Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

11th International Conference
on Electrical Engineering
ICEENG 2018

An Enhanced Transport Layer Protocol for Cognitive Mobile Ad Hoc Networks

Mohamed Naguib*, Yahya Mohasseb*, Hisham Dahshan*, and Ahmed Elghandour*

Abstract

The cognitive mobile ad hoc network is considered as a promising solution for the scarcity of wireless resources. The performance of the transport layer is degraded due to the cognitive radio characteristics like spectrum sensing and spectrum switching. This paper discusses the effects of cognitive operations on the classical TCP. A comparison between the existing transport layer protocols limitations and performances under the cognitive networks circumstances is introduced. Also, this paper proposes a flow control during the channel switching. A feedback packet to enhance the performance of TCP during the sensing period. Improvement in the throughput of the proposed protocol is shown in simulation results.

Keywords

Transport protocol, cogintive radio, mobile ad hoc networks, congestion window, flow control, spectrum sensing.

1. Introduction

The Cognitive Mobile Ad Hoc Networks or CoMANET is an effective solution to the scarcity of the wireless resources because it enables the Secondary Users (SU) to transmit on the vacant portions of the spectrum with no impact on the primary users (PU) [1].
CoMANET have many research topics such, spectrum sensing framework to achieve maximum sensing efficiency [2], routing protocols that predicts the destination location to extend the route suitably to the new location of cognitive radio user [3], studying the impact of CoMANET characteristics over route formation and over TCP performance [4].

In the area of TCP, The well-known classical TCP approaches [5], [6], [7], suffer from poor performance over the Mobile Ad Hoc Net (MANET). The main factors that cause this poor performance are the congestion, nodes mobility and the channel uncertainty, additional factors which are caused by CoMANET such as: spectrum sensing, spectrum switching and PU's existence [8].

---

* Egyptian Armed Forces.

In MANET the main problem at the transport layer level is the performance degradation due to congestion, packet losses due to fading or shadowing, and packet losses due to mobility. The TCP source needs to distinguish between these factors, to update its Congestion Window ($cwnd$) in a manner which is suitable for each case. In the case of congestion, the greater queueing delay in the buffer of the intermediate node the greater Round-Trip Time (RTT). If the RTT exceeds a threshold, a Retransmission Time Out (RTO) event causes the TCP to reduce its congestion window to one segment. In the second case, packet losses due to channel fading or shadowing prevent the sender from receiving an acknowledgment (ACK), the sender misunderstands this as a congestion event. In case of packet losses due to mobility, also may be mistaken by the sender.

All of previous performance degradation cases are equally applicable to CoMANET in addition to the following cases which result from the cognitive operations. First, periodic spectrum sensing, which leads to increasing the RTT and in some cases the packets are dropped after the maximum retry threshold is reached at the MAC layer. Second, if the SU detects a PU activity during the sensing period, then the SU must change its channel which may lead to change in the bandwidth. This change affects the transmission time and the RTT of the packet, the sender must change the $cwnd$ to adapt the new characteristics of the channel. Third case, If the SU misdetects the PU existence, this leads to interference between them and packets loss.

The TCP Westwood protocol [9] was proposed as a solution to reduce the performance degradation in a single hop connection. In this protocol the Bandwidth Delay Product (BDP) and the minimum RTT are used to updates the congestion window and the Slow Start Threshold ($ssthresh$), thereby prevents large reduction of the congestion window. The estimated bandwidth used in TCP Westwood depends on the interval between two consecutive ACKs. ACK filtering reduces the interval time between ACKs, so the bandwidth is overestimated. TCP Westwood+ [10] is proposed to solve the problem of TCP Westwood, where the estimated bandwidth depends on the size of data acknowledged and the RTT.

The Ad Hoc TCP Protocol (ATCP) [11] uses feedback information from intermediate nodes to differentiate between the different packet loss reasons while a Reliable Transport Protocol for Ad Hoc Networks (ATP) [12] also uses feedback information and cross-layer cooperation.
In TCP with Out of Order Detection and Response (TCP DOOR) [13] and TCP with fixed RTO [14] an analysis of TCP events to determine the reason for packet loss is introduced. In [15] an enhancement to the performance of TCP is proposed without modifying the existing standard by modifying link layers.

In [16] cogTCPE proposes an enhancement of the estimation of bandwidth used in TCP Westwood. This transport protocol is suitable for single hop cognitive radio link, when it is used for multihop cognitive radio link a buffer overflow in the switching is caused. TCP CRAHN [8], TCP COBA [17] and TFRC-CR [18] are the existing protocols that support multihop communication in CoMANET.

TFRC-CR is based on TFRC and it does not need feedback information from intermediate nodes and collaboration with lower layers. It allows a fine adjustment of the transmission rate by updating the congestion window continuously. TFRC-CR uses the recent Federal Communications Commission (FCC) spectrum database information, so the sender is assumed to be able to obtain various information such as which channel in the path is used by the PU.

TCP-CRAHN was proposed to enhance the performance of the transport layer in CoMANET [8]. It modifies the TCP NEWRENO to update its congestion window in a proper way to adapt the Cognitive characteristics. It provides an Explicit Feedback Notification (ECN), which alert the TCP source that the buffer of an intermediate node reached the limit, so the source should decrease its congestion window to one segment. A flow control is used during the sensing time to prevent buffer overflow in the intermediate nodes. To achieve efficient use of wireless resource it updates its congestion window after spectrum switching based on the BDP. A mobility prediction framework is used to alert the source about the intermediate nodes motion, consequently the source reduce it transmission rate to reduce packet losses due to mobility.

