LN-MAC: a Cross-layer Explicit Loss Notification Solution for TCP over IEEE 802.11

Ayyappan Ravichandran[1], Marco Tacca[1], Michael Welzl[2], and Andrea Fumagalli[1]
[1]OpNear Lab, Erik Jonsson School of Engineering and Computer Science,
The University of Texas at Dallas, 800 West Campbell Road, Richardson, TX 75083, USA
Email: [axr065100|mtacca|andreaf]@utdallas.edu
[2]Institute of Computer Science, University of Innsbruck, A-6020 Innsbruck, Austria
Email: [michael.welzl]@uibk.ac.at

***Abstract*— WiFi, i.e., IEEE 802.11 is one of the most widely used technologies to implement internet access in today's networks. Typical frame error rates in IEEE 802.11 are much higher than in wired links. This negatively impacts the behavior of TCP, which assumes that packet loss is due to congestion. This paper presents LN-MAC, a cross-layer solution based on explicit loss notification (ELN), where the IEEE 802.11 MAC is enhanced to provide loss notifications to the TCP layer. Extensive simulation results demonstrate the benefits of LN-MAC. Numerical results demonstrate that under any condition TCP throughput is always higher when using LN-MAC than when using conventional IEEE 802.11.**

I. Introduction

With wireless data transfers becoming more and more popular, the expectations of these networks increase everyday. The IEEE 802.11 [1] family of standards, referred to as WiFi, is the most commonly used in establishing wireless access. However, wireless channels are subject to heavier packet loss due to corruption and other synchronization issues when compared to wired channels. Additionally, most household wireless networks are subject to interferences from home appliances, such as microwaves, mobile phones, and other electronic devices which produce interfering signals. As a result, typical wireless link error rates are in the range 10^{-3} to 10^{-1}, while typical wired link error rates are in the range of 10^{-6} to 10^{-8} [2]. This hugely affects the transport layer (TCP) performance [3] which was originally designed to operate over networks with wired links. Unless otherwise specified, TCP interprets the occurrence of a packet loss as an indicator of network congestion, which is resolved by promptly decreasing the congestion window, i.e., the TCP sender's transmission rate. One possible approach to contain the TCP performance loss due to packet corruption losses is based on performance enhancing proxies (PEP) [4], Berkeley SNOOP protocol [5] or TULIP [6]. Another solution, i.e., Split-TCP [7] and indirect TCP [8] tries to enhance performance by splitting the TCP connection between the wireless links and the wired links. This paper presents a study of the performance of LN-MAC, a cross-layer explicit loss notification (ELN) [9] based method in TCP/IP over IEEE 802.11. The study is based on the use of two TCP options introduced by one of the authors, i.e., the Corruption Detection Option (CDO) and Corruption Notification Option (CNO) [10]. In a real scenario, the CNO/CDO option technique alone improves marginally the TCP performance, as demonstrated in [11]. The CNO/CDO technique is augmented with a cross-layer solution, where the IEEE 802.11 MAC and TCP layers cooperate to achieve better performance in the presence of errors due to the wireless channel. Performance gain is obtained by taking advantage of the knowledge of the link status at the MAC layer, i.e., the MAC obtains information about the sequence number of the TCP segments. Upon a MAC frame loss, the MAC can directly inform the TCP layer. This effectively decreases the round trip time in the presence of errors and the TCP reaction time to losses. Simulation results demonstrate that the cross-layer LN-MAC protocol allows to always gain in terms of throughput under any wireless link conditions. Furthermore, the protocol design is backward compatible with existing implementations of TCP.

II. System Description

This section briefly describes how CNO/CDO options are used in conjunction with TCP and how the LN-MAC cross-layer solution takes advantage of the ELN technique. Notice that IEEE 802.11 uses a CRC that covers both MAC headers and MAC payload. IEEE 802.11 cards discard frames where the CRC detects an error. The CNO/CDO solution requires a modification to the behavior of IEEE 802.11: when a frame has error(s), it should not be discarded, it should be passed to the higher layers. Higher layers will handle the packets as described.

A. TCP Using CNO/CDO

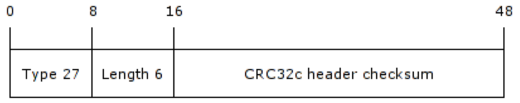

Fig. 1. CDO as specified in [10].

