

How to Lower Congestion with Cross-Layer MPR-PHY/MAC Design?

Sanaz Barghi Hamid Jafarkhani Hodayoun Yousefi'zadeh

Center for Pervasive Communications and Computing

University of California, Irvine

[sbarghi, hamidj, hyousefi]@uci.edu

Abstract—In this paper we introduce a cross-layer design between PHY and MAC for multiple packet reception (MPR). Based on the simulation results, three phases of operation has been identified for MPR-PHY/MAC design depending on the offered load to the network. The design is suitable for multi-hop ad-hoc networks. The heterogeneous nature of the colliding packets in multi-hop ad-hoc networks, both in size of the packets and the arrival times of the packets, requires a sophisticated design to enable MPR. MPR can resolve congestions in a multi-hop network and improve the Goodput of the network. Simulation results show that the saturation Goodput of a network using our proposed MPR-PHY/MAC design is better than that of a network using standard IEEE 802.11 PHY and MAC by a factor of 50% or more.

I. INTRODUCTION

Multiple Packet Reception (MPR) is a promising method for improving the throughput of the network. The throughput of an MPR network scales with the number of nodes in the network [1], while previous results for single packet detectors and interference networks [2] show that by increasing the number of nodes in wireless ad-hoc networks, throughput will not scale. Besides, MPR can help in congested areas of the network, where routes share a node/link or pass close to each other such that their transmissions collide. In networks that utilize random access MAC algorithms, collisions can dominate the transmissions in these areas and cause congestion. In transport layer, congestion control protocols such as TCP, resolve these problems by lowering the number of packets that the sources feed into the network, and therefore lower the total Goodput of the network. With MPR activated in these areas, the congestion problem will be less severe and the overall Goodput of the network will improve.

To move towards MPR for ad-hoc networks, practical constraints must be taken into account. For example, it is not possible for nodes to detect all the packets on the air simultaneously. The number of packets that can be detected simultaneously is usually limited by some characteristics of the system. In CDMA-based MPR, the number of orthogonal codes defines the maximum number of simultaneously detectable packets. In MIMO-MPR methods, the number of antennas at the receiver is the limiting factor. In addition, assuming synchronization between different phases of packet

transmissions in multi-hop ad-hoc networks, as in [3], is impractical.

In this work, we design a cross-layer MPR-PHY/MAC algorithm. The PHY detects two simultaneous packets and takes into account the asynchrony between the reception of the two overlapping packets. We provide solutions to detect different length packets received asynchronously, with a random delay between the reception of two packets, using multiple-antenna terminals. We also design a compatible MAC protocol. Our MAC algorithm can accept and process multiple packets and acknowledge the reception of more than one packet taking into account the asynchronous nature of the multi-hop ad-hoc networks. MPR-PHY/MAC design is compared against IEEE 802.11 over a regular PHY. Considering the behavior of the MPR-PHY/MAC, simulation results show three different regions. In the low-traffic region, there is a minor gain in using MPR-PHY/MAC, because it has slightly lower average delay compared to that of the IEEE 802.11 network, while the Goodput is the same for both networks. In the medium-traffic region, MPR-PHY/MAC has better Goodput and smaller delay compared to IEEE 802.11. The benefit increases as the load of the network increases. In the high traffic region, simulation results illustrate that our MPR-PHY/MAC design can constantly improve the Goodput of a standard IEEE 802.11 network at least by a factor of 50%, while increasing the delay.

The organization of the paper is as follows: In Section II, our MPR-PHY is introduced. Section III discusses our MPR-MAC algorithm and in Section IV simulation results are presented. Section V concludes the paper.

II. PHYSICAL LAYER

We consider an ad-hoc network in which all nodes have two antennas. In what follows, first, we briefly present two different methods to detect multiple packets coming from two-antenna sources. These techniques are, array processing STBC-MPR and zero-forcing for spacial multiplexing. Then, we explain extension of the STBC-MPR method to the case of detecting two asynchronous equal-length packets. Finally, a method to detect two overlapping unequal-length packets will be introduced.

This work was supported in part by a grant from the Boeing Company.

