Transport Control Protocol for Cognitive Radio Ad Hoc Networks

Ritu Chauhan
Department of Electronics & Communication Engineering,
Technocrats Institute of Technology, Bhopal
India

Ashish Manusmare
Department of Electronics & Communication Engineering,
Technocrats Institute of Technology, Bhopal
India

ABSTRACT

The tremendous increase in wireless communication networks in the last few decades has lead to spectrum scarcity, which has already set the needs for efficient spectrum utilization. The juvenile and up-coming field of Cognitive radio (CR) networks is the promises to alleviate the spectrum scarcity problem, by allow devices to share the spectrum with secondary users. This improves the inefficient spectrum utilization in the bands reserved for the licensed users. This paper work investigates the use of proposed TCP-friendly protocol for variation in spectrum availability with time and periodic spectrum sensing done by the CR users. The proposed protocol employs the secondary nodes in not-fully blocked mode, to achieve significant improvements in classical TCP in a CR ad hoc network environment. In this paper we present a multipath routing protocol, that uses on demand distance vector routing, called Ad hoc On-demand Multipath Distance Vector (AOMDV) Routing Protocol. We analyze its performance using Network simulator (NS-2) software tool for the simulation purpose to give effective result of cognitive radio ad hoc networks.

Keywords

TCP, UDP, Cognitive Radio Network (CRN), spectrum sensing, AOMDV

1. INTRODUCTION

Imagine if a smart radio, it could learn about services available in the locally available wireless networks. And further could intermingle with these networks in their favored protocols, so that it can find the right wireless network to tune itself to or for a video download or a file transfer without a drop. Furthermore, it could use the frequencies without creating or say minimizing the interference with existing radio communication systems. The up-and-coming field of Cognitive Radio (CR) promises to make such a smart radio communication system a reality. A cognitive radio does can be termed as a transceiver that automatically detects available channels and accordingly adjusts its transmission or reception parameters so as to allow increased number of users to concurrently use the same wireless spectrum [1].

The erudition possible in a software-defined radio (SDR) has actually turned the possibility of Cognitive Radio systems into reality. SDRs today have already reached a level where it can credibly carry out favorable tasks that help the user, the network, and further minimize spectral congestion. Radio communication systems today are already demonstrating one or more of these capabilities in limited ways. SDRs today provide software control of a variety of modulation techniques, transmission security and even wideband and narrowband operation. A cognitive radio adds both a sensing and an adaptation element to the software defined and software radios. Cognitive Radios embodies four new capabilities in SDR to help enable dynamic use of the spectrum: flexibility, agility, RF sensing, and networking [2].

- Flexibility is the ability to change the waveform and the configuration of a device.
- Agility is the ability to change the spectral band in which a device will operate. Combining both agility and flexibility is the ultimate in “adaptive” radios because the radio can use different waveforms in different bands.
- Sensing is the ability to observe the state of the system, which includes the radio and, more importantly, the environment. It allows a radio to be self-aware, and thus is necessary if a device is to change in operation due to location, state, condition, or RF environment.
- Networking is the ability to communicate between multiple nodes and thus facilitate combining the sensing and control capacity of those nodes. Networking enables group-wise interactions between radios, which can be used for measurements that can provide a better understanding of the environment.

The above added features in Cognitive Radio (CR) network achieve opportunistic assignment of vacant portions of the spectrum to secondary users. Figure 1 below shows the four spectrum management functions in a cognitive radio cycle.

- Spectrum Sensing: A CR user should continuously observe the available spectrum bands, capture their information, and then detect spectrum holes.
- Spectrum Decision: Once the available spectrums are identified, it is essential that the CR users select the best available band according to their QoS (Quality of Service) requirements. The local spectrum decisions undertaken by a node strongly influence the end-to-end performance.
- Spectrum Sharing: The transmissions of CR users must be harmonized with spectrum sharing functionality to prevent multiple users from colliding in overlapping portions of the spectrum. Spectrum sharing provides efficient channel and power allocations to avoid interference caused to the primary network. It also adds an intelligent packet scheduling scheme enabled by a spectrum-aware link layer along with spectrum sensing [3].