Corruption detection is performed using a TCP header option in the data segment. CDO is a six byte option (Fig. 1).

This research is supported in part by NSF grants No. ECS-0225528 and CNS-0435429

Four of the six bytes contain the CRC32c, which is the same as the one used in the SCTP protocol [12], [13]. CRC32c covers a subset of both the TCP header and the pseudo header, thus providing an additional detection mechanism to the regular 16-bit TCP checksum. The covered fields are the source and destination address of the IP header, the source and destination port, the sequence and acknowledgment number, the CWR, ECE, ACK, RST, SYN and FIN control bits, and the ECN field [14]. CDO can be used to check the integrity of fields the that are needed to deliver a partially corrupt packet to the TCP receiver. In turn, the TCP receiver informs the sender about the reception of a partially corrupt packet.

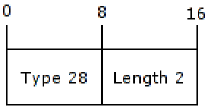

Fig. 2. CNO as specified in [10].

Corruption notification is performed using a TCP header option in the ACK segment. CNO is a 2 byte option (Fig. 2). Upon the reception of a partially corrupted data segment, the TCP receiver can send an ACK segment with CNO. The CNO is used to request the retransmission of the data segment, while indicating that the segment loss was not caused by congestion. Upon the reception of an ACK segment with CNO, the TCP sender

- updates the values of the round trip time (RTT) and the retransmission timeout (RTO),
- retransmits the corrupt data segment,
- updates the congestion window size (CWND).

The first two actions are specified in [10], whereas the third one is discussed next. The specification of CNO/CDO [10] does not define how to update CWND upon reception of an ACK with CNO. One possible implementation of a sender's congestion control behavior is to react similarly to both corruption and congestion. Even with such a conservative choice, several benefits remain, e.g., earlier retransmission, retained control information for connection setup or tear down, correctly updated RTO. In this paper, an aggressive approach is used instead, i.e., an approach similar to the one proposed in [11], [15] where the value of cwnd is not modified upon reception of an ACK segment containing the CNO option.

B. LN-MAC

LN-MAC is a cross layer solution designed to take advantage of the cooperation between the MAC layer and the TCP layer. LN-MAC operates at both the transmitting end and at the receiving end of the wireless link (in this paper the wireless link is IEEE 802.11). The cooperation takes place in identifying losses due to collisions or noise on the wireless link. The MAC operating at the transmitting end of the wireless link performs the following operations. When a frame containing a TCP segment is received, the MAC reads the TCP header information. For each TCP connection, i.e., for each pair source IP address/port and destination IP address/port, the MAC records the value of the sequence number (SN) of the last segment that was transmitted (SN_T)[1]. Notice that the transmitting end MAC is stateful, i.e., it maintains state for each outgoing TCP connection[2]. The MAC then includes such a sequence number in the frame being transmitted, i.e., the frame contains both the current sequence number in the TCP header and the sequence number of the last attempted TCP segment. The inclusion of the sequence number can be done without modification of the existing frame structure at the MAC. The TCP sequence number of the last TCP segment can be included into the MAC Frame without adding any bits as depicted in fig. 3.

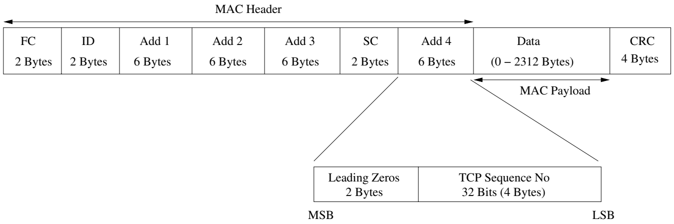

Fig. 3. MAC frame structure with TCP sequence number field for LN-MAC.

The MAC operating at the receiving end of the wireless link performs the following operations. Similar to the transmitting end MAC, the receiving end MAC maintains state for all TCP connections. Additionally, for each TCP connection, the MAC maintains record of the sequence number of the last successfully received TCP segment (SN_R). Upon reception of a frame containing a TCP segment, the MAC first identifies to which connection it belongs. Then it compares sequence numbers SN_T contained in the received frame and SN_R (stored locally). If they match, then no action is necessary. If they do not match, a TCP segment was lost. Then, the MAC generates an empty TCP segment where the source IP address/port number and destination IP address/port number are the same as the packets belonging to the connection. The sequence number in the packet is set to the value SN_T. The packet is empty and includes the CDO option to indicate to the destination the loss of a packet. Upon reception of a segment with the CDO option, the receiving end of the TCP connection operates according to the procedure outlined in section II-A, i.e., it generates an ACK segment with the CNO option.

