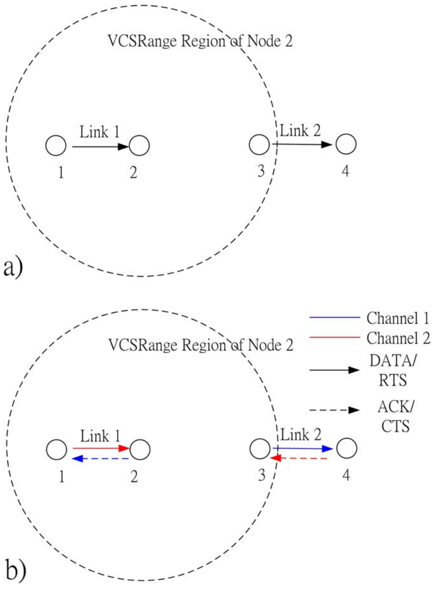


Figure 5. a) Hidden-node problem in the orginial 802.11 protocol does not exist in b) our proposed MAC protocol

If the transmission from node 4 is still in progress, node 2 will continue to sense the channel as busy, and it will not receive the packet from node 1. As a result, node 2 will not return an ACK to node 1. Node 1 may then time out and double the contention window size for retransmission later.

Meanwhile, node 3 transmits the packet successfully and is not aware of the collision at node 2. When transmitting the next packet, node 3 will use the minimum contention window size. The hidden-node scenario favors node 3, and the chance of collision at node 2 can not be reduced even though node 1 backs off before the next retry.

The RTS/CTS mechanism in 802.11 is designed to solve the hidden node problem. However, using RTS/CTS in ad-hoc networks does not eliminate the hidden node problem [12]. The effectiveness of RTS/CTS mechanism is based on the assumption that transmissions by mutually hidden nodes are to a common receiver. Before the transmission of a hidden node begins, the receiver will forewarn other hidden nodes to prevent them from transmitting. This assumption may not hold in an ad-hoc network.

In our proposed MAC protocol, the above hidden-node scenario does not exist thanks to the channel assignment property of the protocol. When using our proposed protocol, if node 3 uses channel 2 to send RTS/DATA to node 4, node 1 will use channel 1 to transmit RTS/DATA to node 2 (as shown in Fig.5b). Since the transmissions are in independent channels, both node 2 and node 4 can receive the signals successfully.

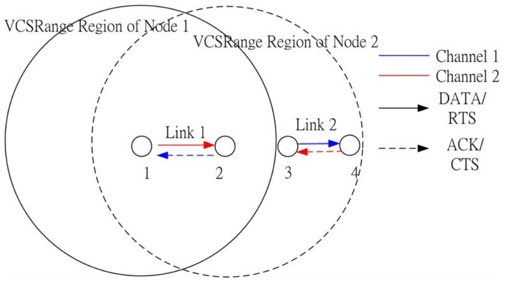


Figure 6. Hidden-node problem in our proposed protocol

Beside the above scenario, there is a "hidden-node" scenario specific to our proposed protocol. Consider Fig. 6. Since nodes 3 and 4 are within the *VCSRange* of node 2, node 2 can receive both the RTS and CTS of link 2. Thus, links 1 and 2 must take turn to transmit. However, node 1 cannot receive the RTS and CTS from link 2 as nodes 3 and 4 are outside the *VCSRange* of node 1. In this way, node 1 may send a RTS to node 2. If the transmission of link 2 is still in progress, node 2 will not reply a CTS to node 1. Thus, node 1 may time out and double its contention window for retransmission later. This will induce an unfairness problem between links 1 and 2. From the protocol's point of view, only one link is allowed to transmit at each time since node 2 can receive both the RTS and CTS from link 2. Although link 2 dominates most of the channel bandwidth, the overall capacity of link 1 and 2 remains the same as allowed by the protocol. Many solutions (e.g. [12][13]) have been proposed in the literature with the attempt to solve the hidden-node problem. A possible solution for the unfairness problem is to extend the *VCSRange* to ensure all potential interfering nodes are covered by the RTS/CTS mechanism. In this example, if we set the *VCSRange* to 700m and the transmission range (*TxRange*) to 250m, node 4 will be at least 450m away from node 2. The signal from node 1 at node 2 will then be at least 10dB stronger than the signal from node 4. Thus, node 2 can capture the packets from node 1 successfully. Due to the space limitation, we refer interested readers to reference [14] for details.