A. Array Processing for STBC-MPR

In this MPR method, User k transmits a pair of symbols, s_1^k, s_2^k using a 2×2 orthogonal Space-Time Block Code (STBC) called Alamouti code [4], [5]:

$$C^k = \begin{pmatrix} s_1^k & s_2^k \\ -s_2^{k*} & s_1^{k*} \end{pmatrix} \quad (1)$$

If two users simultaneously transmit, the received signal at the m th antenna of the receiver can be represented by the following equation:

$$\begin{pmatrix} r_{1,m} \\ r_{2,m} \end{pmatrix} = \begin{pmatrix} s_1^1 & s_2^1 \\ -s_2^{1*} & s_1^{1*} \end{pmatrix} \begin{pmatrix} g_{1,m}^1 \\ g_{2,m}^1 \end{pmatrix} + \begin{pmatrix} s_2^2 & s_1^2 \\ -s_1^{2*} & s_2^{2*} \end{pmatrix} \begin{pmatrix} g_{1,m}^2 \\ g_{2,m}^2 \end{pmatrix} + N$$

$$R_m = \begin{pmatrix} r_{1,m} \\ r_{2,m} \end{pmatrix} = G_{1,m} \begin{pmatrix} s_1^1 \\ s_2^1 \end{pmatrix} + G_{2,m} \begin{pmatrix} s_2^2 \\ s_1^2 \end{pmatrix} + N' \quad (2)$$

where $G_{i,m} = \begin{pmatrix} g_{1,m}^i & g_{2,m}^i \\ g_{2,m}^{i*} & -g_{1,m}^{i*} \end{pmatrix}$. The interference of User 2 on User 1 can be canceled using array processing [6]:

$$R_{free} = \frac{G_{2,1}^H R_1}{\|G_{2,1}\|^2} - \frac{G_{2,2}^H R_2}{\|G_{2,2}\|^2}$$

$$= \left(\frac{G_{2,1} G_{1,1}}{\|G_{2,1}\|^2} - \frac{G_{2,2} G_{1,2}}{\|G_{2,2}\|^2} \right) \begin{pmatrix} s_1^1 \\ s_2^1 \end{pmatrix} + \eta \quad (3)$$

where R_{free} is only a function of User 1's signal and free of User 2's interference. User 1's signal can be easily detected from Eq. (3). With similar steps, the receiver can detect symbols from User 2 as well. This method provides diversity of two for the signal of both users.

B. Zero-forcing for 2-by-2 Spacial Multiplexing

For spacial multiplexing, a pair of symbols, s_1, s_2 are transmitted through the two antennas of the transmitter at each time slot. Unlike space-time coding, there is no coding and there will be no diversity gain at the receiver. Here, we assume only one user with two antennas. Therefore, the received signal at Antenna m of the receiver is:

$$R = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \begin{pmatrix} g_{1,1} & g_{1,2} \\ g_{2,1} & g_{2,2} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + N \quad (4)$$

Using zero-forcing [4] and multiplying both sides of Eq. (4) by the inverse of matrix $G = \begin{pmatrix} g_{1,1} & g_{1,2} \\ g_{2,1} & g_{2,2} \end{pmatrix}$, we will have:

$$G^{-1}R = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + G^{-1}N \quad (5)$$

Decoding s_1 and s_2 from Eq. (5) is straightforward. This method provides diversity one for each symbol. More efficient methods such as V-BLAST [7] can be used to detect the symbols with better bit error rates and higher efficiency.

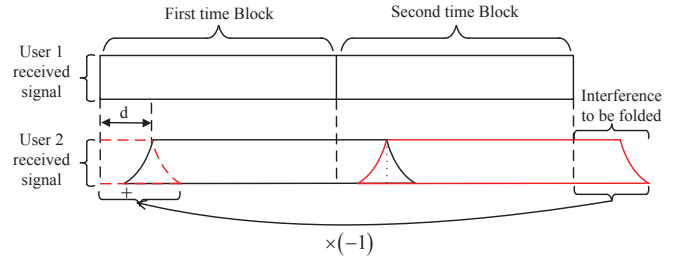


Fig. 1. The diagram of noiseless received signals at the receiver and the interference folding process.

