ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

Performance Analysis of Transport Control Protocol for Cognitive Radio Ad Hoc Network

Divya K[1], Nisha D[2], Surya M[3]

PG Student, ECE Department, Sriguru Institute of Technology, Coimbatore, India[1]

PG Student, ECE Department, Sriguru Institute of Technology, Coimbatore, India[2]

Assistant Professor, ECE Department, Sriguru Institute of Technology, Coimbatore, India[3]

Abstract: A Cognitive radio (CR) network is an intelligent radio network that can be programmed and configured dynamically. It allows users to opportunistically transmit in the Primary User's spectrum bands without any interference. Relevantly, frequent sensing of the spectrum and spectrum availability variation with time by the CR users has an effect on the OSI layer protocol performance, especially at the transport layer. This paper investigates the drawbacks of TCP newReno in a CR ad hoc network environment, and put forwarded TCP CRAHN. This approach also includes incorporating spectrum awareness by a combined method of explicit feedback from the intermediate nodes and the receiver node. During the quality of being mobile of unlicensed users there occurs a scheduling problem which increases the delay. So, a scheduling algorithm is being introduced to reduce the delay and enhanced the throughput.

Keywords: Cognitive radio, congestion control, flow control, spectrum sensing, TCP

I. INTRODUCTION

Cognitive radio (CR) is a form of wireless communication in which a transceiver can intelligently detect which communication channels are in use and which are not, and instantly move into vacant channels while avoiding occupied ones. This optimizes the use of available radio-frequency (RF) spectrum while minimizing interference to other users.TCP is the most widely-deployed transport layer protocol on the Internet. TCP connects two applications running on two hosts. The host that establishes the connection is called the sender, and the other host is called the receiver. The sender gets data from the application process and stores it in a buffer. The sender periodically takes some chunk of data from the buffer, encapsulates the chunk into what is called a segment, assigns a sequence number to the segment, and then sends the segment to the receiver. On receiving each segment, the receiver sends an acknowledgement message to the sender. The receiver uses the sequence numbers to reorder the received segments correctly. In cognitive radio networks, the arrival of a licensed user interrupts data transmission. Also, spectrum sensing causes temporary disconnections that lead to TCP time-outs. Unfortunately, TCP assumes that these time-outs are caused by congestion in the path. Hence, a TCP ender unnecessarily decreases the rate at which it sends segments to the receiver.

TCP is employed to work in cognitive radio networks. In addition to the events that lead to the reduction of sending rate, as we saw in the previous section, there are other new events. Suppose the sender is transmitting at a rate and an LU arrived. The sender then has to look for another channel and resume transmission. However, TCP always starts transmission at a slower rate. Hence, arrival of an LU leads to the reduction of the sending rate. Therefore, frequent arrival of LUs will lead to frequent reduction of the sending rate. Another event that leads to reduction of sending rate is when the receiver is engaged in spectrum sensing. Here, we assumed that the receiver is using a single transceiver. Therefore, when in sensing mode, it cannot participate in transmission, hence; the sender will experience a temporary disconnection which will lead to timer expiration. Unfortunately, the TCP sender will assume that the timer expired because of congestion and it will reduce its sending rate. In our design, we will assume a cross layer design. Therefore, we assume that information owes from one layer to another. We will assume that TCP relies on the lower layers for spectrum sensing and spectrum decision. The lower layers notify TCP about which channel to use for transmission and which channel to move to when an LU arrives on the current channel. Therefore, TCP does not have to make any decision regarding which channel to sense.

In our design, we have to prevent the reduction of sending rate due to LU arrival and due to engagement of the receiver in spectrum sensing. The sender can also go into sensing mode, therefore, we have to prevent the sender from reducing its sending rate when transmission resumes. CR Ad Hoc Networks that do not have a centralized entity for obtaining the spectrum usage information in the neighbourhood, or external support in the form of a spectrum broker that enables the sharing of the available spectrum resource. Thus compared to infrastructure-based networks, relying on local decisions makes the problem of node-coordination and end to-end communication considerably more involved. While the mobility of the intermediate nodes and the inherent uncertainty in the wireless channel state are the key factors that affect the reliable end-to-end delivery of data in classical ad-hoc networks, several additional challenges exist in a CRAHN.