- Spectrum Mobility: If the specific portion of the spectrum in use is required by a Primary User, the Secondary User should be switched to another unoccupied portion of the spectrum. This necessitates efficient spectrum handoff and reliable end-to-end connection management schemes [4].

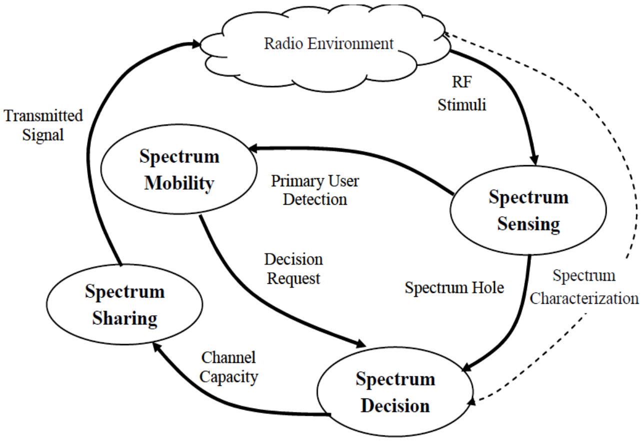

Fig 1: The Cognitive Radio Cycle

These spectrum management functions demand novel design techniques spanning virtual to all layers of the of the protocol stack on a single device. Already existing link, transport and network layer protocols designed for standard fixed bandwidth ad hoc networks cannot optimally exploit available radio resources and simultaneously provide all the communication services required for CR ad hoc network. Hence a lot of research is carried out on network and transport layer protocol design to achieve the requirements for cognitive radio ad hoc networks. This paper work evaluates the use of modified protocol referred to as TCP CRAHN on transport layer, in conjunction with AOMDV on network layer in a CR ad hoc network environment.

2. LITERATURE SURVEY

TCP works on the principle of conservation of packets by using the acknowledgements to clock outgoing packets. It further maintains a congestion window to reflect the network capacity. However there are certain issues with standard TCP. One major being circularity i.e., as a connection first starts up causes to have acknowledgements one needs to have data in the network and to put data in the network it needs acknowledgements [4]. Standard or classical TCP thus performs poorly over wireless links due to the additional packet losses caused by node mobility or bad channel conditions. Hence in the recent past many different modified versions of transport protocols have been proposed for wireless ad hoc networks, employing different approaches such as cross-layer or layered approach.

In a CRN environment the Standard TCP offers poor throughput as it treats lost or delayed acknowledgments as congestion. To overcome the problems of standard TCP Tahoe was proposed [5]. It overcomes the circularity problem by employing a slow-start procedure whenever a TCP connection starts or re-starts after a packet loss. Here the sender set the congestion window to 1 and then for each acknowledgment received it increases the Congestion Window by 1, so the number of packets sent in every round trip time (RTT) increase exponentially until there is a loss of a packet which is a sign of congestion [6].

In 1990, V. Jacobson proposed TCP Reno which worked on similar [7] basic principle of Tahoe. It retains ‘slow starts’ and coarse grain re-transmit timer principle as in TCP Tahoe and further adds some intelligence over it so as to detect packet loss earlier and preventing pipeline to be emptied every time for a packet loss. But in case of multiple packet loss condition in one window then RENO doesn’t perform too well and its performance is similar to that of Tahoe [8]. TCP New-Reno is a slight modification over TCP-Reno. It introduced a progressively fast recovery algorithm that could easily recover from multiple losses in a single window, thus avoiding many of the retransmission timeout. It is much more efficient in the event of multiple packet losses [9].

TCP Vegas is another modification of TCP Reno, which extends on the re-transmission mechanism of Reno [10]. Further it is different in congestion avoidance, as it determines congestion by a decrease in sending rate. In [11] an improvement in TCP Vegas is proposed with better BW estimation scheme. The proposed method in comparison to TCP Reno it has better delay bias. Further TCP Vegas has gives efficient use of network resources.