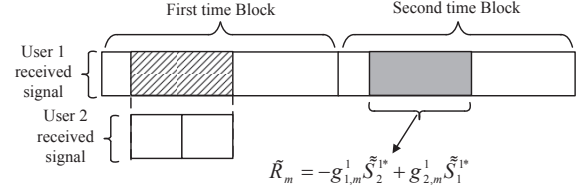


Fig. 2. Detection of unequal-length packets

C. Detecting Equal-Length Packets

Packets from different sources arrive at the receiver at different times unless there is a synchronization algorithm that synchronizes the transmissions from different users. In ad-hoc networks, making such a synchronization assumption is impractical. In addition, Eq. (3) is based on the assumption of perfect symbol-by-symbol synchronization between the two arriving signals at the receiver. For asynchronous signals, Eq. (3) will not cancel the interference completely, and bit error rate will be higher than the ideal synchronous transmissions. To solve this problem, in [8] we introduced a modified transmission and detection technique. Each transmitter splits a packet into two equal blocks, S_1, S_2 , called super-symbols, and performs Alamouti coding on the pair of super-symbols to transmit them over two antennas in two time blocks. We depict the received signals from the two users, after the folding process [8], in Fig. 1. User 2's signal in Fig. 1 is distorted because of the imperfect sampling and asynchronous reception. In summary, the folding process and super-symbol coding, together, make it possible to use Eq. (3) to cancel the interference from an asynchronous transmission (for details please refer to [8]). This method provides diversity two and has low bit error rate.

D. Detecting Unequal-Length Packets

If the length of one of the overlapping packets is more than twice that of the other packet, as depicted in Fig. 2, the receiver can still detect both packets. In this case, the folding method will not work and the interference cannot be canceled. To overcome this limitation, we use the fact that, with Alamouti coding each super-symbol is transmitted twice, once in each time block, as shown in Eq. (1). In Fig. 2, the striped area in User 1's packet, in the first time block, overlaps User 2's packet. If we call the overlapping part of User 1's super-symbols transmitted in striped time \tilde{S}_1^1 and \tilde{S}_2^1 , the signal

received in gray area in Fig. 2 is given by:

$$R_{gray} = \begin{pmatrix} \tilde{R}_1 \\ \tilde{R}_2 \end{pmatrix} = \begin{pmatrix} g_{1,1}^1 & g_{1,2}^1 \\ g_{2,1}^1 & g_{2,2}^1 \end{pmatrix} \begin{pmatrix} -\tilde{S}_2^{1*} \\ \tilde{S}_1^{1*} \end{pmatrix} + \tilde{N} \quad (6)$$

R_{gray} is the interference free signal from User 1. By comparing Eq. (6) with Eq. (4), it is easy to see that by replacing s_1, s_2 with $-\tilde{S}_2^{1*}, \tilde{S}_1^{1*}$ in Eq. (4), we will get Eq. (6). Using symmetrical constellations, such as PSK and QAM, if s is a constellation point, s^* and $-s^*$ are also points from the same constellation. There is a simple mapping between the corresponding bits of s and those of s^* or $-s^*$ and demodulation is fairly simple. Therefore, the receiver can use zero-forcing, presented in Section II-B, to recover User 1's signal in the gray area. The rest of User 1's data, i.e., the white area in Fig. 2, is interference free in both time blocks and a single packet detection method can recover it. Single packet detection of Alamouti coded data with two receive antennas has diversity four, but, as mentioned before, the zero-forcing method has diversity one. Therefore, the bit error rate (BER) of the gray area will be higher than that of the white area, and it will dominate the BER for detection of the packet. We also note that if the shorter packet arrives at the second time block of User 1's packet or interferes with User 1's packet in both time blocks, the same method is still applicable with some small modifications. Since the details are straightforward and in the sake of space, we skip the details.

None of the methods discussed above can help to remove the interference on User 2's packet in Fig. 2. However, interfering signal of User 1 in the striped area can be fully reconstructed if the receiver has the channel coefficients and User 1's data transmitted over the striped area. \tilde{S}_1^1 and \tilde{S}_2^1 were previously detected by applying zero-forcing method on the received signal in the gray area. If the long packet is received correctly, we can proceed to detect User 2's packet. Receiver can use cyclic-redundancy check (CRC) codes to make sure that User 1's data is detected correctly.