Transport protocols constitute a well investigated topic in traditional wireless ad hoc networks, but they are quite

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

unexplored in CR networks. It is well known that classical TCP implementations which run over the Internet (e.g., TCP newReno, TCP Vegas], and TCP SACK perform poorly over wireless links because of the additional packet losses caused by bad channel conditions or by node mobility, which are often misinterpreted as indicators of network congestion. In paper [4] the main objective was that the CR networks are envisaged to solve the problem of spectrum scarcity by making efficient and opportunistic use of frequencies reserved for the use of licensed users of the bands The main challenge in CRAHNs is to integrate these functions in the layers of the protocol stack, so that the CR users can communicate reliably in a distributed manner, over a multi-hop/multi-spectrum environment, without any infrastructure support. In paper [7] the main objective of this transport protocol TP-CRAHN, integrates as an end-to-end metric, the spectrum sensing and switching functionalities in a CR network, apart from the classical concerns of congestion, flow control and connection losses due to node mobility. By relying on updates from the intermediate nodes and the destination feedback, the source maintains information about the network state and responds appropriately by adjusting its transmission rate. Both the papers did not give the information regarding the delay that occur during the switching mobility and also the limitations of TCP NewReno is been rectified. The radio behaviour and potential security threats in cognitive radio networks are investigated in order to successfully deploy CR networks and realize its benefits.

II. TCP CRAHN

The periodic spectrum sensing, channel switching operations, and the awareness of the activity of the Primary Users (PUs) are some of the features that must be integrated into the protocol design. For these reasons, protocol development at the higher layers of the network stack for CR ad hoc networks, involving end-to end communication over multiple hops, is still in a nascent stage. A window-based, TCP-like spectrum-aware transport layer protocol for CR ad-hoc networks, called TCP CRAHN that distinguishes between the different spectrum specific conditions in order to undertake state-dependent recovery actions been proposed. During data transfer CSMA/CA at the Medium Access Control (MAC) layer that has a pre-decided Common Control Channel (CCC) for coordination of the spectrum band and channel. Use a priority queue, Qp at the MAC layer for the TCP CRAHN control packets, which may also be drawn from intermediate positions in Qp.

In a CR network, nodes maintain a list of unoccupied channels (other than the current one in use) that may belong to different spectrum bands. In our work set of channels is identified through spectrum sensing, undertaken during the back off interval following a packet transmission or reception at the link layer. On the current operational channel, an accurate idea of the PU activity is done. For this, we do not rely on probabilistic sensing times. Rather, nodes sense their current channel for the sensing time at regular intervals at the cost of continued network connection.

- Connection establishment state
- Spectrum sensing state
- Normal state
- Spectrum change state
- Mobility predicted state
- Route failure state

A. Connection Establishment State

TCP CRAHN modifies the three-way handshake in TCP newReno so that the source can obtain the sensing schedules of the nodes in the routing path. First, the source sends out a Synchronization (SYN) packet to the destination. An intermediate node, say i, in the routing path appends the following information to the SYN packet: 1) it's ID, 2) a timestamp, and 3) the tuple. The time left before the node starts the next round of spectrum sensing, measured from the timestamp is the constant duration between two successive spectrum sensing events, and tsi is the time taken to complete the sensing in the current cycle. On receiving the SYN packet, the receiver sends a SYN-ACK message to the source. The sensing information collected for each node is piggybacked over the SYN-ACK.

B. Spectrum Sensing State

TCP CRAHN adapts to spectrum sensing through 1) flow control, which prevents buffer overflow for the intermediate nodes during sensing and 2) regulating the sensing time to meet the specified throughput demands the goal of TCP CRAHN is to adapt the flow control mechanism. In TCP, so that the node prior to the sensing node, is not overwhelmed with incoming data packets. If another node j has an overlapping sensing schedule, TCP CRAHN uses the residual buffer space of the previous hop of the node closest to the source during the period of overlap, say i.