VII. SIMULATION RESULTS

We have implemented our proposed MAC protocol in the NS-2 [15] simulator. For fair comparisons with the settings of the original 802.11 protocol in NS-2, we assume the same values for the *VCSRange*(=550m) and the *TxRange*(=250m). The data rate (for sending DATA/ACK) sets at 12Mbps (OFDM, QPSK) and the basic rate (for transmitting RTS/CTS) sets at 2Mbps (DSSS, DQPSK). In our simulations, all data sources are saturated UDP traffic stream with fixed packet size of 1460bytes.

Figure 7 shows eight links in an 8x2 lattice topology. As shown in Fig. 9a), using the original single-channel 802.11 protocol results in 17.65Mbps total network throughput, thus the targeted capacity for using dual channels is by definition 17.65Mbps*2=35.3Mbps. With our proposed scheme, a total network throughput of 37.64Mbps is achieved, which is 107% of the targeted capacity. In other words, our protocol improves the capacity by 213%. This shows our proposed protocol can *double* the network capacity by using only *two* channels.

Figure 7b shows the channel assignments of links using our proposed protocol. Assume node 11 uses channel 2 to transmit RTS/DATA to node 12. Since nodes 3 and 10 are within the *VCSRange* of node 11, they can receive the RTS but not the CTS of link 6. Nodes 3 and 11 then assign their transmission channels to channel 2. As they are outside the *VCSRange* of node 12, nodes 3, 10 and 11 can transmit simultaneously without interfering with the receptions of signals at node 12. However, when using the original 802.11 protocol, once link 6 is transmitting, links 2 and 5 can not transmit because they can receive the RTS from node 11 of link 6. Our proposed protocol significantly boosts the network capacity by allowing tri-link simultaneous transmissions.

Figure 8 shows another example of an irregular topology with channel assignments. Our proposed scheme obtains total network throughput at 31.49Mbps, which is 106% of the targeted capacity (14.81Mbps*2=29.62Mbps). In other words, our protocol improves the capacity by 213%. Again, the tri-link simultaneous transmission property of our proposed protocol doubles the network throughput. In addition to the capacity enhancement, our proposed protocol achieves a fairer bandwidth allocation in both the lattice and irregular topologies (as shown in Fig. 9).

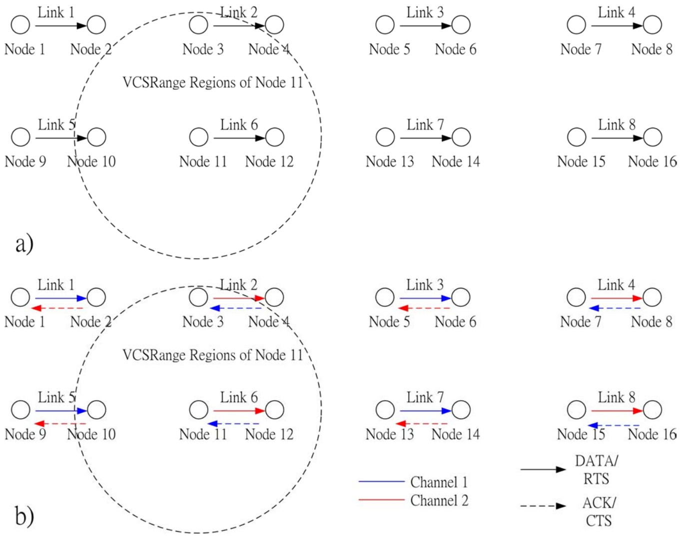


Figure 7. Eight links in a lattice topology using a) original 802.11 and b) our proposed scheme