TCP-COBA used the classical transport layer protocol (TCP-NEWRENO) [5] except for the time after channel switching, it updates the congestion window after spectrum switching in efficient way which is better than the way which is used in TCP CRAHN [17]. The previous protocols cease the transmission during the spectrum switching period, also they suffer from RTO events during the sensing periods because the MAC layer drops a packet after a maximum retry limit, also sensing periods increase the RTT which may result in RTO events.

A new transport layer protocol based on TCP-CRAHN is proposed in this paper to avoid the effect of packets dropped due to the maximum retry limit during the sensing period and the channel switching period. Another improvement to the transport protocol is made by making it uses the available resources during the switching period. This is achieved by controlling the flow of packets, in contrary the other protocols which cease the transmission during this time.

The rest of the paper is organized as follows: we present an overview about CRAHN and COBA in section 2. Our motivation for the new protocol in section 3. The system description of the proposed protocol is presented in section 4. Section 5 presents the performance evaluation, and finally section 6 concludes the paper.

2. Overview of CoMANET Transport Layer Protocols

The proposed protocol in this paper is based on TCP CRAHN and TCP COBA, so these protocols is discussed in details in this section.

In [8] TCP CRAHN was proposed to consider the special characteristics of CoMANET. The authors modified the classical transport protocol TCP NEWRENO in the following cases:

(a) Connection Establishment: The synchronization (SYN) packet is modified to carry the following information (node ID, timestamp, sensing duration), this information is sent to the source through the ACK packet. The source creates a schedule for sensing operation which contains when each node starts sensing the channel and its sensing duration.

(b) Normal State: In this state an ECN packet is sent to the source in response to congestion event, which enforce the source to reduce its $cwnd$ to one segment to prevent buffer overflow. Link layer information which is used to update the congestion window during the sensing time and after spectrum switching is appended to the data packets flowing to the destination, which is sent back to the source with every new ACK. This information is residual buffer space (B_i^f) for each node, observed link bandwidth ($W_{i,i+1}$), and total link latency ($L_{i,i+1}^T$).

(c) Spectrum Sensing State: A flow control is designed to prevent buffer overflow in the intermediate nodes. During this time, the TCP sender continues sending segments, but the transmission is controlled by the effective window $ewnd$ which is calculated as follows:

$$ewnd = \min(cwnd, rwnd, B_{i-1}^{f}). \quad (1)$$

where $rwnd$ is the received window advertised by the receiver, and B_{i-1}^{f} is the residual buffer at the previous node of that executing sensing. The sensing time (t_i^s) is calculated as follows [2]:

$$t_i^s = \frac{1}{W\gamma^2}[Q^{-1}(P_f) + (\gamma+1)Q^{-1}(\frac{P_{off}P_f}{P_{on}})]^2. \quad (2)$$

It based on the channel bandwidth (W), the external signal to noise ratio (γ), the probability of on period (P_{on}), off period (P_{off}), the standard Q function, and it minimizes the probability of missed detection P_f. A regulation scheme is proposed to change the sensing time for each node, but it depends on a central decision from the TCP source. However, it may happen that a node receives messages from different TCP sources which will force the node to have more than one sensing schedule.

(d) Spectrum Switching State: When a PU uses the channel then the SU must switch away from this channel, an Explicit Pause Notification (EPN) is sent to the source to freeze its connection. After the coordination of a new channel, a new channel information CHN packet is sent to the source with the new channel information (new $W_{i,i+1}$, new $L_{i,i+1}^T$). The source estimates the bottleneck bandwidth W_b = min $\{W_{l,l+1}\}$, also estimates the new RTT, where RTT = RTT' + $L_{i,i-1}^T$ - $L'^T_{i,i-1}$. If the channel bandwidth is changed only by 20%, then no scaling is needed. If the change is larger than 20% then the congestion window is changed as follows:

$$cwnd = \alpha_c W_b RTT. \quad (3)$$

The factor α_c in (3) is used to prevent the risk of overestimating the $cwnd$.

The TCP COBA protocol [17] is the same as TCP NEWRENO except after the spectrum switching. An efficient algorithm is used to update the congestion window after spectrum switching which depends on link layer information from intermediate nodes. This information is collected in four cases: first case in three-way handshake, second case in forwarding each data packet, third case in start of channel switching, and the last case at the end of channel switching. This information is:(a) BW_i is the bandwidth of link i, $(b)L_{i,i+1}^T$ is the RTT between two neighboring nodes $(i, i+1)$, and $(c)B_i^f$ is the residual buffer space in node i. In this protocol it is assumed that the feedback information is available only from a node changing its channel. The control of TCP COBA is designed in a manner which is completely similar to TCP NEWRENO except during channel switching, to prevent performance degradation in case of feedback information loss. It updates the congestion window based on the available buffer space in the bottleneck, if necessary, in addition to the BDP. In this protocol if the W_b or RTT is changed after channel switching by more than 20% of the previous value, then the congestion window must be updated. COBA freezes data transmission during the channel switching. An intermediate node sends a feedback packet to the source with the new channel BW_i and $L_{i,i+1}^T$ after a channel switching is done. The TCP source updates its congestion window based on the minimum available resource as follows: $\min(\frac{BW_i}{Num.of\ flows}, \frac{B_i^f}{Num.of\ flows})$, consequently preventing buffer overflow.

In COBA, if the condition of updating the congestion window after spectrum change is occurred, the source updates the congestion window as in the following two cases:

1. The bottleneck node is located in the path between the source and the switching node. In this case the BDP using equation (5) is good enough to update its congestion window.