Fig.4 describes the Frame Control (FC) Field bits of the MAC header format for the IEEE 802.11 standard and the changes required to support LN-MAC. The type option bits B_3B_2 : 10 is reserved for data descriptions and can be used to indicate that the MAC supports LN-MAC options with the SubType Bits $B_7B_6B_5B_4$: 1000.

III. Performance Evaluation

This section discusses the assumptions used to obtain simulation results. The performance evaluation is carried on by comparing the TCP layer throughput for three versions of

[1] This includes both successful segments and unsuccessful segments.

[2] Notice that, although not scalable in general, in the context of IEEE 802.11 used as an access point, the proposed solution does not present scalability problems.

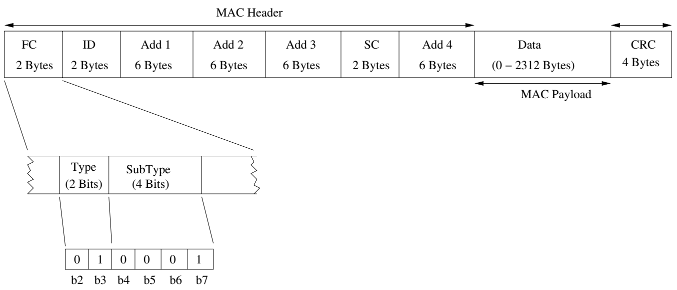

Fig. 4. MAC Frame Structure: Enable/Disable Field for LN-MAC Option

TABLE I
PARAMETERS USED IN THE SIMULATIONS

Path Loss Exponent β = 4	Fading is Flat Rayleigh
Average Transmitter Power = 100 mW	PHY Header = 192 bits
SIFS = 10 μs	RIFS = 30 μs
DIFS = 50 μs	Slot Time = 20 μs
Vulnerable Period = 20 μs	CWmin = 31 slots
CWmax = 255 slots	MAC Header = 34 bytes
MAC ACK = 14 bytes	Sensitivity = -107 dBm

TCP: TCP plain, TCP with CNO/CDO, and TCP with LN-MAC. TCP plain refers to the TCP layer operating in conformance with the NewReno standard [16]. Simulation results are obtained using a custom built simulator. For details of the simulator see [17], [18]. We assume a Rayleigh distributed flat fading channel whose coherence time is constant over the duration of one packet. The sensing threshold of the receiver is set to -107 dBm. When a node senses a power level that is higher than -107 dBm, it will detect a busy channel. The interference threshold is clamped at -137 dBm, i.e., any interference is accounted only if the power level at the receiving node is greater than -137 dBm[3]. The simulation parameters for the link layer are summarized in Table I. Each point in the plots is an average of 6 simulation runs.

A. Simulation Results

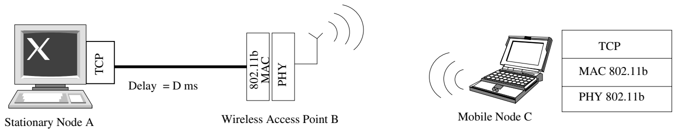

Fig. 5. Network topology. A delay of D = 0 ms or 20 ms is used in the experiments

Fig. 5 shows the network setup used to obtain results. Node A acts as the source node which is behind the wired backbone. The TCP layer at node A receives 1200 packets per second. Node C is the wireless node which is connected to the source A through the wireless link provided through B. The latency D in Fig. 5 emulates the presence of the network. It is assumed that no losses occur due to congestion at node B, nor packet losses due to corruption occurs on the link between nodes A and B. The wireless channel between nodes B and C is based on the IEEE 802.11b standard and operates at 11Mb/s.