Jian Liu and Suresh Singh in their work in [12], have a thin layer between Internet protocol and standard TCP is implemented so as to correct this problem and maintains high end-to-end TCP throughput. Their proposed ATCP algorithm treats loss due to packet loss and due to congestion differently. It also recomputes the congestion window after every new route recomputation. They propose to improve TCPs throughput by a factor of 2–3. In [13] modification in the congestion window adaptation in the classical TCP based on the available bandwidth estimation is proposed. This approach thwarts erroneously plummeting of transmission rate during spectrum sensing. Conversely, the work fails to consider the impact of network mobility, sensing activity and spectrum switching durations on the congestion control functions.

In [14] TCP CRAHN: a protocol for Cognitive Radio Ad Hoc Networks is proposed. The proposed method incorporates spectrum awareness in the classic TCP protocol. This is achieved by a combination of explicit feedback from intermediate and destination nodes. Here the classical TCP rate control algorithm at the source is made to closely interact with the physical layer channel information, the link layer functions of spectrum sensing and buffer management, and a predictive mobility framework that is developed at the network layer. The work claims to achieve significant improvements in the use of proposed TCP CRAHN as transport layer protocol for CR ad hoc networks.

3. TCP CRAHN

TCP CRAHN is a window-based, spectrum-aware protocol for CR ad-hoc networks, that distinguishes between the different spectrum specific conditions in order to undertake state-dependent recovery actions [14]. TCP CRAHN comprises of 6 states addressing a particular CR network condition namely, Connection establishment, Normal, Spectrum sensing, Spectrum change, Mobility predicted, and Route failure (as shown in Figure 2). Figure 2 below shows the States of TCP CRAHN.

3.1 Connection Establishment

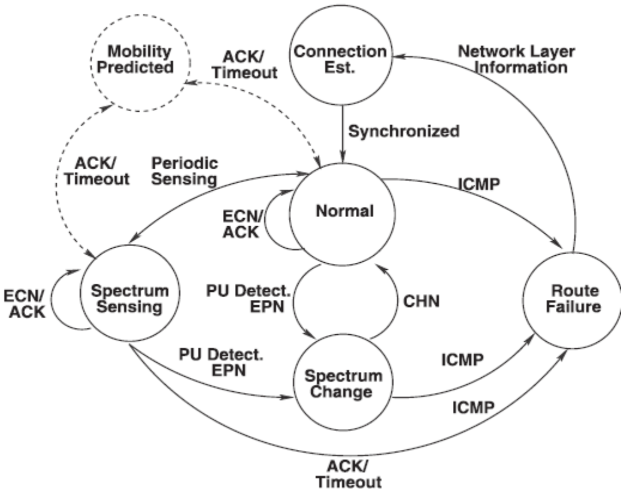

Fig 2: States of TCP CRAHN

To obtain the sensing schedules of the nodes in the routing path, TCP CRAHN modifies the three-way handshake in TCP NewReno. First, the source sends SYN packet to the destination. An intermediate node in the routing path appends its ID, a timestamp and other sensing related information to the SYN packet. The receiver in response to the SYN packet sends a SYN-ACK message back to the source. The sensing information collected for each intermediate node is saved over the SYN-ACK. This process helps the source to know when a node undertakes spectrum sensing and its duration. Finally an ACK is sent by the source to the destination to complete the 3-way handshake process, constituting the connection establishment process. In TCP CRAHN, these collected sensing times from the nodes are dynamically updated.

3.2 Normal State

This is the default state and is similar to functioning of the classical TCP NewReno. TCP CRAHN enters this state when there is no node in the path engaged in spectrum sensing or when there are no connection breaks or there are no signaled route failures. The differences between TCP CRAHN in the normal state and the classical TCP are as follows. In this stage, TCP CRAHN differs slightly from the classical TCP, in that the congestion event is signaled through an Explicit Feedback Congestion Notification (ECN) generated by the affected node and the feedback through ACK i.e., the ECN and ACK regulates the congestion window.

3.3 Spectrum Sensing State

To meet the high throughput demand, TCP CRAHN acclimatizes to spectrum sensing through flow control and regulating the sensing time. The flow control helps to prevent buffer overflow for the intermediate nodes during sensing. As in case of no Primary User activity on a given channel and comparatively large sensing times degrade the end-to-end throughput of the channel. Thus TCP CRAHN conservatively reduces sensing time for the nodes that see limited PU activity. To identify a node for adjusting its sensing time, TCP CRAHN ranks the nodes in the path based on the number of times the operational channel was changed due to Primary User activity.