When the sensing time of the closest node is completed, the buffer space of node j is used in the ewnd computations the maximum number of bytes of unacknowledged data allowed at the sender is the minimum of the current congestion window, and the receive window advertised by the destination, rwind. The rwind represents the free space in the receiver's buffer that can accommodate additional transmitted packets. During the sensing duration, no ACKs are received by the source and hence the rwind remains unchanged. This also results in a constant cwnd as TCP is self clocked and does not increase in the absence of the receiver ACK.

C. Normal State

The normal state in TCP CRAHN is the default state and resembles the classical functioning of the classical TCP new Reno protocol. Our protocol enters this state when 1) no node in the path is currently engaged in spectrum sensing, 2) there are no connection breaks due to PU arrivals, and 3) no impending route failure is signaled. Thus, the path to the destination remains connected and ACKs sent by the latter are received at the source. The differences between TCP CRAHN in the normal state and the classical TCP are as follows. Feedback through the

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

ACK the intermediate nodes of the path piggyback the following link-layer information over the data packets to the destination, which is then sent to the source through ack's.

D. Spectrum Change State

In the ideal case, the effective bandwidth of the TCP connection is dependent on several factors, such as contention delays and channel errors at the link layer, apart from the raw bandwidth of the channel. In this section, we show how TCP CRAHN scales its cwnd rapidly, say from point B to a different value B'. Its then sends back a link layer ACK to node i to inform the node of its choice. All the coordination up to this point occurs on the old channel. A second set of Probe and ACK messages are then exchanged on the channel to be switched, a confirmation and also to approximately estimate the new link transmission delay times.

E. Mobility Predicted State

In order to address the problem of route failure notification a mobility predicted framework based on Kalman filter-based estimation, which uses the Received Signal Strength (RSS) information from the link layer. The set of Kalman equations similar to the disposition is used for calculating sensor location. But for a simpler scalar case, a single dimension of the received power value is calculated. The nodes of the path monitor the connectivity to their next hop downstream node by measuring the RSS of the ACKs and the periodic beacon messages. At each epoch, the prediction value is compared with the minimum RSS required for receiver operation. If the condition of possible link failure is predicted in the next epoch, the destination is informed, which then sets the Mobility Flag (MF) in the outgoing ACKs.

The source responds to this by limiting the cwnd to the ssthresh and the congestion avoidance phase is never initiated. The aim of this adjustment, cwnd _ ssthresh, is to limit the number of packets injected into the route which has a possibility of an outage, as the CR
specific function of the nodes may delay the arrival of the actual link failure notification. If no ICMP message is received at the source subsequently, signaling that a route failure has indeed occurred or the incoming ACKs do not have the MF flag sent, the mobility prediction state is cancelled and TCP CRAHN reverts back to the state.

F. Route Failure State

The node i sends a destination unreachable message in the form of an ICMP packet if

1. the next hop node i-1 is not reachable based on link layer retries,
2. there is no ongoing spectrum sensing based on the last known schedule, and
3. no EPN message is received at node i signaling a temporary path disconnection due to PU activity. At this stage, the source stops transmission and a fresh connection needs to be formed over the new route by TCP CRAHN.

