Modeling and Performance Evaluation of Transmission Control Protocol over Cognitive Radio Ad Hoc Networks

Marco Di Felice
Department of Computer Science
University of Bologna, Italy
difelice@cs.unibo.it

Kaushik Roy Chowdhury
Department of Electrical and Computer Engineering
Northeastern University
Boston, USA
krc@ece.neu.edu

Luciano Bononi
Department of Computer Science
University of Bologna, Italy
bononi@cs.unibo.it

ABSTRACT

Cognitive Radio (CR) technology constitutes a new paradigm to provide additional spectrum utilization opportunities in wireless ad hoc networks. Recent research in this field has mainly focused on devising spectrum sensing and sharing algorithms, to allow an opportunistic usage of licensed portions of the spectrum by Cognitive Radio Users (CRUs). However, it is also important to consider the impact of such schemes on the higher layers of the protocol stack, in order to provide efficient end-to-end data delivery. Since TCP is the de facto transport protocol standard on Internet, it is crucial to estimate its ability in providing stable end-to-end communication over Cognitive Radio Ad Hoc Networks (CRAHNs). The contributions of this paper are twofold. First, we propose an extension of the NS-2 simulator to support realistic simulation of CRAHNs. Our extension allows to model the activities of Primary Users (PUs), and the opportunistic spectrum management by CRUs in the licensed band. Second, we provide an accurate simulation analysis of the TCP performance over CRAHNs, by considering the impact of three factors: (i) spectrum sensing cycle, (ii) interference from PUs and (iii) channel heterogeneity. The simulation results show that the sensing interval and the PU activity play a critical role in deciding the optimal end-to-end performance, and reveals the inadequacy of classical TCP to adapt to variable spectrum conditions.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design – Wireless Communication.

General Terms

Performance, Design, Experimentation.

Keywords

Cognitive Radio Ad Hoc Networks, Spectrum Management, Transmission Control Protocol (TCP), Modeling and Simulation.

MSWiM'09, October 26–29, 2009, Tenerife, Canary Islands, Spain.

1. INTRODUCTION

The increasing deployment of wireless ad-hoc networks in military and commercial applications is revealing the current inadequacy of the fixed spectrum assignment policy [1]. According to the Federal Communications Commission (FCC) report [1], most of the spectrum bands (e.g. TV band) which are allocated to licensed holders are used sporadically while the 2.4 GHz ISM band (unlicensed band) used by 802.11 wireless ad-hoc networks is becoming increasingly saturated. The disparity in spectrum availability between the crowded unlicensed band and the under-utilized licensed band requires a new communication paradigm, where wireless devices are allowed to use the spectrum in an opportunistic way. Cognitive radio (CR) [2] technology is currently proposed to solve the spectrum inefficiency problems.

Cognitive Radio Ad Hoc Networks (CRAHNs) [2,3] are composed of two kind of users: cognitive radio users (CRUs) and primary users (PUs). PUs have license to access the licensed spectrum. CRUs access the licensed spectrum as "visitors", and have the capabilities of sensing the available spectrum bands to detect spectrum holes, reconfiguring radio frequency, and switching to another spectrum band if PUs activity is detected on the current one. The CR concept is extended in CRAHN, in which the network is deployed in an ad-hoc manner with no centralized controllers. Most of research on CRAHNs has focused on designing efficient spectrum sensing and sharing algorithms at physical (PHY) and medium access control (MAC) layers [2,3]. However, as CRAHNs operate in a multi-hop fashion, it is also important to integrate these algorithms in the implementation of the end-to-end network protocols, i.e. routing and transport protocols. In this paper, we highlight the research challenges of CRAHNs from the perspective of the transport layer.

