This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

TP-CRAHN: A Transport Protocol for Cognitive Radio Ad-hoc Networks

Kaushik R. Chowdhury
Broadband Wireless Networking Lab
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: kaushikc@ece.gatech.edu

Marco Di Felice[1]
Department of Computer Science
University of Bologna
Bologna, Italy
Email: difelice@cs.unibo.it

Ian F. Akyildiz
Broadband Wireless Networking Lab
Georgia Institute of Technology
Atlanta, GA 30332, USA
Email: ian@ece.gatech.edu

***Abstract*—Existing research in transport protocols for wireless ad-hoc networks has focused on reliable end-to-end packet delivery under uncertain channel conditions, route failures due to node mobility and link congestion. In a cognitive radio (CR) environment, there are several key challenges that must be addressed apart from the above concerns. The intermittent spectrum sensing undertaken by the CR users, the activity of the licensed users of the spectrum, large-scale bandwidth variation based on spectrum availability, and the channel switching process need to be considered in the transport protocol design. In this paper, a window-based Transport Protocol for CR Ad-Hoc Networks, TP-CRAHN, is proposed that distinguishes each of these events by a combination of explicit feedback from the intermediate nodes and the destination. This is achieved by adapting the classical TCP rate control algorithm running at the source 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. To the best of our knowledge, this is the first work on the transport layer to specifically address the concerns of the CR ad-hoc networks and our approach is thoroughly validated by simulation experiments.**

I. INTRODUCTION

A wireless ad-hoc network is comprised of nodes that forward data packets in a decentralized manner over multiple hops to the destination. The increasing deployment of such networks in military applications, vehicular surveillance, disaster relief, commercial messaging, among others have led to a growing congestion and spectrum scarcity in the unlicensed $2.4\,\mathrm{GHz}$ ISM band. The emerging field of cognitive radio (CR) networks attempts to alleviate the problem of spectrum shortage in the ISM band by opportunistically transmitting on other vacant portions of the spectrum, such as frequencies licensed for television broadcast and public services [1]. 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 CR environment. 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 [2]. In this paper, we propose a window-based, TCP-like spectrum-aware transport layer protocol for CR ad-hoc networks, called TP-CRAHN, that distinguishes between these different conditions in order to undertake state-dependent recovery actions.

As the transport protocol usually runs at the end nodes (source and destination), it has limited knowledge of the conditions of the intermediate nodes. Unknown to the source, the route may be disconnected due to node mobility. Also, packet losses may be wrongly attributed to network congestion rather than bad channel conditions at the link layer. Classical TCP suffers from some of the above issues and efforts have been made to address them for wireless scenarios in [9] [11]. However, these protocols for classical wireless ad-hoc networks do not consider the cases that may arise in CR ad-hoc networks. As an example, in a classical wireless ad-hoc network, packets may incur a longer round trip time (RTT) owing to network congestion or due to a temporary route outage. In CR ad-hoc networks, a similar effect on the packet RTT may be caused if an intermediate node on the route is engaged in spectrum sensing and hence, unable to forward packets. Also, the sudden appearance of a primary user may force the CR nodes in its vicinity to limit their transmission leading to an increase in the RTT. In such cases, the network is partitioned until a new channel is identified and coordinated with the nodes on the path. The duration of the periodic spectrum sensing decides, in part, the end-to-end performance - a shorter sensing time may result in higher throughput but may affect the transport layer severely if a PU is misdetected. While several works have focussed on spectrum sensing algorithms in the last few years [1], the integration of the channel information collected at the nodes and the performance study of these approaches from the viewpoint of an end-to-end protocol remains an open challenge.

TCP, in general, is a well researched area and several theoretical models exist that explain and predict its behavior in wireless networks [12]. It is also implemented at the transport layer for commercially available devices. In addition, the ad-hoc network may ferry user traffic to and from the external infrastructure network, receiving configuration commands from remote stations. TCP is the de-facto standard in the wired world and a measure of compatibility is useful from the network management perspective. Hence, the goal of TP-CRAHN is to retain the window-based approach of the

[1]This work was conducted during his stay at the BWN lab in 2008.

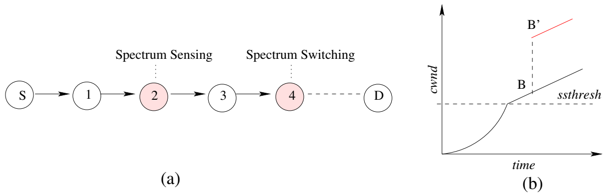


Fig. 1. A multi-hop CR ad-hoc network (a) and the forced *cwnd* scaling (b)

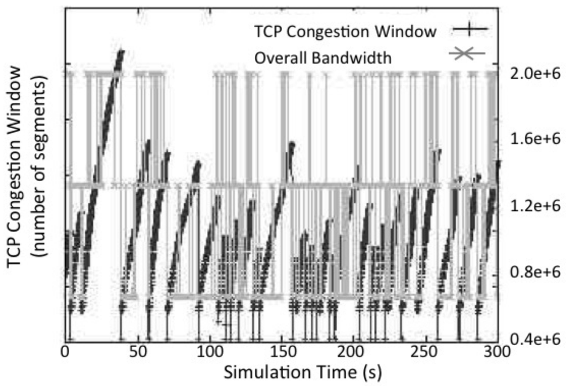


Fig. 2. Effect of changing channel bandwidth on *cwnd*

classical TCP, and at the same time introduce novel changes that allow its applicability in CR ad-hoc networks.

The rest of this paper is organized as follows. In Section II, we motivate the need of a new transport protocol for CR networks. The network architecture is given in Section III. In Section IV, we describe our transport layer protocol in detail. We undertake a thorough performance evaluation in Section V, and finally, Section VI concludes our work.

II. Motivation

In this section, we discuss the problems with the existing implementations of transport protocols based on TCP in CR ad-hoc networks, in which, nodes are equipped with a single radio transceiver. The features of the CR network that we study are: (i) spectrum sensing (ii) effect of primary user (PU) activity, and (iii) spectrum change.