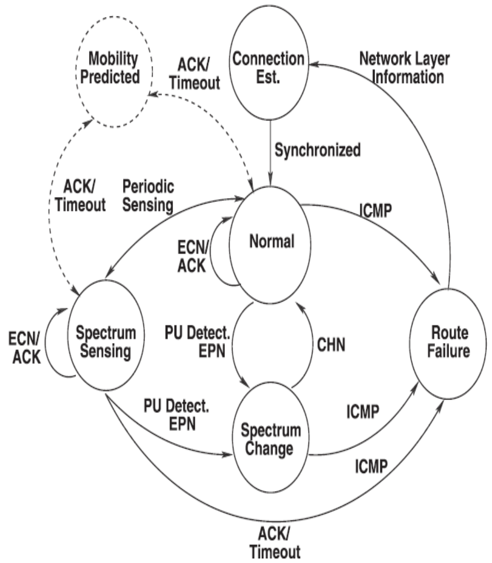
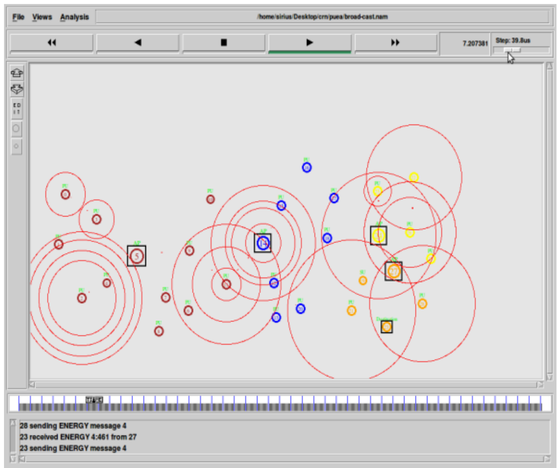
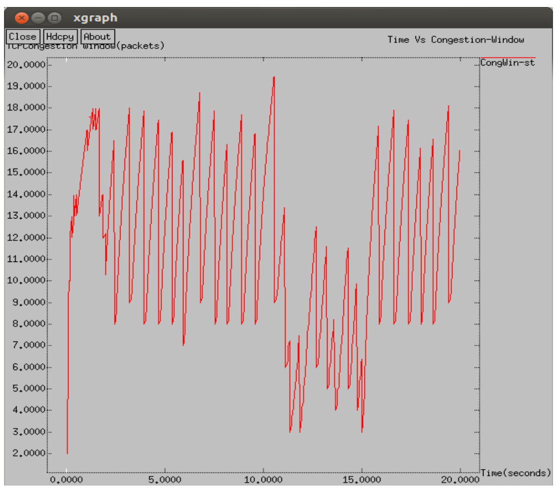


Fig. 1 Model of TCP CRAHN

III. SIMULATION RESULTS



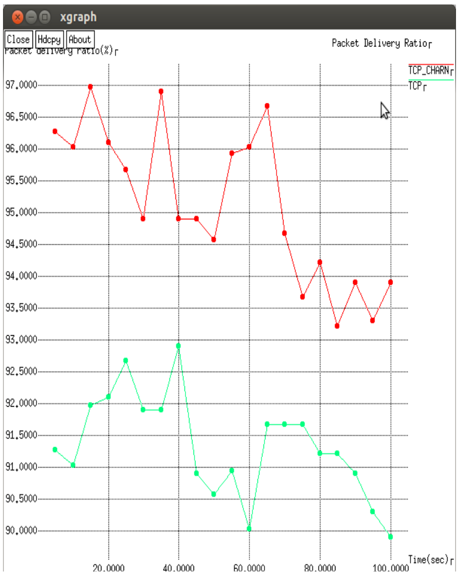
(a)



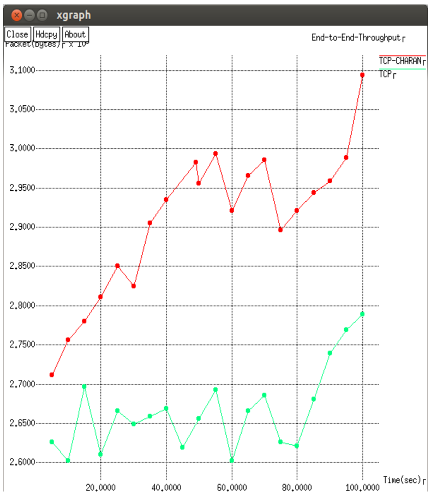
(2)

Fig 2. (a).Transmission and Reception of energy messages: Sending energy messages to the required nodes (b).Time Vs Congestion Window: Variation of the congestion window with time

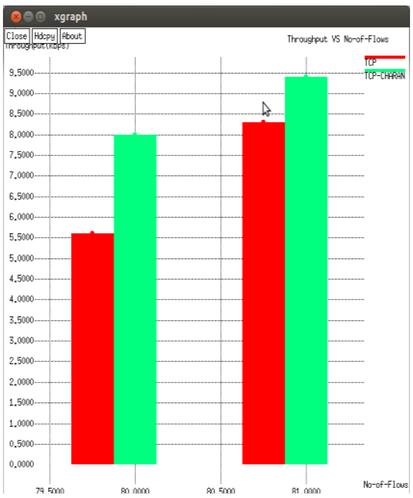
ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 4, April 2014



(a)



(b)



(c)

Fig.3 (a)Packet Delivery Ratio Vs Time: Both the throughputs of TCP CRAHN and TCP have been compared. Figure (b) End-End Throughput Vs Time: The effect of dynamically changing the duration on throughput is shown for TCP and TCP CRAHN. Figure (c) Throughput Vs No. of Flows: The upper bound on the TCP CRAHN and TCP throughput is shown as a function of varying length.