Knowing channel coefficients, \tilde{S}_1^1 , and \tilde{S}_2^1 , our reconstruction of the received signal in the striped area, \hat{R} , is given in the following equation:

$$\hat{R} = \begin{pmatrix} g_{1,1}^1 & g_{1,2}^1 \\ g_{2,1}^1 & g_{2,2}^1 \end{pmatrix} \begin{pmatrix} \tilde{S}_1^1 \\ \tilde{S}_2^1 \end{pmatrix} \quad (7)$$

The goal is to reconstruct the interference from User 1's packet on User 2's packet and subtract it from the received signal to have an interference free signal that is only a function of the signal transmitted by User 2. To reach this goal, the asynchrony between the two users must be taken into account. Therefore, the receiver samples the received signal twice, once in sync with User 1's signal, R_{s1} , and once in sync with User 2's, R_{s2} . Then, the receiver uses the former signal for the detection of User 1's packet and the latter signal for the detection of User 2's packet. The interference from User 1 on

User 2's packet in R_{s2} is given by:

$$R_{1 \rightarrow 2} = \hat{\mathbf{R}} \mathbf{D} \quad (8)$$

$$\mathbf{D} = \begin{pmatrix} p_0 & p_1 & \dots & p_{L+2l} \\ p_{-1} & p_0 & \dots & p_{L+2l-1} \\ \vdots & & \ddots & \\ p_{-(L+2l)} & p_{-(L+2l-1)} & \dots & p_0 \end{pmatrix}$$

Where l is the length of the pulse shaping filter's impulse response, \mathbf{D} is the dispersion matrix and $p_i = P(\delta_1 + iT_s)$ is the i th sample of the pulse shaping filter $P(\cdot)$ with sampling error for User 1's signal in R_{s2} , i.e., δ_1 . After subtracting Eq. (8) from R_{s2} , the receiver passes the result through an Alamouti decoder to detect User 2's data.

III. MPR-MAC PROTOCOL

The MPR-MAC is a random access MAC designed based on that of IEEE 802.11 [9] algorithm. The back off procedure used by MPR-MAC is the same as the binary exponential back off (BEB) of IEEE 802.11. MPR-MAC uses the Data-ACK cycle to transmit a data packet to a receiver and then send an acknowledgment to the transmitter after successful reception of that packet. There are three specific features for our proposed MPR-MAC:

- 1) Asynchronous ACKs: As explained in Section II, a receiver cannot detect two overlapping packets and acknowledge them before both packets are received completely. This means that ACK packets cannot be always transmitted within the standard IEEE 802.11 time frame. Therefore, we define asynchronous ACK packets. A transmitter will increase its timeout value for the ACK packets and the intended receiver will send out asynchronous ACK packets as soon as it detects the channel is free, which is not necessarily happening in the timeout window of the transmitter.
- 2) ACK structure: ACK packets have two receiver address fields, as oppose to one in the case of IEEE 802.11. Associated with each receiver address field, there is a 32-bit packet ID field carrying information about the packet being acknowledged. These ID fields are required to notify the transmitter which packet is being ACKed because of the use of an asynchronous ACK mechanism. Unacked data packets are queued and kept until they are expired as reflected by their timeout value. This 32-bit packet ID is chosen to be a 32-bit hash of the packet's source IP address and IP sequence number, such that an explicit ID field is not required for data packets, similar to [10].
- 3) ACK delay estimation: Increasing the timeout duration for ACK packets increases the time overhead for data transmission, and a small timeout value initiates unnecessary retransmissions. Therefore, our MPR-MAC needs to adaptively set the timeout value of the ACK packets, depending on the network condition. For this purpose, MPR-PHY/MAC utilizes the same algorithm