A. Spectrum Sensing State

CR users periodically monitor the current channel over a pre-decided sensing duration for the occurrence of PUs before using it for transmission. During this interval, the nodes are not actively involved in transmitting data packets and the multi-hop network is virtually disconnected at the node that is engaged in spectrum sensing. As an example, for the multi-hop network shown in Figure 1 (a), let S and D are the source and destination nodes, respectively. When node 2 senses the spectrum, the path forms two connected segments, from $(S-1)$ and $(3-D)$ giving the following cases:

- As the acknowledgments (ACKs) from the destination can no longer reach the source, the RTT of the transmitted segments increases and this may trigger a retransmission timeout (RTO), even in the absence of true congestion.
- If the source S does not limit its transmission rate for the sensing duration, packets queue up at node 1 and may soon cause a buffer overflow.

Apart from the above considerations, the sensing interval also plays a critical role in deciding the optimal end-to-end throughput, as pointed out in [10]. If the sensing time is longer, the CR user monitors the channel rather than forward the data packets while a very short sensing interval increases the risk of interfering with the activity of a PU [8]. Thus, the transport layer has to balance this tradeoff so that the throughput of the connection is maintained at the desired level and the PU interference is minimized.

B. Effect of PU Activity

On detecting the presence of a PU, either during spectrum sensing or an ongoing data transfer, the CR users cease their operation on the affected channel and search for a different vacant portion of the spectrum. While the spectrum sensing on the current channel is periodic and has a well defined interval, the time taken to (i) search for a set of available channels on different spectrum bands, and (ii) coordinate with the next hop neighbors to find a mutually acceptable channel in this set, is of an uncertain duration. The path to the destination is disconnected until the new channel is successfully found. Moreover, unlike spectrum sensing, this duration is not known to the source in advance. Thus, the transport protocol needs to differentiate this state from other causes of route disconnections with the help of an explicit feedback from the nodes affected by the PU activity.

The spectrum sensing time and the delay induced by PU arrivals also influence the way in which notifications about route failures should be used in the CR ad-hoc transport protocols. In classical ad-hoc networks, ATCP [9] and TCP-EFLN [5] react to the route disruption *after* it happens by an explicit notification in the form of the Internet Control Message Protocol (ICMP) message at the IP layer. In Figure 1 (a), if the node 4 generates the ICMP message, it must traverse through the intermediate hops before the source S is reached. However, this packet may suffer a large wait period as the nodes in the path $(S-3)$ complete their sensing schedules or negotiate a new channel on detection of a PU. A classical scheme for for reducing the packet losses by routing layer feedback is proposed in [13]. However, this method uses cached routes and does not involve new route discovery. For CRAHNs, the changing spectrum environment may not guarantee the feasibility of the cached route. Thus, we believe that a predictive framework is needed so that that source can restrict its sending rate in advance of the route failure and limit the number of transmitted packets, when the notification of failure is actually received.

C. Spectrum Change State

A key concern in CR networks is the efficient utilization of the spectrum resource, as the opportunity for transmission in the licensed bands is available for a limited time. The licensed channels may have a large variation in bandwidth, especially

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

as nodes switch from one spectrum band to the other. In Figure 2, we study through simulation how classical TCP increases the *cwnd* as it probes for the additional bandwidth available on a single link. There are three different channel bandwidths possible- $2/3\,\text{Mbps}$, $4/3\,\text{Mbps}$, and $2\,\text{Mbps}$. The vertical bars denote the bandwidth available to the node and at any given time, this is the upper limit that can be utilized by the TCP connection. This gives three distinct levels of bandwidth availability with time. On each channel, the PU is modeled as a Poisson arrival, with an "*on*" time ($\alpha = 4$s) and "*off*" time ($\beta = 5$s). When the PU arrives, the CR user switches to a different channel, and consequently TCP must adjust to the new available bandwidth. From the figure, we observe that the *cwnd* is unable to correctly track the available bandwidth. Moreover, the spectrum opportunity is often lost before the *cwnd* has increased to half the segments that may be supported on the new channel. A similar conclusion is drawn in [10], where TCP cannot effectively adapt to brief reductions in capacity, if the end-to-end delay is large. We believe that the *cwnd* in TCP must be scaled appropriately to meet the new channel conditions, as shown in the transition from the operating point B to the point B' in Figure 1 (b). Estimating this new operating point is a challenge and link layer metrics, that determine the effective bandwidth, must also be considered apart from the raw bandwidth. Bandwidth estimation techniques have been proposed in [3] [7], that do not require information from the intermediate nodes, but also do not respond immediately to the available spectrum.

III. Network Architecture

The nodes forming the CR ad-hoc network have a single radio transceiver, that can be tuned to any channel in the licensed spectrum. We assume ψ spectrum bands are present with $n(x)$ channels in a given band x. The channels of this spectrum band are denoted by $\xi_p^x,\ p = 1, \ldots, n(x)$. In general, two channels in different spectrum bands may have dissimilar raw channel bandwidth, i.e. $\xi_p^x \neq \xi_q^y$. In addition, we assume the statistical knowledge of the PU arrival (α) and departure rate (β) for each channel are known, so that an initial estimate of the channel sensing time can be calculated.

We use 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 during data transfer. We also use a priority queue, Q_p at the MAC layer for the TP-CRAHN control packets, which may also be drawn from intermediate positions in Q_p.

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, we assume that this set of channels is identified through spectrum sensing, undertaken during the *backoff* interval following a packet transmission or reception at the link layer. On the current operational channel, however, it is important to have an accurate idea of the PU activity. For this, we do not rely on probabilistic sensing times. Rather, nodes sense their current channel for the sensing time t^s at regular intervals at the cost of continued network connection [8]. We now describe our protocol and discuss its operation in a CR network.

IV. TP-CRAHN: A Transport Protocol for CR Ad-Hoc Networks

TP-CRAHN comprises of the following 6 states, as shown by the state diagram in Figure 3. They are (i) Connection Establishment, (ii) Normal, (iii) Spectrum Sensing, (iv) Spectrum Change, (v) Mobility Predicted, and (vi) Route Failure. Each of these states addresses a particular CR network condition and we describe them in detail as follows.