2. The bottleneck node is the switched node or in the path between the switching node and the destination. In this case the $cwnd$ should be limited to prevent buffer overflow, so the new value of $cwnd$ is calculated based on the following equation:

$$cwnd'[pkts] = B_i^f \frac{BW^*}{BW^* - W'_b} + cwnd. \quad (4)$$

Where, BW^* is the minimum bandwidth between source and node previous to the switching node, W'_b is the bottleneck bandwidth after channel switching. COBA took into account BDP which is suitable for updating the $cwnd$ using Eq. (5), so the effective window is calculated using Eq. (6):

$$BDP[pkts] = \frac{W'_b[b/s].RTT_{new}[s]}{PS[b/pkt]}, \quad (5)$$

$$cwnd''[pkts] = \min(BDP, cwnd'). \quad (6)$$

Where, $PS[b/pkt]$ is the packet size, and $RTT_{new} = L_{1,2}^T + \cdots + L_{i-1,i}^T + L'^T_{i,i+1} + \cdots$.

3. Motivation

In this section, the extension of NS-2 simulator which models the PU activity and the multichannel operation performed by CoMANET [4] is used. However, since no support for CRAHN and COBA is provided, these protocols are implemented and used. A study on the effect of spectrum sensing and the channel switching time is done.

3.1. Spectrum Sensing

Nodes periodically sense the channel for a period of time to ensure that it is not used by the PU before starting to use it. During this time the connection is virtually disconnected, which means the transport layer continues creates new segments and send it to the lower layers but the MAC layer does not send any packets.
In Fig 2, the impact of sensing time on the TCP CRAHN and TCP COBA performance is shown, it is assumed that there is no PU activity on the channel used. This figure illustrates the congestion window of a single hop communication at different sensing times.

CRAHN will have different response from COBA only during the sensing time, because CRAHN updates its congestion window using equation (1). CRAHN also updates the congestion window every $L_{i,i+1}^T$ between two nodes to reduce B_i^f by one packet and estimates the new $ewnd$. COBA is the same with the TCP NEWRENO, which means that the congestion window remains constant during the sensing time. From Fig 2 it is noticed that the two protocols curves are almost the same except for the sensing duration, when the sensing time = 0.1 second both COBA and CRAHN enters the fast-retransmit state. This case happens because during sensing time MAC layer drops packet after a retry limit (L) is reached (L = 7 in the MAC DCF 802.11 standard), but both CRAHN and COBA misunderstand this situation as congestion and reduce the $cwnd$ to half and send the dropped packet again. When sensing time = 0.5 second RTT becomes larger and beside the dropped packets during the sensing time, so COBA and CRAHN suffer from RTO events which force the $cwnd$ to be reduced to one segment and enters the slow start phase.

3.2 Spectrum Change Effect

In Fig 1, single hop communication with sensing time 0.3 s, and transmission time 5.0 s. At time 20 second the PU appears on the channel which force the SU user to switch to a new channel. In this case both COBA and CRAHN freeze the transmission and reduce the $cwnd$. In Fig 1. (b) nodes starts sensing at time 20 seconds which detects the PU, so they stop transmission and after spectrum cahnge period end the source updates the $cwnd$ depending on the BDP. We notice that both COBA and CRAHN neglects the spectrum change period, which may be used to increase the throughput.

4. TCP NEWCRAHN

In this section, we present the assumptions and the system description of our proposed protocol TCP NEWCRAHN.

4.1. Assumptions

It is assumed that each node has a list of different unoccupied channels with different raw bandwidths and may belong to different spectrum bands. Each channel has a PU as in [4] with the knowledge of arrival (α) and departure (β) rates for each channel. CSMA/CA at MAC layer that has a pre-decided Common Control Channel (CCC) is used. As in [8], the proposed protocol does not depend on probabilistic sensing times. Therefore, nodes sense their channels at regular intervals [2]. Also, it is assumed that all nodes are synchronized and they can sense the channel at the same time [17].

4.2. System Description

By studying CRAHN and COBA, it is obvious that COBA considers only the spectrum change feature of the CoMANET. COBA as mentioned in Sec. 2 is the same as NEWRENO except for updating $cwnd$ after spectrum change. In CRAHN, congestion is considered by using ECN, using a flow control during spectrum sensing to prevent buffer overflow, updating the $cwnd$ after spectrum change, and introducing a mobility prediction to reduce packets dropping because of mobility.

Almost all CoMANET characteristics are considered in the proposed protocol. The proposed protocol extends CRAHN protocol to overcome the shortage of CRAHN and COBA in two cases: dropped packets during spectrum sensing and freezing the connection during spectrum change.

4.2.1. Spectrum Sensing

A new feedback packet called (DTCP) is introduced in the proposed protocol in case of dropping a packet during spectrum sensing. If the MAC layer receives a packet from the buffer when the node performs spectrum sensing, the MAC layer tries to send it. Because the MAC layer is busy by sensing the channel and after L retransmission times, the packet is dropped and a DTCP packet is sent to the source. When DTCP packet is received by the source, it knows that the corresponding packet is dropped due to sensing operation. Therefore, it retransmits the dropped packet again and does not reduce the $cwnd$. In this case, the source does not misunderstand that event as congestion so there is no need to reduce the transmission rate, which is not the case in CRAHN and COBA.
The same flow control during the spectrum sensing used in CRAHN is used in the proposed protocol. The effective window at the sender can be computed using the following equation:

$$ewnd = \min\left(cwnd, B_S^f\right). \quad (7)$$

Where, B_S^f is the residual buffer in the source node. The residual buffer in equation (7) is only the source buffer, because all nodes perform periodical sensing at the same time and only the buffer of the source is available.

4.2.2. Spectrum Switching

In this phase, CRAHN is modified in two stages: (A) During the spectrum switching. (B) After the spectrum switching.

A. During Spectrum Switching

A flow control is used to allow the source continuously produces more segments during the switching time. When a node starts to switch its spectrum, an EPN is sent to the source and the source then updates its $cwnd$ according to the following equation:

$$ewnd_{Spec.Swtiching} = \min\left(cwnd, B_{min}^{f}\right). \quad (8)$$

Where, $ewnd_{Spec.Switching}$ is the effective window during spectrum switching, and B_{min}^{f} is the minimum residual buffer of all nodes from the source to the switching node. This control allows resuming of transmission during the switching period and prevents buffer overflow for the intermediate node.