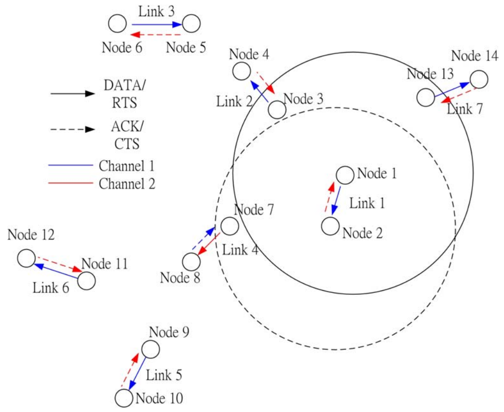
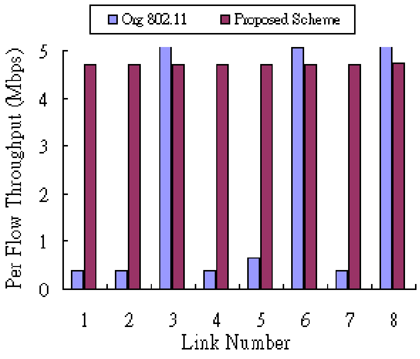
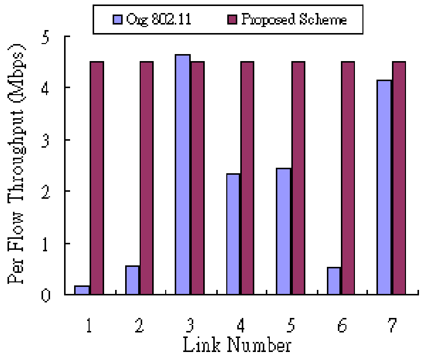


Figure 8. Seven links in an irregular network topology using our proposed scheme



a)



b)

Figure 9. Per-link throughput of the networks of a) Fig. 7 and b) Fig. 8 with the original 802.11 protocol and our proposed MAC protocol

VIII. CONCLUSION

This paper has presented an approach with the aim of doubling the network capacities of 802.11 ad-hoc networks by using a link-directionality-based MAC protocol. The proposed protocol allows tri-link simultaneous transmissions to compensate throughput degradations caused by protocol overheads and various possible network topologies. We have shown that our proposed scheme can boost the network capacities of single channel IEEE 802.11 ad-hoc networks by more than 213%, which we believe it outperforms other multi-channel protocols in terms of capacity per channel resource. We have also demonstrated that the potential of multiplying the network capacities with two channels.

REFERENCES

[1] N. Choi et al., “Multi-channel MAC protocol for mobile ad hoc networks”, *IEEE VTC 2003-Fall*, 2003.

[2] A.Baiocchi et al., “Why a Multichannel Protocol can boost IEEE 802.11 Performance”, *ACM MSWiM’04*.

[3] J. Mo, H. W. So and J. Walrand, “Comparison of Multi-Channel MAC Protocols”, *ACM MSWiM’05*, Oct. 2005.

[4] S. Wu et al, “A New Multi-channel MAC Protocol with On Demand Channel Assignment for Multi-hop Mobile Ad-hoc Networks”, *I-SPAN’00*, Dec. 2000.

[5] Y. Tseng, “A Multi-channel MAC Protocol with Power Control for Multi-hop Mobile Ad-hoc Networks”, *IEEE Distributed Computing Systems Workshop*, April 2001.

[6] J. Chen et al, “A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs”, *IEEE PIMRC’03*, Sept. 2003.

[7] J. So, N. Vaidya, “ Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver”, *ACM MobiHoc’04*, May 2004.

[8] A. Tzamaloukas, J. J. Garcia-Luna-Aceves, “A Receiver-initiated Collision-Avoidance Protocol for Multi-channel Networks”, *IEEE InfoCom’01*, April 2001.

[9] T. Rappaport, “Wireless Communications: Principles and Practice”, Prentice Hall, New Jersey, 2002

[10] P.C. Ng, D. Edwards, S. C. Liew, “A Link-directionality-based Dual Channel MAC Protocol for IEEE 802.11 Ad-hoc Networks”, *IEEE INFOCOM’06 Student Workshop*, April 2006.

[11] “802.11g”, http://www.vocal.com

[12] K. Xu, M. Gerla, S. Bae, “How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?”, *IEEE GLOBECOM '02*, Vol. 1 , pp. 17-21, Nov. 2002

[13] C. L. Fullmer et al, “Complete single-channel solutions to hidden terminal problems in wireless LANs”, *IEEE ICC’97*, June 1997.

[14] L. Jiang and S. C. Liew, “Removing Hidden Nodes in IEEE 802.11 Wireless Networks,” *IEEE Vehicular Technology Conference*, Sept 2005

[15] “The Network Simulator – ns2”, http://www.isi.edu/nsnam/ns.