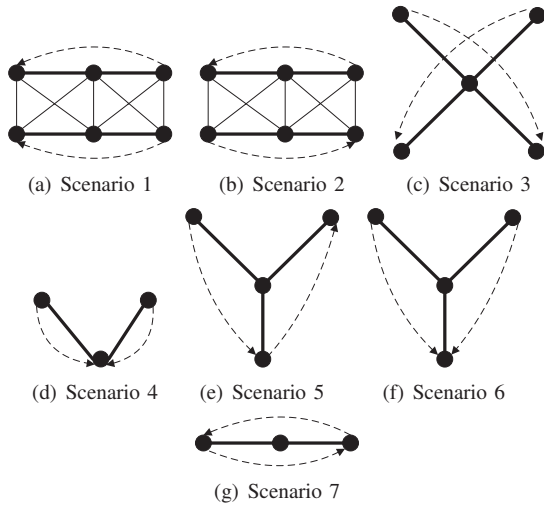


Fig. 3. An illustration of the two-hop simulation scenarios. Only nodes with solid line between them can hear one another's transmissions. Dashed arrows are pointing from the sources of the traffic to the destinations.

as the one used by TCP in order to estimate the round trip time and estimates the ACK packet's arrival delay. The latter allows for adaptively setting the timeout value of each ACK packet.

IV. SIMULATION RESULTS

A. Scenarios under Study

In multi-hop networks, sources employ congestion control algorithms to alleviate congestion in high-traffic areas. Congestion usually happens when sources are loading the network with more than its transmission capacity. With MPR-PHY/MAC methods, the capacity of network increases and as such we expect that the utilization of MPR-PHY/MAC in a multi-hop ad-hoc network relieve congestion in high-traffic areas. MPR can help relieving the congestions rooted in high number of collisions happening in some part of the route to the destination. The collisions are usually caused by routes passing physically close to each other such that the transmissions collide. The scenarios of our study consider two-hop traffic flows passing in the vicinity of each other, destined to the same node, or passing over the same link. There are seven scenarios studied in this paper that are illustrated in Fig. 3. In this figure, for each traffic a dashed arrow connects source and destination of the traffic.

B. Simulation Settings

We utilize discrete event network simulator (NS-2) to perform our simulations. We modify the IEEE 802.11 algorithm in order to accept asynchronous ACKs and also add a timeout approximation algorithm to its MAC algorithm. Assuming each node is equipped with two antennas, each ACK message has two receiver address fields and two packet ID fields. We run a TCP Reno agent on top of the MAC and use a Poisson traffic generating application. Our Poisson traffic generator generates packets of length $L = 1024$ bytes in average every t_s ms. Therefore, the average offered load to TCP will be

TABLE I
RESULTS OF MATLAB SIMULATIONS: PER FOR MPR.

Packet Type	Data	MAC layer ACK	TCP ACK packet
Data	0.1729	0.1650	0.3704
MAC layer ACK	0.1650	0.0069	0.1650
TCP ACK packet	0.3704	0.1650	0.0201

$\frac{8 \times 10^3}{t_s}$ kbps. Hence, parameter t_s is a representation of the input load to the network.

PHY MPR methods are being simulated in MATLAB environment and the results of average packet error rates (PERs) for $SNR = 25$ dB are given in Table I. In detecting unequal length packets, the algorithm drops the shorter packet without trying to decode it if the detection of a longer packet fails. That is why the probabilities of error for detecting unequal length overlapping packets are almost the same. Since the IEEE 802.11 MAC in NS-2 is designed to be aware of the arrival times of the packets and their types, we implement the MPR-PHY in our MAC code. Further, we approximate the behavior of the PHY in our network simulations since a real time simulation of the PHY is problematic due to the time scale difference between network events and bit-level events. Our approach calls for generating a Bernoulli random variable and using the average PER in Table I to drop packets in the MAC layer based on packet types overlapping each other.

C. Results and Discussion

As measures of performance, we look at the application layer Goodput and average delay in the MPR-PHY/MAC network scenarios of Fig. 3 and compare them to those of IEEE 802.11 MAC over a regular PHY. Our results are illustrated in Fig. 4. In all scenarios, there are three regions in the relative performance of the two networks. The regions width vary slightly depending on the scenario under study and given values for t_s reflects the average behavior.