B. After Spectrum switching

The way which is used by CRAHN to update its $cwnd$ after spectrum switching is modified because it can excessively increase the $cwnd$ compared to the available buffer size as mentioned in [17]. COBA updates $cwnd$ appropriately in response to change in the bottleneck bandwidth (W_b) and RTT, so the $cwnd$ is updated after spectrum change in the same way like in COBA using equations (4), (5), and (6).

5. Performance Evaluation

In this section, the simulation results of the proposed protocol will be presented. The proposed TCP NEWCRAHN is implemented as an extension of NS-2 simulator used in [4]. The network consists of five nodes connected in a chain topology with five different channels. The simulation area is $1000 \times 1000\ m^2$. The transmission range is 250 m. There are five different possible channels bandwidth given by {2, 4, 1, 2, 4} Mbps. Each channel has a corresponding PU located in the range of a secondary node. The evaluation of the proposed protocol is performed to show the impact of spectrum sensing and spectrum change on the size of the congestion window as shown in Fig 3, and Fig 4. The effect of sensing time on throughput is illustrated in Fig 5. Also, the performance of the proposed protocol with varying the number of hops is shown in Fig 6.

5.1. Spectrum sensing

The evaluation of NEWCRAHN during spectrum sensing state is carried out by observing the improvement in the $cwnd$ and the throughput, resulting from the feedback during sensing time. In Fig 3, it is observed that the NEWCRAHN does not suffer from time out events like CRAHN and COBA in the same single hop communication scenario. This great improvement is achieved because the feedback informs the source that there is no congestion and no need for reducing the $cwnd$. These results under assumptions of constant sensing time t^s = 0.5 s, and constant transmission time T_P = 5.0 s.

5.2. Spectrum Change

In Fig 5, the effect of spectrum changes on the performance of NEWCRAHN, COBA, and CRAHN is shown. It is shown that NEWCRAHN does not cease the transmission during spectrum change and updates the $ewnd$ appropriately in a manner that prevents buffer overflow in the intermediate nodes.

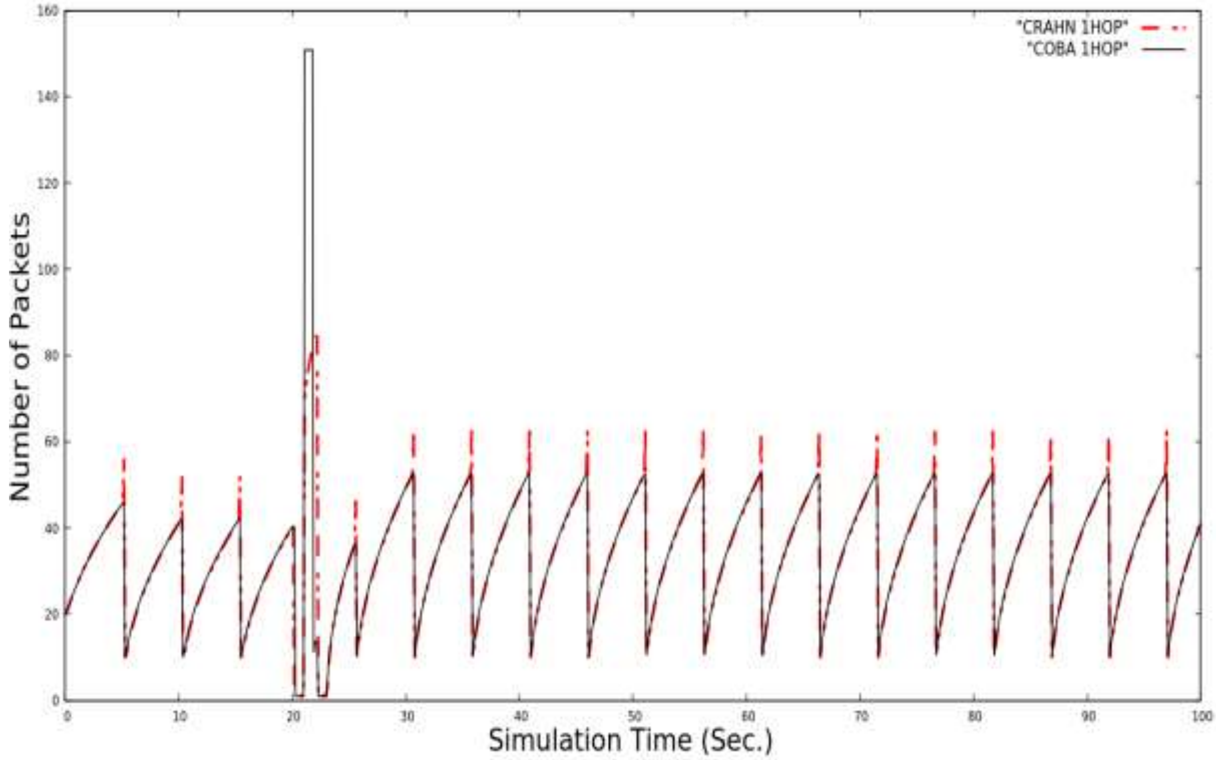
Fig 5 shows that the proposed TCP NEWCRAHN protocol outperforms COBA, CRAHN, and NEWRENO because it does not stop the transmission using an $ewnd$ to transmit packets but it uses the feedback DTCP packet. It is observed that the end to end throughput of the proposed protocol is improved significantly because the proposed NEWCRAHN reduces the RTO events. Another reason for the improvement is that the proposed NEWCRAHN does not stop transmission during spectrum sensing and spectrum switching but transmits at a reduced rate to prevents buffer overflow.