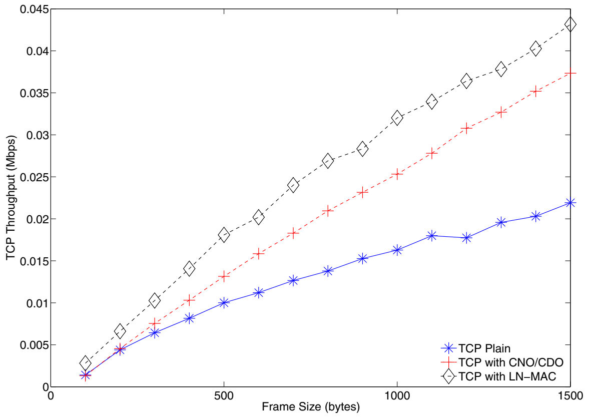

Fig. 6. TCP throughput vs. frame size, link frame error rate = 40%, max MAC retry limit = 1

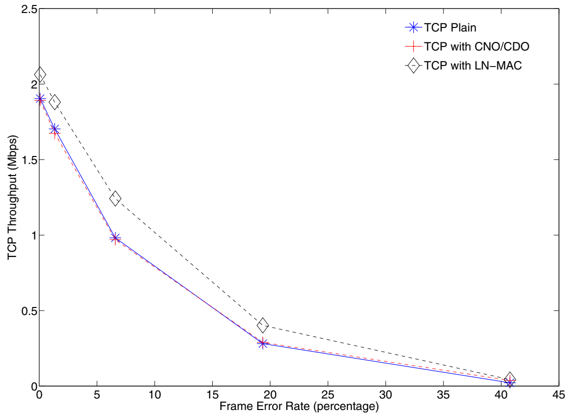

Fig. 7. TCP throughput vs. link frame error rate, frame size = 1500 bytes, max MAC retry limit = 1

Fig. 6 shows TCP throughput versus the IEEE 802.11 frame size when the maximum number of retries at the MAC layer is set to 1. The plots show that throughput improvement for LN-MAC increases as the frame size increases. This can be explained by the fact that when the frame size is small, the frame error rate is small, therefore the performance gain introduced by LN-MAC is less significant. The plots demonstrate that LN-MAC performs better than TCP with CNO/CDO. The reason lies in the fact that TCP with CNO/CDO is effective only if the errors in the frame are confined to the payload, while LN-MAC can overcome this limitation. Fig. 7 shows the TCP throughput versus the frame error rate. The plots show that LN-MAC outperforms the other versions of TCP under any condition.

To assess the impact of the proposed solution, simulations with an increased number of MAC retransmissions are performed. Fig. 8 reports the TCP throughput versus the IEEE

[3] -107 dBm sensing range with a transmitted power of 100 mW correspond to a range of about 150 m at 1Mb/s in a non-fading channel where the path loss exponent β is 4.

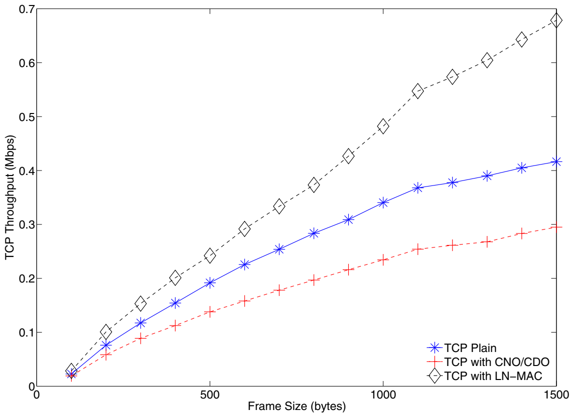

Fig. 8. TCP throughput vs. frame size, link frame error rate = 40%, max MAC retry limit = 2.

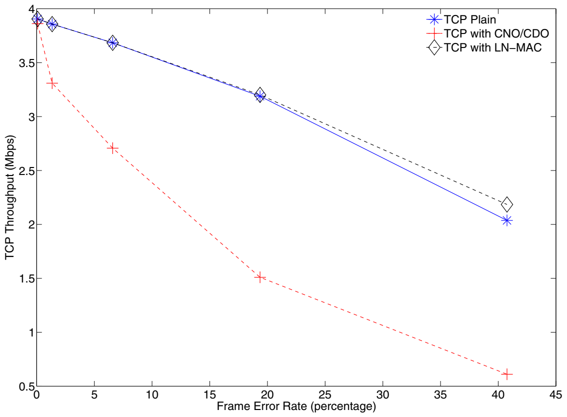

Fig. 9. TCP throughput vs. link frame error rate, frame size = 1500 bytes, max MAC retry limit = 6.