A. Connection Establishment

TP-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: (i) its ID, (ii) a timestamp, and (iii) the tuple $\{t_i^1, t_i^2, t_i^s\}$. Here, t_i^1 is the time left before the node starts the next round of spectrum sensing, measured from the timestamp. t_i^2 is the constant duration between two successive spectrum sensing events, and t_i^s 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 and thus, the source knows when a node in the path shall undertake spectrum sensing and its duration. The final ACK is then sent by the source to the destination completing the handshake.

We note that the calculation of the sensing time t_i^s by a node i is undertaken locally. Based on the bandwidth of the channel (W), the external signal to noise ratio (γ), and the probabilities of the on period (P_{on}) and the off period (P_{off}), a framework to calculate this time is given as follows [8],

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

Equation (1) gives the sensing time t_i^s that minimizes the probability of missed primary user detection P_f, i.e., incorrectly stating the channel is vacant when indeed there is an active PU and Q is the standard Q function. The sensing times collected from the nodes are the preliminary values which are dynamically updated by TP-CRAHN, as described in Section IV-C.

State Transitions: On successful handshake, the source and destination are synchronized and the *Normal* state is entered.

B. Normal State

The *normal* state in TP-CRAHN is the default state and resembles the classical functioning of the classical TCP newReno protocol. Our protocol enters this state when (i) no node in the path is currently engaged in spectrum sensing, (ii) there are no connection breaks due to PU arrivals, and (iii) no impending route failure is signaled. Thus, the path to the destination remains connected and ACKs sent by the latter are

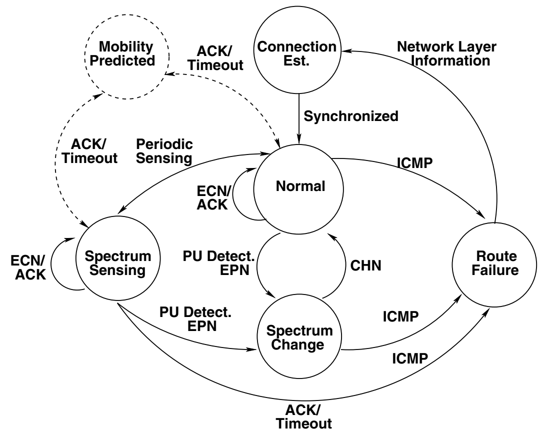


Fig. 3. Finite state machine model of our transport protocol

received at the source. The differences between TP-CRAHN in the *normal* state and the classical TCP are as follows:

1) Explicit Congestion Notification: The congestion control algorithm in classical TCP operates in two phases, namely, the *slow start* and the *congestion avoidance*. In the slow start, the congestion window *cwnd* is initially set to 1 (Maximum Size Segment or MSS) and doubled for each incoming ACK. As TCP probes for the available bandwidth, the *cwnd* increases exponentially until the threshold, *ssthresh*, is reached. It then enters the collision avoidance phase, where the *cwnd* is incremented by 1 MSS for every acknowledged packet. During network congestion, indicated by the RTO timeout events, TCP reduces the *cwnd* to 1 and the new threshold is set to the value $\frac{ssthresh}{2}$.

While the above ACK based self-clocking mechanism that increases the *cwnd* is retained in TP-CRAHN, the congestion event is signaled through an explicit feedback congestion notification (ECN) generated by the affected node. The congestion is detected at the node by comparing the current buffer usage for the given flow with a pre-decided threshold value B^f_{con}. The ECN is sent in two ways to guarantee its timely delivery. First, a packet is sent from the affected node to the source directly. In addition, the ECN is piggybacked to the destination over the data packets and then sent to the source through the ACK. This is done as the remainder of the connection from the affected node to the destination may suffer delay from a temporary disruption caused by channel sensing or switching. When an ECN is received by the source, TP-CRAHN first evaluates if it is still relevant to the network congestion state by checking the time lag from its generation at the affected node to its reception. If this time is within the time lag threshold L_{max} and no prior action has been taken for an earlier ECN from the same node, for the detected congestion event, TP-CRAHN reduces the *cwnd* to 1 and cuts the *ssthresh* by half. In our work, we set $L_{max} = 1.5 \cdot RTT$, as any further delay suggests that the path was temporarily disconnected due to a sensing or channel switching event. In either case, the transmission rate at the source is reduced, as we shall see later in the protocol description.

2) Feedback Through the ACK: The intermediate nodes of the path piggyback the following link-layer information over the data packets to the destination, which is then sent back to the source through the ACK.

- *Residual buffer space* (B^f_i)*:* Consider a node i that has B^u_i bytes currently of unoccupied buffer space. Let the number of flows passing through it be n^f_i. The fair share of the residual buffer space per flow is, $B^f_i = \frac{B^u_i}{n^f_i}$.
- *Observed link bandwidth* ($W_{i,i+1}$)*:* Each node i maintains a weighted average of the observed bandwidth on the link formed with its next hop, i.e. $\{i, i+1\}$, during the *normal* state. This is obtained from the link layer as the ratio of the acknowledged data bits to the time taken for this transfer between the nodes i and $i+1$.
- *Total link latency* ($L^T_{i,i+1}$)*:* Let $L_{i,i+1}$ be the sum of the (i) time taken by a packet of the current flow to move to the head of the queue (ii) the time for contending the access to the channel and finally, (iii) the transmission time measured at node i with respect to the next hop $i+1$. The total link latency is now defined considering the bidirectional link latencies, $L^T_{i,i+1} = L_{i,i+1} + L_{i+1,i}$.

Apart from these fields that are updated every time by the nodes, the ACK also carries the ECN notification whenever a node experiences congestion and the mobility predicted flag (MF), when a possible route outage is identified.