3.4 Spectrum Change State

This where the TCP CRAHN scales its congestion window rapidly accounting factors such as contention delays and channel errors at the link layer, so that the available spectrum resource is most efficiently utilized. If PU activity is reported to the source through an Explicit Pause Notification (EPN), Spectrum Change state is entered and reverts back to the usual operation on receiving the new channel information through the Channel (CHN) message. It enters into the Route Failure state on the receipt of an ICMP (through Internet Control Message Protocol) message signifying route outages.

3.5 Mobility Predicted State

TCP CRAHN uses a mobility prediction framework based on Kalman filter-based estimation to address delayed route failure notification problem [14]. This is a virtual state, in which the congestion window is restricted to the spectrum sensing threshold. Finally either the Normal or the Spectrum Sensing state is continued.

3.6 Route Failure State

This state is entered when an intermediate node sends a destination unreachable message in the form of an ICMP packet and it takes precedence over all the others states. The ICMP is achieved when the next hop node is not reachable or there is no ongoing spectrum sensing based on the last known schedule or no EPN message is received at node signaling a temporary path disconnection due to PU activity. At this stage, the source terminates the current cycle and a fresh TCP connection needs to be formed over the new route.

4. AD-HOC ON-DEMAND MULTIPATH DISTANCE VECTOR (AOMDV)

AOMDV routing protocol is an extension to the AODV (As-hoc On-demand Vector) protocol for computing multiple loop-free and link disjoint paths [15]. AOMDV finds routes on demand by using a route discovery procedure. In AOMDV, RREQ propagation from the source towards the destination ascertains multiple reverse paths both at intermediate nodes and destination. The routing entries for each destination contain a list of the next-hops along with the corresponding hop counts. A node maintains the advertised hop count for each destination, which is defined as the maximum hop count for all the paths. Loop freedom is assured for a node by accepting alternate paths to destination if it has a less hop count than the advertised hop count for that destination. Multiple RREPs traverse these reverse paths back to form multiple forward paths to the destination at the source and intermediate nodes. The trajectories of each RREP may intersect at an intermediate node, however to ensure link disjointness, each RREP takes a different reverse path to source. AOMDV has three novel features: 1) it has minimum inter-nodal coordination overheads; 2) it guarantees disjoint alternate routes via distributed computation and 3) it computes alternate paths with minimal additional overhead by exploiting already available alternate path routing information. There are three phases of the AOMDV protocol. The first phase is the Route Request, second is the Route Reply and the third phase is the Route Maintenance phase.

- Route Request

The protocol propagates RREQ from source towards the destination. The figure1 will show the working of AOMDV, which allows multiple RREQ to propagate. The node S has to set a path to the destination node D. So node S in AOMDV broadcasts multiple requests to its neighboring nodes 1 and 2. This means that request with same sequence numbers are sent to the destination node. They further broadcast the request to the other neighboring nodes, which are further sent to the destination node D.

- Route Reply

In this session, protocol establishes multiple reverse paths both at intermediate nodes as well as destination. Multiple RREPs traverse these reverse paths back to form multiple forward paths to the destination at the source and intermediate nodes. If the intermediate nodes have the route defined for the destination then they send the RREP to the source node S. The protocol is designed to keep track of multiple routes where the routing entries for each destination contain a list of next hops together with the corresponding hop counts. All the hop counts have the same sequence number then the path with the minimum hop count is selected and all the other paths are discarded. The protocol computes multiple loop-free and link disjoint paths. Loop-freedom is guaranteed by using a notion of "advertised hop count". Each duplicate route advertisement received by a node defines an alternative path to the destination.