- 1) Low-Traffic Load ($75ms \leq t_s \leq 200ms$): In this region, both IEEE 802.11 MAC and MPR-PHY/MAC almost have the same throughput. The improvement in this part is in the average delay of packet delivery which is slightly lower for MPR-PHY/MAC. In the low-traffic region, the network offered load is lower than the actual capacity of the network. Therefore, the single packet detector PHY and MAC of IEEE 802.11 are able to resolve collisions and deliver all packets to intended receivers. Consequently, the MPR-PHY/MAC cannot increase the number of delivered packets. However, MPR-PHY/MAC can reduce the average delay of packet delivery since it can resolve collisions in a more efficient and faster way compared to IEEE 802.11.
- 2) Medium-Traffic Load ($20ms < t_s < 75ms$): In this phase, the MPR-PHY/MAC delivers a higher number of packets in a unit time compared to IEEE 802.11. The average delay is considerably lower for the MPR-PHY/MAC. The Goodput of the MPR-PHY/MAC is higher than that of IEEE 802.11 and the improvement varies for different scenarios. In this region the average delay can be reduced to lower than $\frac{1}{6}$ of the original

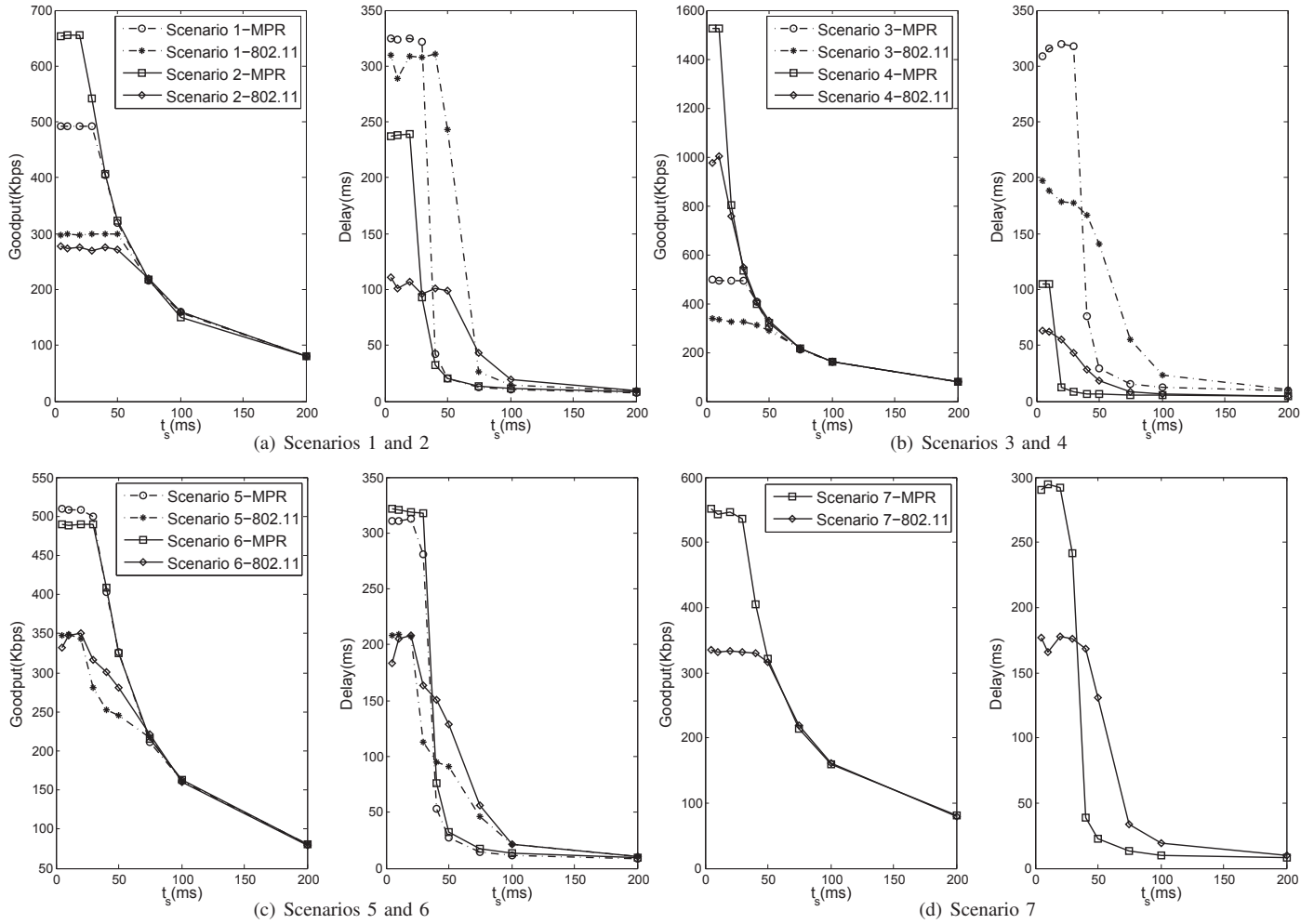


Fig. 4. Simulation results associated with different scenarios of Fig. 3.

delay in the IEEE 802.11 network. The improvement in the Goodput of the network increases by increasing the load in the network.

- 3) High-Traffic Load ($t_s \leq 20ms$): This region is the saturation region in which the offered load is high enough allowing the sources to always have packets to transmit. In this phase, the MPR-PHY/MAC improves the Goodput of the network at least by 50% in comparison to the Goodput of IEEE 802.11. In saturation traffic, the improvement in the Goodput of the network is independent of the offered load to the network. However, the average delay is also up to 50% higher than that of IEEE 802.11. This is expected as the average delay is a function of the queuing delay among other delay factors. As the number of queued packets significantly increases in this case, it is expected that the average packet delivery delay outweighs the delay improvements of MPR.

V. CONCLUSION

We introduced a pair of MPR-PHY and MPR-MAC algorithms attempting at improving the performance of IEEE 802.11 networks. The MPR-PHY method benefited from the