Fig 6 shows that TCP NEWCRAHN improves the performance of the multi-hop communications. Each node in the path increases the total delay by periodical spectrum sensing and spectrum switching. The excess delay during spectrum sensing is predicted by NEWCRAHN and CRAHN, but it is not predicted by NEWRENO and COBA.

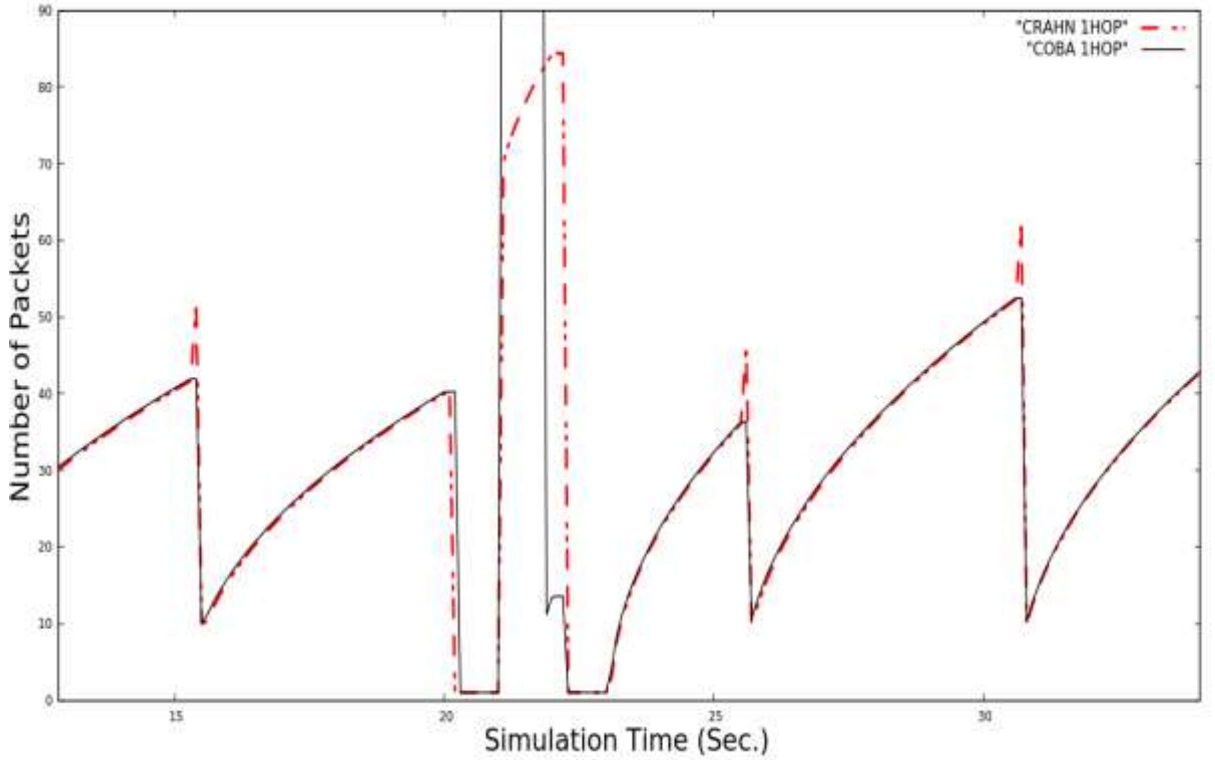
6. **Conclusion**

This paper focused on transport protocols in cognitive mobile Ad Hoc networks. We study the effect of the spectrum sensing and switching the spectrum which is made by cognitive users. The proposed transport protocol makes an efficeint use of the spectrum during channel switching. Results show also that the proposed protocol reduces the RTO events during sensing time.

Figures



(a)



(b)

Fig 1: (a) *cwnd* size in response to spectrum change. (b) focusing on the switching time