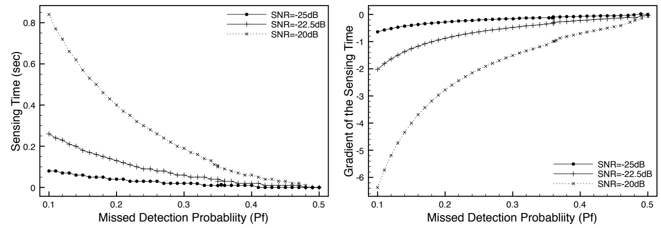
State Transitions: The ECN message and the ACKs regulate the *cwnd*. When a node in the path performs sensing, TP-CRAHN transitions into the *Spectrum Sensing* state. If PU activity is reported to the source through an Explicit Pause Notification (EPN), it enters the *Spectrum Change* state and resumes the usual operation on receiving the information about the new channel through the Channel (CHN) message. Possible route disruptions may be signaled by the information contained in the ACKs leading to the temporary *Mobility Predicted* state, where the *cwnd* is restricted to *ssthesh*. If the ICMP message is received, TP-CRAHN enters into the *Route Failure* state and stops transmission.

C. Spectrum Sensing State

We describe how TP-CRAHN adapts to spectrum sensing through (i) flow control, that prevents buffer overflow for the intermediate nodes during sensing and (ii) regulating the sensing time to meet the specified throughput demands.

1) Flow Control: When a node i undertakes spectrum sensing, the path gets virtually disconnected for a finite duration. At this time, the goal of TP-CRAHN is to adapt the flow control mechanism in TCP, so that the node $i-1$, prior to the sensing node, is not overwhelmed with incoming data packets. If another node j has an overlapping sensing schedule, TP-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 $j-1$ is used in the *ewnd* computations.

We recall that, in classical TCP, the maximum number of bytes of unacknowledged data allowed at the sender is



(a) (b)

Fig. 4. The sensing duration for increasing error probability and the gradient of the curves are shown in (a) and (b), respectively, for different SNR ranges

the minimum of the current congestion window, *cwnd*, 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. The effective window, *ewnd* at the sender is modified to include an estimate of the free buffer space, B^f_{i-1}, at the previous hop node $i-1$ as follows,

$$ewnd = \min\{cwnd, rwnd, B^f_{i-1}\} \quad (2)$$

As the packets fill up the buffer in the node $i-1$, the remaining free buffer space needs to be progressively reduced, so that the effective window can be computed from equation (2). The intermediate node, unlike the destination, does not send back the ACKs with the new advertised receive window and hence, this is estimated as follows: From the last received ACK, we know the free buffer space for the given flow, at the node $i-1$, is B^f_{i-1}. The approximate time for successfully transmitting a packet over the link $i-2, i-1$ can be calculated at the source as $L_{i-2,i-1} = \frac{L^T_{i-1,i-2}}{2}$, where $L^T_{i-1,i-2}$ is the bidirectional link latency piggybacked over the ACK. Thus, the space available B^f_{i-1} in the node $i-1$ is decremented at intervals of $L_{i-2,i-1}$, when node i is engaged in sensing. We note that while the rate of decrease of the buffer space is not exact, the node $i-1$ is oblivious to this sender-side adaptation. It can still force the source to reduce its sending rate through the congestion notification. If its buffer is reaching the overflow limit, the congestion condition will be signaled and the *cwnd* will be reduced to 1 at the source as a response.

If any of the intermediate nodes on the path from the source to the node $i-1$ detect congestion, the ECN packet is sent by them and the *cwnd*, used in equation (2), is then reduced to 1. We note that the effective window *ewnd* remains at 1, as long as the path remains disconnected, as the *cwnd* cannot be increased without the arrival of the ACK.

2) Sensing Time Regulation: In Section II-A, we stated that if there is no PU activity on a given channel, the comparatively large sensing times degrade the end-to-end throughput. To address this, TP-CRAHN conservatively reduces t^s for the nodes that see limited PU activity. This calculation is carried out at the transport layer as it is aware of the observed (τ_o) and the desired (τ_d) throughputs, respectively. The two inputs needed by our protocol are - (i) the value, δ^s_i, by which the sensing time should be decreased, and (ii) the node i at which this reduction must be undertaken:

Sensing Time Decrease: Figures 4(a) and 4(b) give the optimal sensing time and the gradient of the curve for the sensing time, respectively, in order to maintain a given missed detection probability (P_f) for different SNR ranges and a bandwidth of $2\,\text{MHz}$, as per the analytical formulation in [8]. We observe that the t^s is large when the target error probability is very low. Moreover, there is also a large fall in the t^s for a finite change in the P_f, as shown by the gradient curve (Figure 4(b)), when the P_f is small. This means that the t^s can be reduced by a greater margin in the initial stage, when it has a comparatively higher duration, without impacting the error significantly. Also this margin must itself be lowered as the value of t^s gets progressively reduced. The intuitive reasoning is as follows: A node i first sets $t^s = t^s_{max}$ corresponding to the low error probability of $P_f = 0.1$. If the number of spectrum changes that occur over time, is small in proportion to the number of total changes along the entire path, we assume that the node is situated in a region with limited PU activity. Thus, the periodic sensing time t^s_i may be reduced at this node. If the current t^s_i at the node is large, its reduction is consequently higher, as the probability of error is not affected proportionally. However, as the t^s_i value falls, the reduction gets progressively smaller until $t^s_i = t^s_{min}$ is reached, corresponding to the limiting error probability, $P_f = 0.5$. We can now formulate the steps for obtaining the new sensing duration $t^s_i(\text{new})$ from the old value t^s_i as follows,

$$\triangle_t = \frac{t^s_{max}}{2} \quad (3)$$

$$\gamma_{t^s_i} = \frac{\mathrm{d}t^s}{\mathrm{d}P_f}\Big|_{t^s=t^s_i,\ P_f=func(t^s_i)} \quad (4)$$

$$\gamma_{t^s_{max}} = \frac{\mathrm{d}t^s}{\mathrm{d}P_f}\Big|_{t^s=t^s_{max},\ P_f=0.1} \quad (5)$$