existence of multiple antenna nodes. While the MPR-PHY detected two equal-length packets with a diversity of two and a low BER, it detected part of the unequal-length packets with a diversity of one. The associated MPR-MAC handled the reception of two packets at the PHY and acknowledged both of them if required. An asynchronous acknowledgment procedure was also introduced to improve network throughput. We showed that the TCP protocol running on top of the MPR-PHY/MAC experiences a lower packet delivery delay in the low-traffic region and a higher throughput in high traffic (saturation) region. Our simulation results demonstrate that MPR-PHY/MAC provides different types of gains depending on the offered load to the network. For two two-hop traffic flows routed close to each other, MPR-PHY/MAC can improve the Goodput of the network in the high-traffic region by at least 50%. It lowers the average delay of the network by a factor of $\frac{1}{6}$ in the medium-traffic region.

REFERENCES

- [1] J. J. Garcia-Luna-Aceves, H. R. Sadjadpour, and Z. Wang, "Challenges: towards truly scalable ad hoc networks," in *MobiCom '07*. New York, NY, USA: ACM, 2007, pp. 207–214.
- [2] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

- [3] P. Casari, M. Levorato, and M. Zorzi, "MAC/PHY crosslayer design of MIMO ad hoc networks with layered multiuser detection," *IEEE Trans. on Wireless Commun.*, vol. 7, pp. 4596–4607, Nov. 2008.
- [4] H. Jafarkhani, *Space-Time Coding: Theory and Practice*. Cambridge University Press, 2005.
- [5] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. on Select. Areas in Commun.*, vol. 16, pp. 1451–1458, Oct. 1998.
- [6] A. Naguib, N. Seshadri, and A. Calderbank, "Applications of space-time block codes and interference suppression for high capacity and high data rate wireless systems," *Proc. 32nd Asilomar Conf. Signals, Systems and Computers*, vol. 2, pp. 1803–1810, Nov. 1998.
- [7] P. Wolniansky, G. Foschini, G. Golden, and R. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," *Proc. ISSSE-98*, pp. 295–300, Sep. 1998.
- [8] S. Barghi, H. Jafarkhani, and H. Yousefi'zadeh, "MIMO-Assisted MPR-Aware MAC Design for Asynchronous WLANs," *To be published in IEEE/ACM Trans. on Networking*, available at: <http://www.ece.uci.edu/~hyousefi/publ/mprToN.pdf>.
- [9] *IEEE 802.11 Standard - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Std., Jan. 2007.
- [10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 497–510, 2008.