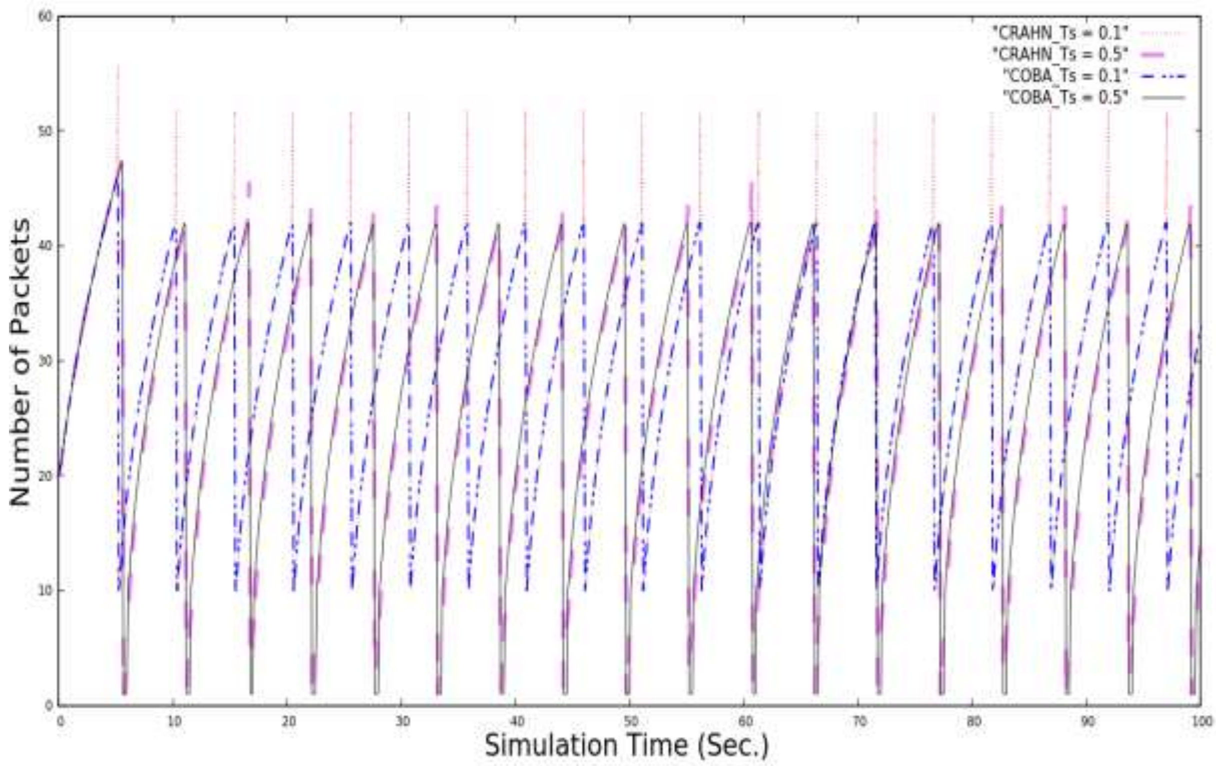


Fig 2: *cwnd* size under different sensing time

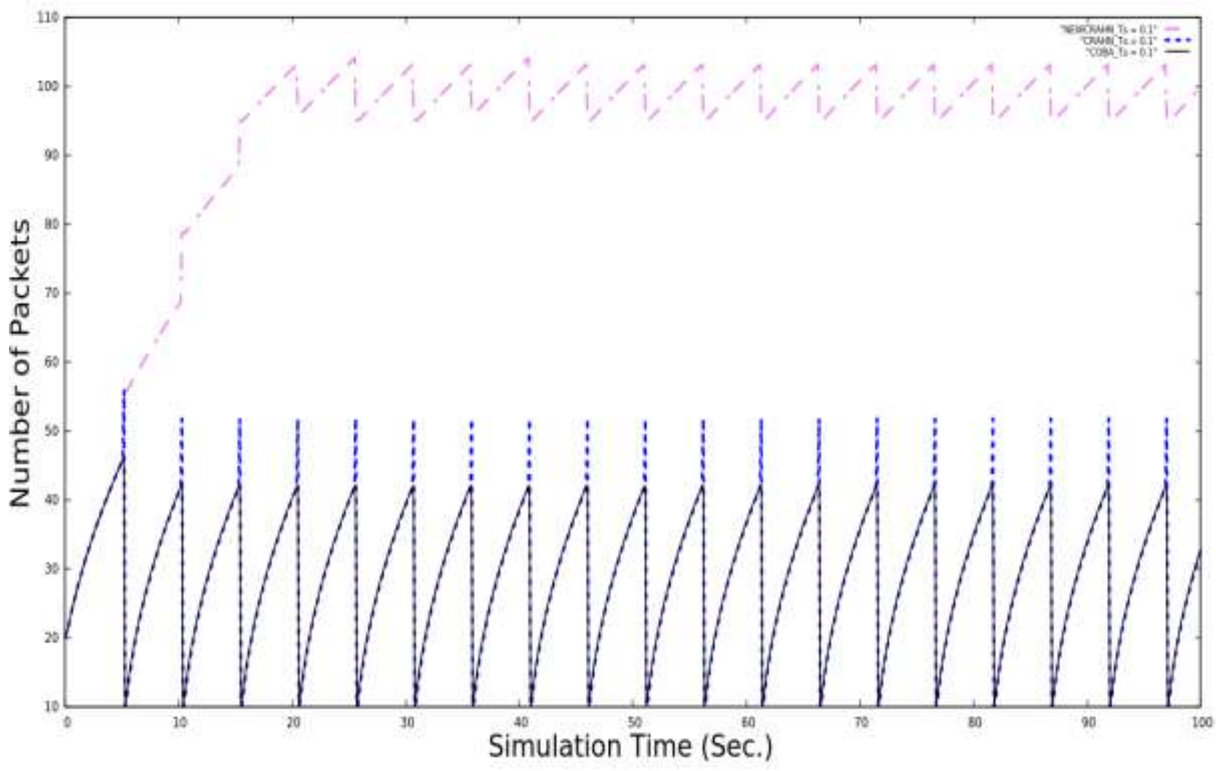


Fig 3: *cwnd* size in response to spectrum sensing.