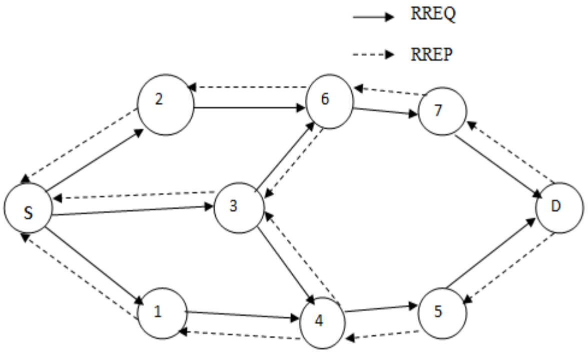

Fig 3: Working of AOMDV

- Route Maintenance Phase

The third phase is the Route Maintenance Phase. In this phase, if the intermediate nodes are not able to receive a response of the HELLO message then they broadcast a Route Error message. After receiving this message all the nodes that use the particular route to reach the destination make this particular route as infinity and inform the source node to run a fresh route discovery.

Routing table entry structure of AOMDV as shown in Table 1 below:

Table 1. AOMDV Routing table.

Destination
Sequence number
Advertised _ hopcount
Expiration_ timeout
route _list {(nexthop1 , hopcount1), (nexthop2, hopcount2)…..}

5. RESULTS

The proposed system employs the use of TCP CRAHN along with AOMDV as the network layer protocol. The proposed system is implemented using NS-2 simulator for a CR ad-hoc environment. Figure 5 shows the Delay Graph of the system. The delay of the system is negligible and the maximum time delay is within range of 2 ms. Figure 6 shows the energy graph for the system. It can be seen that the energy of the system remains nearly constant throughout the period. Figure 7 shows the throughput of the system. The throughput of the system is almost 100% except for a brief moment where there is a decrease, however this can be neglected.