$$\delta^s_i = -\frac{\gamma_{t^s_i}}{\gamma_{t^s_{max}}} \cdot \triangle_t \quad (6)$$

$$t^s_i(\text{new}) = t^s_i - \delta^s_i \quad (7)$$

The default decrement value of the sensing time, $\triangle_t$, is taken as half of the maximum value, t^s_{max}, needed to maintain the error probability at 0.1, as shown in (3). This is later scaled by a factor in the range $[0, 1]$ to get the true decrement δ^s_i. In (4), we calculate the value of the gradient $\gamma_{t^s_i}$ to the sensing curve at the current sensing duration t^s_i, at node i. The corresponding value of P_f is obtained from the current t^s_i from Figure 4(a), which is in turn, a numerical plot of equation (1). The maximum gradient of the sensing curve $\gamma_{t^s_{max}}$ is given in (5) and is computed at $t^s_i = t^s_{max}$. The normalized gradient, $\frac{\gamma_{t^s_i}}{\gamma_{t^s_{max}}}$, at the current tuple given by $\{t^s_i, P_f\}$ is used as the scaling factor to give the true decrement δ^s_i in (6). Finally, the sensing time is adjusted to the new value $t^s_i(\text{new})$ in (7).

When successive missed detection events occur, the node increases the sensing duration in the steps $\{\frac{1}{2} \cdot t^s_{max}, \frac{3}{4} \cdot$

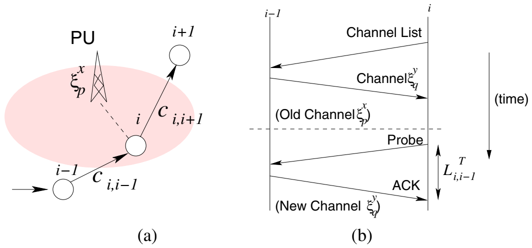


Fig. 5. The PU interference scenario (a) and the link layer total delay estimation (b) are shown

$t^s_{max}, t^s_{max}\}$, in that order. Whenever the sensing time is changed, the node sends back the new value to the sources of the flows passing through it by piggybacking over the ACK.

Node Selection: In order to identify a specific node i for adjusting the sensing time, TP-CRAHN ranks the nodes in the path based on the number of times the operational channel was changed due to PU activity. It keeps a count of the CHN messages sent by each node of the path, which reveals the number of times the connection was paused while a new-channel was being coordinated. Intuitively, the node that generated the highest proportion of the CHN message also experienced the maximum number of PU detection events and thus, must be located in a region of frequent PU activity. Such a node needs to retain a higher sensing duration.

Let the total number of times the spectrum change occurs at a given node i, and that considering all the nodes of the path be given by η_i and η_T, respectively. We define the probability of the node i being susceptible to PU activity, S_i as the ratio $S_i = \frac{\eta_i}{\eta_T}$. Let the set of n nodes along the route have their PU activity susceptibility given by the set $\mathbf{S} = \{S_1, \ldots, S_n\}$. Recalling that τ_d and τ_o are the desired and observed throughputs, the source executes the following algorithm to determine the node q and adjust its sensing time to the new value $t^s_q(\text{new})$:

```
PROCEDURE:Sense-Adjust
Input: τ_d, τ_o, S
Output: q, t^s_q(new)
if τ_d > τ_o then
    q = arg_i min{S_i}, i = 1,...,n
    if t^s_q(old) > t^s_min & S_i < S_max then
        t^s_q(new) = t^s_q(old) − δ^s_i
    end
end
```

We explain the algorithm as follows: If the desired throughput (τ_d) is greater than the observed throughput (τ_o), then TP-CRAHN finds the node q with the minimum PU susceptibility, $S_i,\ i = 1, \ldots, n$. Two conditions are checked for the node q- (i) the current sensing time at the node t^s_q must be greater than the minimum allowed sensing time t^s_{min} and (ii), the PU susceptibility must be below the limiting threshold S_{max}, considered as 0.3 in our work. This ensures that only nodes that have are relatively undisturbed by PU activity over time are chosen for reduction of the sensing duration by the value δ^s_i, as described in equation (7). This new value of the sensing interval is sent to the intermediate node by the source.

State Transitions: On receiving the EPN message, the *Sensing* state is interrupted and our protocol immediately transitions to the *Spectrum Change* state, or else, it reverts back to the *Normal* state on the completion of the sensing duration. The transitions to the *Mobility Predicted* and the *Route Failure* state are similar to the description of the *Normal* state.

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 TP-CRAHN scales its *cwnd* rapidly, say from point B to a different value B', in Figure 1 (b), accounting for these factors, so that the available spectrum resource is most efficiently utilized.

Consider three nodes given by $i-1$, i and $i+1$ on the current path and the channels used by the links $\{i-1, i\}$ and $\{i, i+1\}$ be $c_{i-1,i}$ and $c_{i,i+1}$, respectively (Figure 5) (a). If the PU is on the channel ξ^x_p and either $c_{i-1,i} = \xi^x_p$ or $c_{i,i+1} = \xi^x_p$, the node i must search for a new channel to prevent interference to itself and to the PU, respectively. At this stage, it sends an explicit pause notification (EPN) to the source, which in turn, freezes the protocol state and waits for a new channel CHN message to resume the transmission. We consider the case where TP-CRAHN adjusts to a single affected link $\{i-1, i\}$ and then extend the analysis for the case when both the previous and next hop links need a channel change.