IV. CONCLUSION

A polynomial time heuristic algorithm called S^2DASA to solve our formulated problem. The simulation results show that the throughput that S^2DASA yields is very close to its upper bound. Moreover, S^2DASA is robust to changes in the hardware spectrum switching delay. Furthermore, throughput savings it achieves increase as the number of frequencies in the CRN cell and the hardware switching delay for a unit frequency difference increases. Furthermore, the throughput savings of our algorithm are significant even when there are a small number of SUs, and the savings remain significant as the number of SUs increase.

REFERENCES

[1] A. Capone, L. Fratta, and F. Martignon (2004), “Bandwidth Estimation Schemes for TCP over Wireless Networks,” IEEE Trans. MobileComputing, vol. 3, no. 2, pp. 129-143.
[2] A.O. Bicen and O.B. Akan (2011), “Reliability and Congestion Control in Cognitive Radio Sensor Networks,” Ad Hoc Networks J., vol. 9, no. 7, pp. 1154-1164, Elsevier.
[3] G. Holland and N.H. Vaidya (1999), “Analysis of TCP Performance over Mobile Ad Hoc Networks,” Proc. ACM MobiCom, pp. 219-230.
[4] I.F. Akyildiz, W.Y. Lee, and K. Chowdhury (2009), “CRAHNs: Cognitive Radio Ad Hoc Networks,” Ad Hoc Networks J., vol. 7, no. 2, pp. 810-836, Elsevier.
[5] J. Liu and S. Singh (2001), “ATCP: TCP for Mobile Ad Hoc Networks,” IEEE J. Selected Areas of Comm., vol. 19, no. 7, pp. 1300-1315.
[6] K. Chen, Y. Xue, and K. Nahrstedt (2003), “On Setting TCP’s Congestion Window Limit in Mobile Ad Hoc Networks,” Proc. IEEE Int’l Conf. Comm. (ICC), pp. 1080-1084.
[7] K.R. Chowdhury, M. Di Felice, and I.F. Akyildiz (2009), “TP-CRAHN: A Transport Protocol for Cognitive Radio Ad Hoc Networks,” Proc. IEEE INFOCOM, pp. 2482-2491.
[8] L. Brakmo and L. Peterson (1995), “TCP Vegas: End to End Congestion Avoidance on a Global Internet,” IEEE J. Selected Areas in Comm. vol. 13, no. 8, pp. 1465-1480.
[9] M. Di Felice, K. Chowdhury, W. Kim, A. Kassler, and L. Bononi (2011), “End-to-End Protocols for Cognitive Radio Ad Hoc Networks: An Evaluation Study,” Performance Evaluation, vol. 68, no. 9, pp. 859- 875.
[10] M. Di Felice, K.R. Chowdhury, and L. Bononi (2009), “Modeling and Performance Evaluation of Transmission Control Protocol over Cognitive Radio Ad Hoc Networks,” Proc. 12th ACM Int’l Conf. Modeling, Analysis and Simulation of Wireless and Mobile (MSWIM ’09), pp. 4-12.
[11] S. Floyd and T. Henderson (1999), The NewReno Modification to TCP’s Fast Recovery Algorithm, IETF RFC 2582.
[12] S.M. Kay (1993), Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice -Hall.
[13] T. Dyer and R. Boppana (2001), “A Comparision of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks,” Proc. ACM MobiHoc, pp. 256-266.
[14] W.Y. Lee and I.F. Akyildiz (2008), “Optimal Spectrum Sensing Framework for Cognitive Radio Networks,” IEEE Trans. Wireless Comm., vol. 7, no. 10, pp. 3845-3857.
[15] Y.R. Kondareddy and P. Agrawal (2009), “Effect of Dynamic Spectrum Access on Transport Control Protocol Performance,” Proc. IEEE GlobeCom, pp. 1-6.