Transmission Control Protocol (TCP) is the de-facto standard transport protocol on the Internet. TCP provides reliable end-to-end data transfers, and implements rate and congestion control mechanisms, based on the estimation of channel capacity and end-to-end round-trip time (RTT) between sender and receiver nodes. Packet losses are used as indicators of network congestion [9-11]. However, packet losses in wireless environment can be caused by nodes' mobility or by channel errors, apart from network congestion. As a result, TCP is shown to perform poorly over wireless links, and several variants of TCP have been proposed for wireless ad hoc networks [5,8]. Most of these solutions retains the window-based approach, but at the same time exploits feedback from lower protocol layers, so that TCP is able to detect packet losses caused by congestion, nodes' mobility or channel errors [5-8]. TCP over CRAHNs should face at least three additional challenges, which are not present in traditional wireless ad hoc networks [2-4]:

- *Impact of spectrum sensing*. In order to detect the presence of PUs, CRUs periodically monitor the current channel over a pre-decided sensing duration. During this interval, the nodes are not actively involved in transmitting data packets, and hence, cannot forward in time the TCP-ACKs from the destination back to the source node. This may trigger re-transmission timeout (RTO) events, as the source may mistake the sensing-induced delay as network congestion. Similarly, the sensing duration must be carefully decided to balance the tradeoff between protection of PUs and the CRAHNs end-to-end performance [4].
- *Impact of PU activity*. When a PU is detected, the CRUs must cease their operation on the affected channel and search for a different vacant portion of the spectrum. This phase is called *spectrum handoff*. As the duration and frequency of spectrum handoff is not predictable in advance, spectrum handoff operations may be misinterpreted as congestion conditions. The resulting rate decrease at the source also adversely affects the end-to-end network performance [4].
- *Impact of spectrum mobility*. Since CRs devices are designed to opportunistically utilize a wide range of frequency bands, a large set of frequency-separated channels might be available. The set of available channels for CRUs may be "heterogeneous", i.e. different channels may support different data rates and delay characteristics [12]. As a result, channel switching in CRAHNs may result in brief reduction or increase of network capacity that must be considered as a factor deciding the rate control at the source node.

Transport protocols constitute an unexplored area for CRAHNs. At the best of our knowledge, there is a lack of prior research providing an accurate evaluation of TCP performance over CRAHNs [4,7]. Even more challenging is the proposal of novel transport layer solutions for CRAHNs. Generally speaking, the performance evaluation of end-to-end protocols for CRAHNs requires accurate simulation models and tools. However, existing network simulators (e.g. NS-2, Opnet, Qualnet) are not completely suitable for CR modeling and simulation, due to the following reasons. First, most of them do not adequately address interference and signal detection issues at the PHY layer, which are required to model PUs activities. Second, they do not implement opportunistic spectrum access schemes, which are required to model CRUs activities, i.e. spectrum sensing, decision and mobility schemes. Third, their strict layered organization makes difficult to implement cross-layer solutions based on data exchange among protocols at different layers of the protocol stack, which constitutes an important consideration in CRAHNs.

In order to address these issues, we first propose a valuable extension of the NS-2 tool [16], for the realistic modeling and simulation of CRAHNs. The extended tool provides accurate modeling of PUs activities in a multi-channel wireless environment. Moreover, it models spectrum management functionalities of CRUs, i.e. spectrum sensing, spectrum decision and spectrum mobility functionalities. The block architecture of NS2 is revised, in order to meet two design issues: (i) cross-layering, i.e. how to simplify the exchange of variables and parameters between protocols at different network layers, and (ii) extendibility, i.e. how to enable the seamless integration/replacement of spectrum management models in the NS-2 architecture. Then, we use the extended version of NS-2 to study the performance of TCP over CRAHNs. Since the impact of nodes' mobility and channel errors on TCP performance have been studied in many previous works [6,12], in this paper we focus on the impact of three specific characteristics of CRAHNs on end-to-end performance: (i) sensing cycle, (ii) PU interference and (iii) channel heterogeneity. We evaluate different versions of TCP, under different channel and network conditions, considering the TCP behavior at both macroscopic level, i.e. measuring the TCP aggregated throughput, and microscopic level, i.e. studying the dynamics of the Round Trip Time (RTT) metric and congestion window (CW) size, over simulation time. Simulation results highlight the problems of TCP which must be taken into account in the design of novel transport layer protocols for CRAHNs.