The set of available channels is known at node i, as described in Section III. The preferred list of channels, from this available set, is sent by this node to the previous hop $i-1$ (Figure 5 (b)). The node $i-1$ chooses a channel from this set, say ξ^x_q. It then sends back a link layer ACK to node i to inform the node of its choice, ξ^x_q. 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, ξ^x_q, as a confirmation and also to approximately estimate the new link transmission delay times $L_{i,i-1}$ and $L_{i-1,i}$. If the probe and ACK packets are of the size P_{probe} and P_{ACK}, respectively, the observed link bandwidth $W_{i,i-1}$ is,

$$W_{i,i-1} = \frac{P_{probe} + P_{ACK}}{L_{i,i-1} + L_{i-1,i}} \tag{8}$$

The CHN message contains in it the bidirectional link layer packet delay over the newly identified channel, ($L^T_{i,i-1} = L_{i,i-1} + L_{i-1,i}$), that is used by the source to calculate $W_{i,i-1}$ from equation (8). From Section IV-B2, we recall that the ACKs forwarded over the intermediate hops also carry the total bidirectional link latency, ${L'}^T_{i,i-1}$, corresponding to the earlier used channel. On receiving the CHN message, the source first estimates the new RTT using (i) the earlier observed RTT' during the last *normal* state of the protocol and (ii) adjusting for the new bidirectional link delay, $L^T_{i,i-1}$,

$$RTT = RTT' + L^T_{i,i-1} - {L'}^T_{i,i-1} \tag{9}$$

For the given path of n nodes, let W_b' be the old observed bottleneck bandwidth, before the channel change. After the channel change, the new bottleneck bandwidth is identified as W_b, where $W_b = \min\{W_{l,l+1}\}$, $l = 1, \ldots, n-1$. The updated estimate of the bandwidth $W_{i,i-1}$ is used in this calculation from equation (8). If the ratio of the old bottleneck bandwidth to the new is within the allowed range of $[1-\varpi, 1+\varpi]$, i.e. $\frac{W_b'}{W_b} \in [1-\varpi, 1+\varpi]$, then no scaling of the earlier *cwnd* is needed, where $\varpi = 0.2$. If it lies beyond this range, then we calculate the new value of the *cwnd* as follows,

$$cwnd = \alpha_c \cdot W_b \cdot RTT \tag{10}$$

The factor $\alpha_c = 0.8$ in equation (10) is used to adjust the *cwnd* to a value slightly lower than the predicted bandwidth to prevent the risk of over-estimating the *cwnd* [4]. Over time, the *cwnd* converges to the optimal value around this range. In the event that the channels of both the upstream and downstream links are changed, the bidirectional link latencies, $L^T_{i,i-1}$ and $L^T_{i,i+1}$ are used in the equations (9) and (10).

State Transitions: The *Spectrum Change* state is entered as soon as an EPN message is received. It reverts back to the *Normal* state when the new channel information is received in the CHN message or enters into the *Route Failure* state on the receipt of an ICMP message. Existing sensing schedules are ignored as long as the protocol stays in the current state.

E. Mobility Predicted State

In order to address the problem of delayed route failure notification (Section II), we develop a mobility prediction framework based on Kalman filter based estimation [6], that uses the received signal strength (RSS) information from the link layer.

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* $\leq ssthresh$, 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 TP-CRAHN reverts back to the *normal* state, where the *cwnd* is no longer bounded.

State Transitions: TP-CRAHN regards the *Mobility Predicted* state as a transient or virtual state, in which the *cwnd* is restricted to the *ssthresh* and the current operation either in the *Normal* or the *Spectrum Sensing* state is continued.

F. Route Failure State

The node i sends a *destination unreachable* message in the form of an ICMP packet if (i) the next hop node $i+1$ is not reachable based on link layer re-tries, (ii) there is no ongoing spectrum sensing based on the last known schedule, and (iii) 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 TP-CRAHN.

State Transitions: The *Route Failure* state is the terminal state of the current cycle and a fresh TCP connection must be established when a new route is formed. The protocol enters this state on receiving the ICMP message and it takes precedence over all the others states.

V. Performance Evaluation

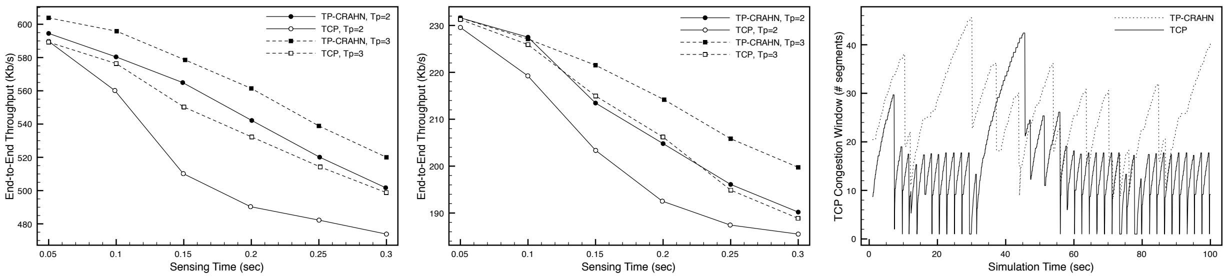
In this section, we study the behavior of TP-CRAHN under the scenarios of (i) spectrum sensing, (ii) spectrum change with PU activity, and (iii) node mobility. To the best of our knowledge, there is no existing transport layer protocol that is designed considering the CR specific functions, and protocols devised for classical ad-hoc networks cannot be compared fairly with our work. Rather, we use TCP newReno (henceforth referred to as TCP) as a benchmark and focus on how TP-CRAHN adjusts to each of the above CR scenarios through a stage-wise implementation of its modules. We extend the NS-2 simulator with multi-channel extensions and a channel switching time of $5\,\mathrm{ms}$ is used. In order to study the effect of the *cwnd* scaling, we consider 5 channels, C $= \{c_1, \ldots, c_5\}$, having varying raw channel bandwidth given by $\{B, \frac{B}{K}, B \cdot K, \frac{B}{K}, B \cdot K\}$, respectively, where $B = 2\,\mathrm{Mbps},\ K = 2$. In addition, a priority queue is implemented at the link layer, as described in Section III, and the allowed retries for the feedback packets in TP-CRAHN is raised to 20. Out of 100 nodes any source-destination pair is chosen forming a chain topology and we vary the number of flows in the path. A parallel chain is then created, that uses the same channel selection as our test chain topology. The corresponding nodes of the two chains are placed within transmission range of each other and this provides the link contention for the bandwidth calculation. The transmission ranges of the PU and the CR users are $300\,\mathrm{m}$ and $120\,\mathrm{m}$, respectively and the desired throughput $\tau_d = 800\,\mathrm{Kbps}$.