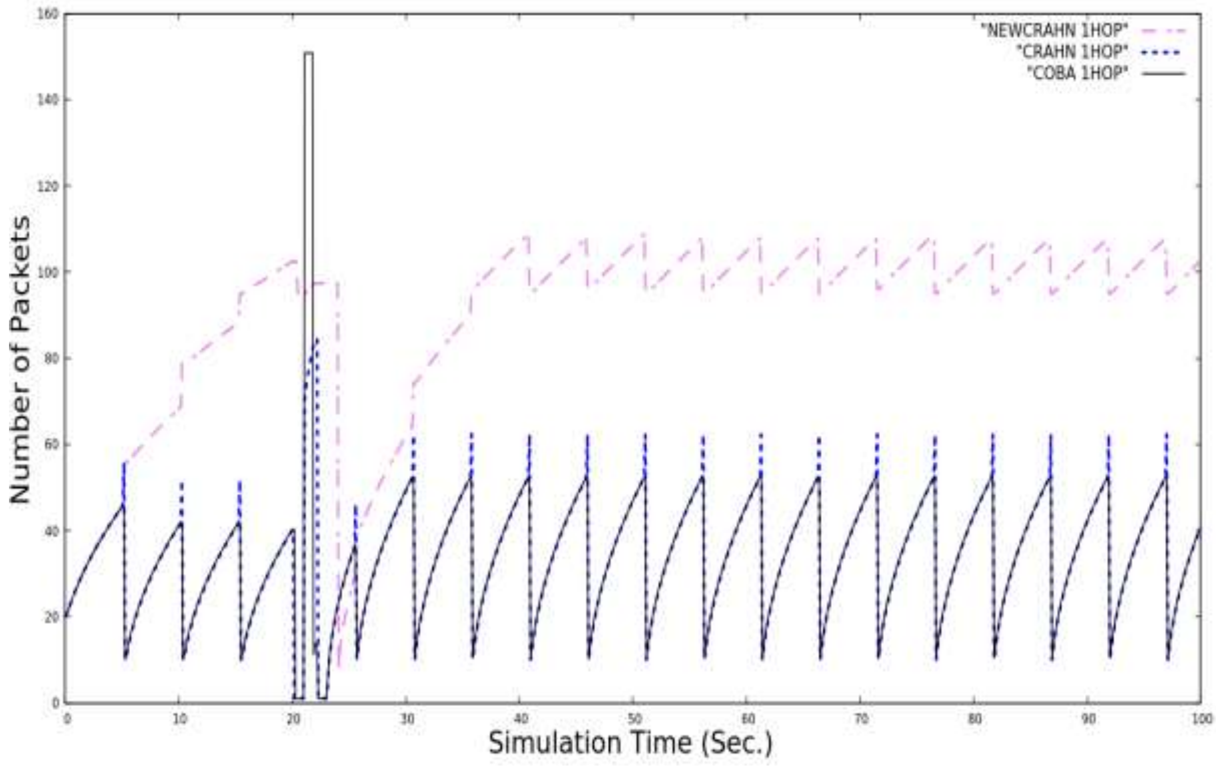


Fig 4: $cwnd$ size in response to spectrum change.

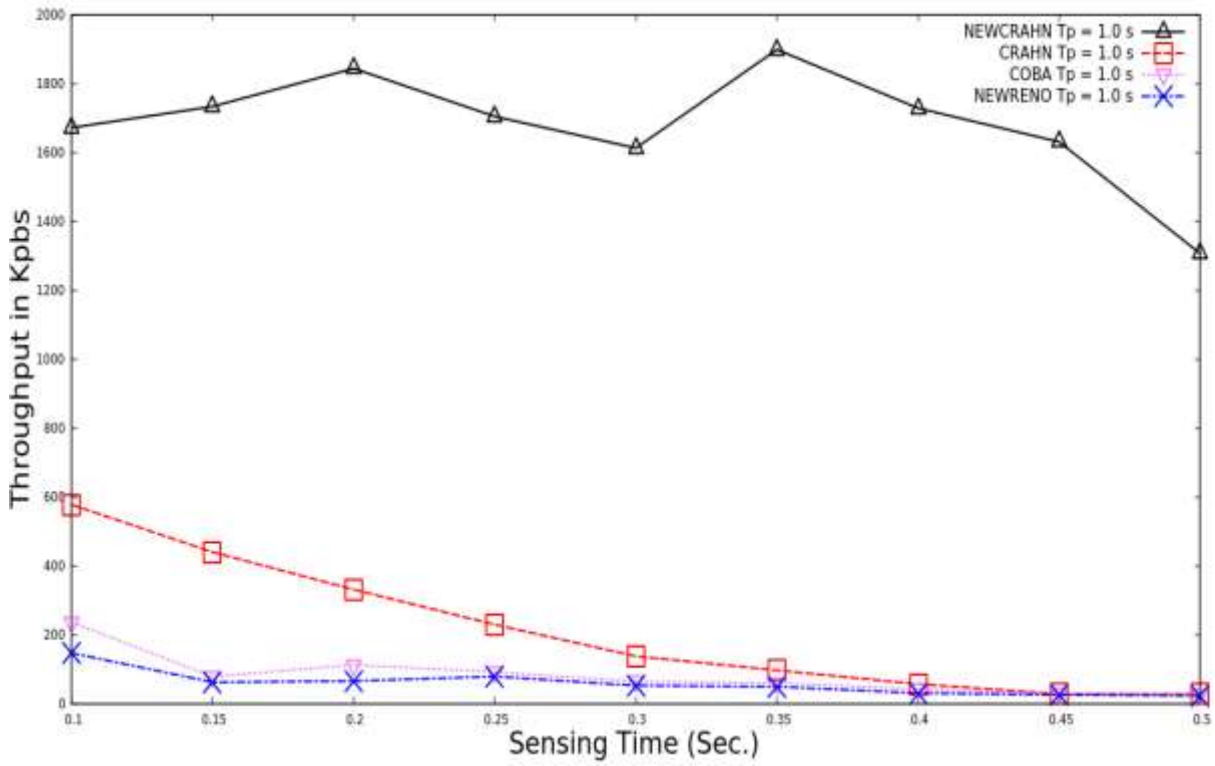


Fig 5: Throughput as a function of sensing time.