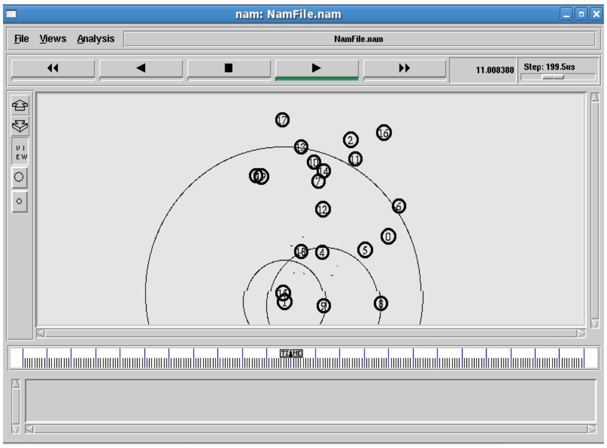

Fig 4: States of TCP CRAHN

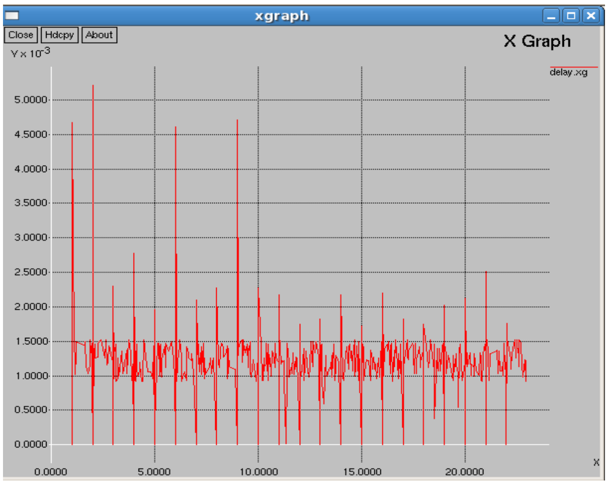

Fig 5: Delay Graph

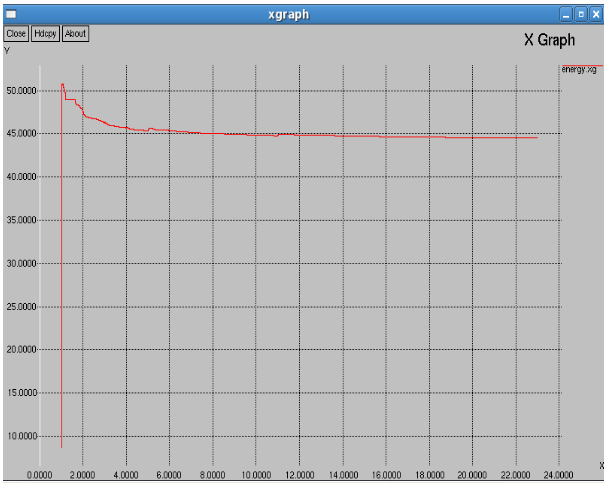

Fig 6: Energy Graph

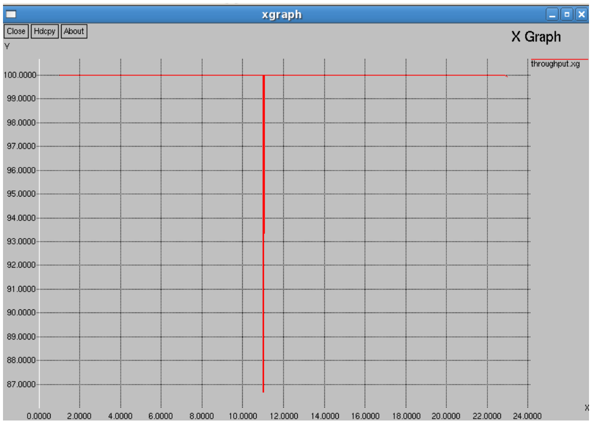

Fig 7: Throughput Graph

6. CONCLUSION

The proposed system employs the use of TCP CRAHN for Cognitive Radio Ad-hoc network environment. The implementation of the TCP CRAHN protocol at the transport layer and AOMDV at network layer gives tremendous results in comparison to classic TCP implementation. The proposed system has a delay of less that 2ms and throughput of nearly 100%. Usually CRAHN is implemented in UDP environment which gives lower values of Throughput and Higher delays due to packet losses, but as we have implemented TCP based CRAHN we are able to reduce the delay and improve the throughput at the same time without increasing the energy consumption in the network.

7. REFERENCES

[1] R.Brodersen, A. Wolisz, D. Cabric, S. Mishra and D. Willkomm, "Corvus: A cognitive radio approach for usage of virtual unlicensed spectrum", UC Berkeley White Paper, 2004.

[2] Kolodzy, P, "Dynamic Spectrum Policies: Promises and Challenges," CommLaw Conspectus, 2004.

[3] Akyildiz, I. F., Lee, W. Y., and Chowdhury, K. R., "CRAHNs: Cognitive Radio Ad Hoc Networks," Ad Hoc Networks (Elsevier) Journal, to appear, vol. 7, July 2009.

[4] Lee SJ, Gerla M, "AODV-BR: backup routing in ad hoc networks", In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2000.

[16]

[5] V.Jacobson. "Congestion Avoidance and Control", SIGCOMM Symposium no Cummunication Architecture and protocols.

[6] V.Jacobson "Modified TCP Congestion Control and Avoidance Alogrithms".Technical Report 30,Apr 1990.

[7] V. Jacobson. Berkeley TCP evolution from 4.3-Tahoe to 4.3 Reno. In Proc. of the 18th IETF, Vancouver, Canada, August 1990.

[8] N. Parvez, A. Mahanti, and C. Williamson, "An Analytic Throughput Model for TCP NewReno," IEEE/ACM Trans. Networking, vol. 18, no. 2, pp. 448-461, Apr. 2010.

[9] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782, April 2004.

[10] L.S.Brakmo, L.L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communication, vol. 13, October 1995.

[11] Jeonghoom Mo, Richard J. La, Venkat Anantharam and Jean Walrand, "Analysis and comparison of TCP Reno and Vegas", University of California at Berkeley, IEEE, 0-7803-5417-6, 1999.

[12] Jian Liu and Suresh Singh, "ATCP: TCP for Mobile Ad Hoc Networks", IEEE Journal on selected areas in communications, Vol. 19, No. 7, July 2001.

[13] D. Sarkar and H. Narayan, "Transport Layer Protocols for Cognitive Networks", INFOCOM IEEE Conference on Computer Communications Workshops , 2010.

[14] Kaushik R. Chowdhury, Marco Di Felice, and Ian F. Akyildiz, "TCP CRAHN: A Transport Control Protocol for Cognitive Radio Ad Hoc Networks", IEEE Transactions on Mobile Computing, Vol. 12, No. 4, April 2013.

[15] Royer E.M. Perkins C.E. Ad-hoc on-demand distance vector routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, p.90, 1999.