A. Spectrum Sensing

The evaluation of TP-CRAHN during spectrum sensing is carried out in two parts - (i) by observing the improvement in throughput resulting from the change in the *ewnd* (Section IV-C1), and (ii) the benefit of reduction of the sensing duration in the absence of PU activity (Section IV-C2).

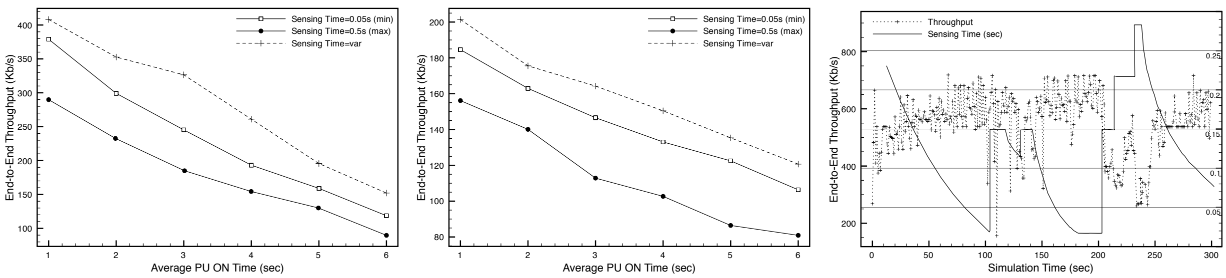
Figures 6(a) and 6(b) show the end-to-end throughput for varying network loads as the sensing time of the nodes is increased to the maximum value $t^s = t^s_{max} = 0.3$, for a constant data transmission time T_p. In the first set of experiments, we disable the dynamic adjustment of the sensing time. We observe that TP-CRAHN outperforms TCP significantly as it

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.



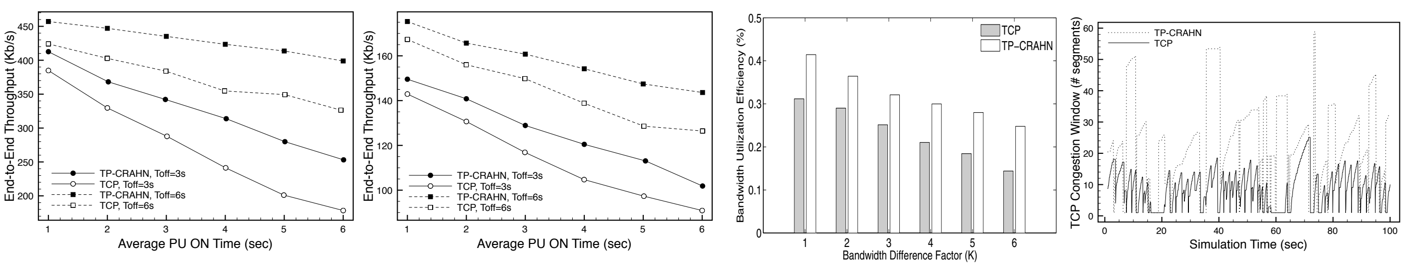
(a) (b) (c)

Fig. 6. The effect of spectrum sensing on the throughput is shown for 1 and 5 flows in (a) and (b), respectively. Variation of the congestion window with time is given in (c).



(a) (b) (c)

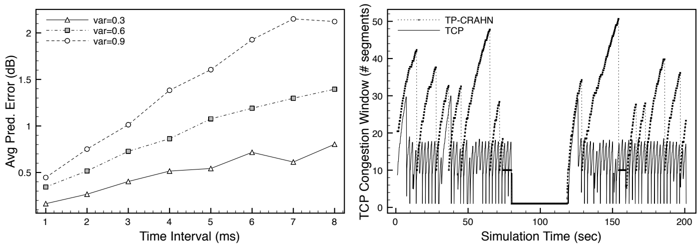
Fig. 7. The effect of dynamically changing the sensing duration on throughput is shown for 1 and 5 flows in (a) and (b) respectively. A study of the throughput as a function of the varying sensing time is given in (c).



(a) (b) (c) (d)

Fig. 8. The effect of the bandwidth scaling adjustment on throughput is shown for 1, and 5 flows in (a), and (b), respectively. The bandwidth utilization efficiency and the *cwnd* scaling are shown in (c) and (d), respectively.

does not *stop* transmitting when the path gets disconnected, but transmits at a reduced rate to prevent a buffer overflow. This ensures a higher throughput in TP-CRAHN, as TCP suffers an RTO timeout whenever a node in the path undertakes sensing. This phenomenon is also seen in Figure 6(c), where the *cwnd* almost never reduces to 1 for the case $t^s = 0.15, T_p = 3$, unless there is congestion in the network (at sim. time = $70\,\text{s}$). Compared to this, TCP undergoes repeated timeouts during the sensing and the *cwnd* is forced to the slow start phase in the absence of true congestion. For the study of the dynamically changing sensing duration, we first define the PU operation as follows: We continuously vary the *on* time (T_{on}) of the PU on the x-axis, and measure the throughput for different PU *off* times, T_{off}, for 1 and 5 flows as shown in Figures 7(a) and 7(b), respectively. We first assign a a low PU susceptibility of 0.01 to a randomly chosen number of nodes S_{low} in the path, and for the remaining nodes, forming the set S_{high}, we set an initial high susceptibility value of 0.95. This ensures that TP-CRAHN changes the sensing duration from the maximum value $t^s_{max} = 0.28$ only for the nodes present in the set S_{low}. When the sensing time is varied dynamically by TP-CRAHN, (shown by sensing time = var), we observe that the throughput improves significantly for both the cases of 1 and 5 flows (Figure 7(a) and 7(b)). The variation of the *cwnd* for throughput as a function of the changing sensing time is shown in Figure 7(c).The sensing time falls in a non-linear manner (Section IV-C2), and in the absence of PU activity, this improves the throughput. For successive PU missed detections, the sensing time is scaled to $\frac{1}{2} \cdot t^s_{max}$, then $\frac{3}{4} \cdot t^s_{max}$ and finally to the maximum value t^s_{max}. For each increase in t^s, though the throughput is partially reduced, the mutual interference with the PU is also mitigated.