802.11 frame size when IEEE 802.11 can attempt up to 2 times to transmit a frame. Fig. 9 reports the TCP throughput versus the frame error rate when IEEE 802.11 can attempt up to 6 times to transmit a frame. Both plots show that LN-MAC outperforms the other 2 versions of TCP. In this plot the performance gain relative to TCP plain is reduced due to the fact that the packet loss probability is reduced because of the increased number of possible retransmissions. The plots demonstrate an interesting effect for TCP with CNO/CDO. When multiple retransmission attempts are possible at the MAC, the same frame might be received with intact header and corrupt payload more than once. This effect triggers the generation of multiple packets with CDO enabled towards the destination, which will in turn generate multiple ACK packets with CNO enabled, effectively reducing TCP throughput.

Fig. 10 shows the number of times the congestion window is reset at node A during duration of the simulation when the maximum number of retransmission attempts at the MAC layer is set to 1. The curves reveal that the main reason for LN-MAC to lead to improved performance is the lower number of congestion window resets.

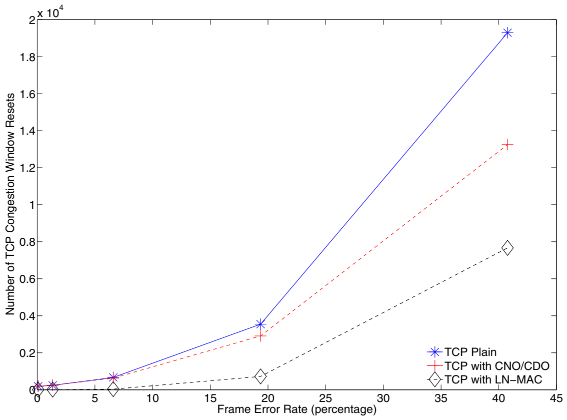

Fig. 10. Number of TCP congestion window resets

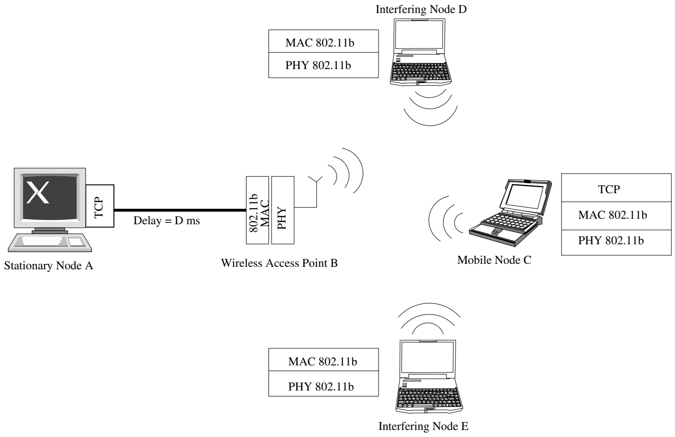

Fig. 11. Network topology with interfering sources.

The network scenario shown in Fig. 11 is considered to assess the impact of multiple interfering sources. The setup is similar to the case in Fig. 5 with the addition of interfering nodes D and E, where node D generates traffic intended for node E. The TCP layer at both node A and D, receives 600 packets per second. Results are obtained by setting D=20ms. Fig. 12 and Fig. 13 show results that are consistent with the ones obtained in the previous section.

IV. Conclusions

This paper introduces an advanced cross-layer loss notification MAC (LN-MAC) aimed at improving the TCP performance over WLAN networks. The solution is based on a cross-layer explicit loss notification technique. This is achieved without interrupting the end-to-end TCP semantics since the protocol does not hide the traffic from TCP. Obtained results demonstrate the performance gain of the proposed solution.