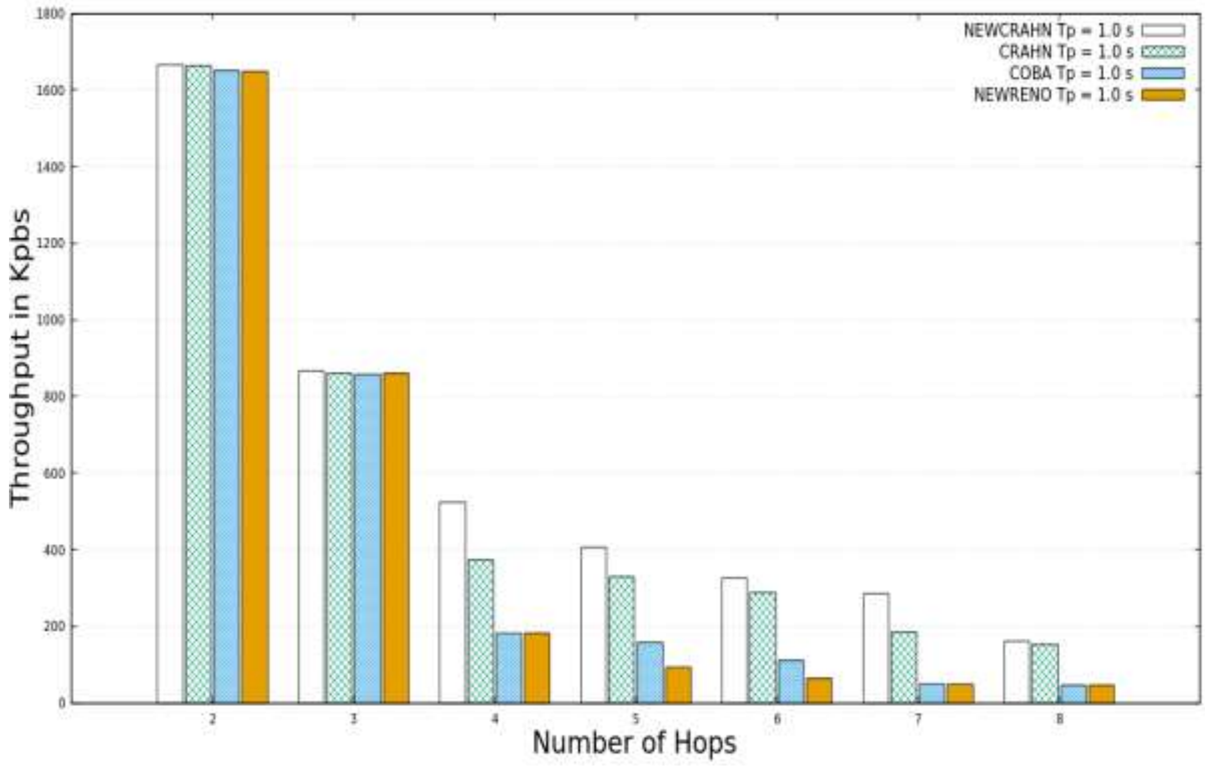


Fig 6: Throughput as a function of number of hops.

References

[1] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, “CRAHNs: Cognitive radio ad hoc networks,” *Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, 2009.
[2] W. Y. Lee and I. F. Akyildiz, “Optimal spectrum sensing framework for cognitive radio networks,” *IEEE Trans. Wirel. Commun.*, vol. 7, no. 10, pp. 3845–3857, 2008.
[3] K. R. Chowdhury and M. D. Felice, “Search: A routing protocol for mobile cognitive radio ad-hoc networks,” *Comput. Commun.*, vol. 32, no. 18, pp. 1983–1997, 2009.
[4] M. Di Felice, K. R. Chowdhury, W. Kim, A. Kassler, and L. Bononi, “End-to-End Protocols for Cognitive Radio Ad Hoc Networks: An Evaluation Study,” *Perform. Eval.*, vol. 68, no. 9, pp. 859–875, 2011.
[5] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, “The NewReno Modification to TCP’s Fast Recovery Algorithm,” *IETF RFC 6582*, 2012.
[6] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment Options,” *IETF RFC 2018*, 2018.
[7] L. S. Brakmo and L. L. Peterson, “TCP Vegas: End-to-End Congestion Avoidance on a Global Internet,” *JSAC*, vol. 13, no. 8, pp. 1465–1480, 1995.
[8] K. R. Chowdhury, M. Di Felice, and I. F. Akyildiz, “TCP CRAHN: A transport control protocol for cognitive radio ad hoc networks,” *IEEE Trans. Mob. Comput.*, vol. 12, no. 4, pp. 790–803, 2013.
[9] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCPWestwood: End-to-End Bandwidth Estimation for Enhanced Transport overWireless Links,” *Proc. 7th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom ’01*, no. January, pp. 287–297, 2001.
[10] L. a. Grieco and S. Mascolo, “Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control,” *ACM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 25–38, 2004.
[11] J. Liu and S. Singh, “ATCP: TCP for mobile ad hoc networks,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 7, pp. 1300–1315, 2001.
[12] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar, “ATP: a reliable transport protocol for ad hoc networks,” *IEEE Trans. Mob. Comput.*, vol. 4, no. 6, pp. 588–603, 2005.
[13] F. Wang and Y. Zhang, “Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response,” *Symp. Mob. ad hoc Netw.*, pp. 217–225, 2002.
[14] T. D. Dyer and R. V. Boppana, “A comparison of TCP performance over three routing protocols for mobile ad hoc networks,” *Proc. 2nd ACM Int. Symp. Mob. ad hoc Netw. Comput.*, pp. 56–66, 2001.
[15] U. N. Red, K. Xu, and M. Gerla, “Enhancing TCP Fairness in Ad Hoc Wireless Networks Categories and Subject Descriptors,” *Proc. 9th Annu. Int. Conf. Mob. Comput. Netw.*, pp. 16–28, 2003.
[16] D. Sarkar and H. Narayan, “Transport layer protocols for cognitive networks,” *Proc. - IEEE INFOCOM*, pp. 1–6, 2010.
[17] K. Tsukamoto, S. Koba, M. Tsuru, and Y. Oie, “Cognitive radio-aware transport protocol for mobile ad hoc networks,” *IEEE Trans. Mob. Comput.*, vol. 14, no. 2, pp. 288–301, 2015.
[18] A. K. Al-ali and K. Chowdhury, “Ad Hoc Networks TFRC-CR : An equation-based transport protocol for cognitive radio networks,” *Proc. - IEEE ICNC*, vol. 11, no. 6, pp. 1836–1847, 2013.