(a) (b)

Fig. 9. The Kalman filter accuracy and the variation in the *cwnd* with time are shown in (a) and (b), respectively

B. Spectrum Change and PU Activity

From Section IV-D, we recall that when PU activity is detected, TP-CRAHN stops the source from transmitting and coordinates the use of a new channel. The source then modifies its *cwnd*, if the new channel on the affected link significantly changes the bottleneck bandwidth available to the connection. We study the performance improvement in TP-CRAHN by considering the throughput, the bandwidth efficiency (ratio of the available bandwidth to average used bandwidth of the bottleneck link), and the variation in the *cwnd*.

A PU is placed on each of the 5 possible channels so that the channel (and hence, the bandwidth) change often and its effect is clearly demonstrated. Figures 8(a) and 8(b) give the throughput for 1 and 5 flows, respectively, when PUs exist on the channel. We observe that the throughput improvement in TP-CRAHN increases with higher PU activity, formally defined as $\sigma = \frac{T_{on}}{T_{on}+Toff}$. For lower values of σ, i.e. when T_{on} is small, the channels are readily available and the gain in TP-CRAHN is due to the scaling of the *cwnd* alone. For higher values of σ, when T_{on} is large, it takes longer for the affected node to find a vacant channel. This delay dominates the network performance and by explicitly specifying the source to pause its transmission, TP-CRAHN prevents packet loss and improves the throughput.

The effect of *cwnd* scaling results in higher bandwidth efficiency in TP-CRAHN, as seen in Figure 8(c). Moreover, the performance improves when the difference in the raw bandwidths of the available channels (given by increasing the factor K) is higher, implying that the forced scaling of the *cwnd* is effective in fully utilizing the spectrum resource. The variation of the *cwnd* against time in Figure 8(d) shows that TP-CRAHN responds to the changed bandwidth immediately and after the scaling, the *cwnd* increases in most of the cases, without causing congestion.

C. Mobility Prediction

The error between the true signal strength (RSS) and the value predicted by the Kalman filter is shown in Figure 9(a) as a function of the epoch interval used between two successive calculations. In our chain topology, nodes move with an average speed of $10\,\mathrm{m/s}$ with locations given by the random waypoint model. The variance (var) in the RSS in dB, due to external noise affects the prediction accuracy as shown in the figure. We observe two cases in Figure 9(b), the first of which shows a route disconnection at time = $70\,\mathrm{s}$. The conservative reduction in the *cwnd* resulted in fewer packets that were lost in the unusable route, as compared to TCP. An incorrect prediction that restricted the *cwnd* to the *ssthresh* till the timeout period is shown at time= $150\,\mathrm{s}$.

VI. Conclusions

Our proposed 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. Future research in this direction would involve proposing an improved predictive framework to reduce the dependence on intermediate feedback by the nodes. In addition, the quality of service demands of the flows will be integrated in the TP-CRAHN protocol.

Acknowledgement

This material is based upon work supported by the US National Science Foundation under Grant No: CNS-07251580. Marco Di Felice was supported by Italian Miur funds for the PRIN-2006 project NADIR.

References

[1] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. *Elsevier Computer Networks Journal*, 50:2127–2159, Sept. 2006.
[2] I. F. Akyildiz, W. Y. Lee, and K. Chowdhury, CRAHNs: Cognitive Radio Ad Hoc Networks. *Elsevier Ad Hoc Networks Journal*, to appear, 2009.
[3] A. Capone, L. Fratta and F. Martignon, Bandwidth Estimation Schemes for TCP over Wireless Networks. *IEEE Trans. on Mobile Computing*, 3(2):129–143, April-June 2004.
[4] K. Chen, Y. Xue and K. Nahrstedt, On Setting TCP's Congestion Window Limit in Mobile Ad Hoc Networks. *in Proc. of IEEE ICC*, pp. 1080-1084, May 2003.
[5] G. Holland and N. H. Vaidya, Analysis of TCP Performance over Mobile Ad Hoc Networks. *in Proc. of ACM MOBICOM*, pp. 219-230, Seattle, WA, Aug. 1999.
[6] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory. *Prentice-Hall, Inc.*, isbn = 0-13-345711-7, 1993.
[7] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi and R. Wang, TCP westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. *in Proc. of ACM MOBICOM*, pp. 287–297, Rome, Italy, 2001.
[8] W. Y. Lee and I. F. Akyildiz, Optimal Spectrum Sensing Framework for Cognitive Radio Networks. *IEEE Trans. on Wireless Comm.*, vol. 7, no. 10, Oct. 2008.
[9] J. Liu and S. Singh, ATCP: TCP for Mobile Ad Hoc Networks. *IEEE Journal on Sel. Areas of Comm.*, 19(7):1300–1315, 2001.
[10] A. M. R. Slingerland, P. Pawelczak, R. V. Prasad, A. Lo and R. Hekmat, Performance of Transport Control Protocol Over Dynamic Spectrum Access Links. *in Proc. of IEEE DySPAN*, pp. 486–495, April 2007.
[11] K. Sundaresan, V. Anantharaman, H-Y. Hsieh, and R. Sivakumar, ATP: A Reliable Transport Protocol for Ad Hoc Networks. *IEEE Trans. on Mobile Computing*, 4(6):588–603, Nov. 2005.
[12] H. Xiao, K. C. Chua, J. A. Malcolm and Y. Zhang, Theoretical Analysis of TCP Throughput in Adhoc Wireless Networks. *in Proc. of IEEE GLOBECOM*, pp. 2714–2719, St. Louis, Mo, Nov-Dec 2005.
[13] X. Yu, "Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-layer Information Awareness," in *Proc. of ACM MOBICOM*, Philadelphia, PA, USA, pp. 231-244, September 2004.