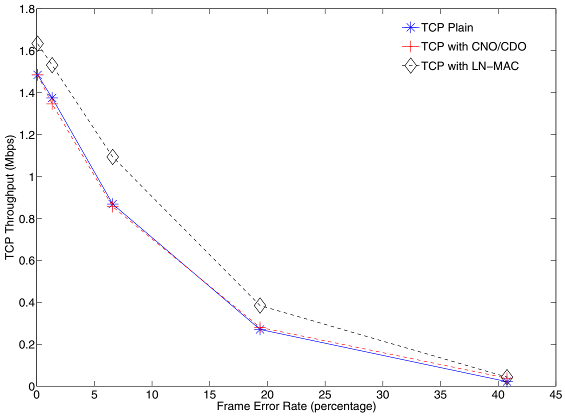

Fig. 12. TCP throughput vs. frame size, number of nodes = 4 , link frame error rate = 40%, max MAC retry limit = 1.

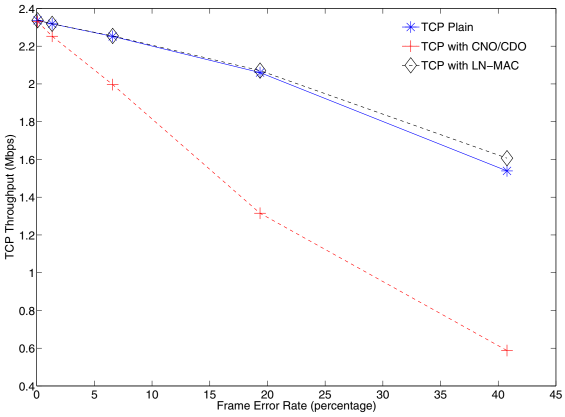

Fig. 13. TCP throughput vs. link frame error rate, number of nodes = 4 , frame size = 1500 bytes, max MAC retry limit = 6.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[2] D. Kliazovich, N. Halima, and F. Granelli, "Cross-layer Error Recovery optimization in WiFi networks," in *Tyrrhenian International Workshop on Digital Communication (TIWDC)*, September 2007.

[3] J. Postel, *Transmission Control Protocol*, September 1981.

[4] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," June 2001.

[5] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," in *ACM Wireless Networks*, 1995.

[6] C. Parsa and J. J. Garcia-Luna-Aceves, "Improving TCP performance over wireless networks at the link layer," *Mob. Netw. Appl.*, vol. 5, no. 1, pp. 57–71, 2000.

[7] S. Kopparty, S. Krishnamurthy, M. Faloutsos, and S. Tripathi, "Split-TCP for Mobile Ad-Hoc Networks," 2002.

[8] B. A. Bakre, "I-TCP:Indirect TCP for mobile hosts," in *In Proceedings of the 15th International Conference on Distributed Computing Systems*, 1995, pp. 136–143.

[9] R. Krishnan, M. Allman, C. Partridge, J. Ster-benz, and W. Ivancic, "Explicit Transport Error Notification (ETEN) for error-prone wireless and satellite networks," 2002.

[10] D. Welzl, "TCP Corruption Notification Options," June 2004. [Online]. Available: http://www.welzl.at/research/publications/draft-welzl-tcp-corruption-00.txt

[11] M. Welzl, M. Rossi, A. Fumagalli, and M. Tacca, "TCP/IP over IEEE 802.11b WLAN: the challenge of harnessing known-corrupt data," in *ICC 08*, May 2008.

[12] R. Stewart, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, *Stream Control Transmission Protocol (SCTP)*, October 2000.

[13] J. Stone, R. Stewart, and D. Otis, *Stream Control Transmission Protocol (SCTP) Checksum Change*, September 2002.

[14] K. Ramakrishnan, S. Floyd, and D. Black, *The Addition of Explicit Congestion Notification (ECN) to IP*, September 2001.

[15] R. Balan, B. Lee, K. Kumar, Jacob, W. Seah, and A. Ananda, "TCP HACK: TCP Header Checksum Option to improve performance over lossy links," in *20th IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2001.

[16] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno modification to TCP's Fast Recovery Algorithm," April 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3782.txt

[17] N. Agarwal, "Cooperative MAC Protocols for IEEE 802.11 Ad-Hoc Networks," Master's thesis, The University of Texas at Dallas, 2006.

[18] N. Agarwal, D. ChanneGowda, L. N. Kannan, M. Tacca, and A. Fumagalli, "IEEE 802.11b cooperative Protocols:A performance study," in *IFIP/TC6 NETWORKING 2007*, vol. 4479, Atlanta, GA, USA, May 2007, pp. 415–426.