The paper is organized as follows. Section 2 reviews classical TCP and TCP-variants proposed for wireless ad hoc networks. The system model used in the analysis is described in Section 3. Section 4 provides an overview of the extended version of the NS-2 tool, which has been used for the simulation of CRAHNs. Section 5 describes simulation results of TCP over CRAHNs. Finally, Section 6 presents the conclusions and discusses future works.

2. RELATED WORKS

Since its original proposal in 1974, several versions of TCP have been proposed for wired networks [9-11]. All of them provide congestion and source-rate control, by means of a congestion window (CW) which limits the total number of un-acknowledged packets which can be in transit end-to-end. Modern implementations of TCP operate in four different protocol phases: slow start, congestion avoidance, fast retransmit and fast recovery. In the slow start phase, the CW size grows exponentially by one segment for each TCP-ACK received. In the congestion avoidance phase, the CW size is increased by one segment per round-trip-time (RTT), till a packet loss occurs. Packet losses are used as an indicator of congestion. In such a case, the rate is decreased by reducing the CW size at the sender side. Different versions of TCP differ in the way they detect and react to packet losses. TCP New-Reno [9], which is probably the most commonly used TCP on Internet, reduces the CW size when (i) three duplicate TCP-ACK packets are received (ii) or when the Retransmission Time Out (RTO) of a segment expires before receiving the corresponding TCP-ACK. Other modern variants of TCP are Vegas [10] and TCP-SACK [11]. TCP Vegas adapts the current CW size at the sender side, so that the number of queued packets in the network is always between a minimum and maximum threshold value. TCP-SACK implements a selective retransmission scheme, so that only packets actually missing are retransmitted by the sender node. While all these variants work well over stable wired connections, many recent papers have investigated and provided evidence that TCP performs poorly on wireless environments, and on mobile ad hoc networks in particular. Simulation studies for wireless networks have investigated the impact of hidden terminal, wireless channel errors and nodes' mobility on TCP performance [6,12].

Moreover, many TCP modifications have been proposed for wireless ad hoc networks. Ad hoc TCP (ATCP) [8] utilizes network feedback to detect packet losses caused by mobility or by channel errors rather than by network congestion. In ATCP, the standard TCP is not modified but a thin layer between IP and transport layer is added to filter network feedback and to adapt the CW accordingly. A different approach is exploited by the authors of [5], where a completely new protocol (called ATP) is developed for wireless ad hoc networks. ATP decouples congestion control from reliability mechanisms, and exploits feedback from the intermediate nodes traversed by the connection to adapt the sending rate. Other techniques attempt to enhance the TCP performance on wireless environment by designing novel solutions at the MAC or link layers [14,15]. While TCP over traditional ad hoc networks constitutes a well investigated research area, there is a lack of papers addressing transport protocols for CRAHNs. In [4], a novel transport protocol (TP-CRAHN) is proposed for CRAHNs. TP-CRAHN retains the window based approach of TCP, but it also comprises additional protocol phases to adapt the protocol behavior to specific network conditions, such as sensing cycle intervals, spectrum and mobility handoffs, channel bandwidth variations, etc. In [7], an evaluation of TCP over Dynamic Spectrum Access (DSA) links is proposed. Compared to the work presented in [7], we provide the following novelties: (i) we separately analyze the impact of different CRAHNs characteristics such as sensing time, PU activity and channel heterogeneity, on TCP performance (ii) we study the TCP behavior at both macroscopic level e.g. measuring the aggregate throughput, and at microscopic level e.g. analyzing the RTT and CW dynamics over time, and (iii) we address modeling and simulation issues for CRAHNs, which are described in Section 4.

3. SYSTEM MODEL

CRUs forming the CRAHNs are equipped with a single radio transceiver, which can be tuned to any channel in the licensed band. We assume B primary spectrum bands are present, with n CRU channels in each band. Channels are assumed to be "heterogeneous", i.e. two channels in different spectrum bands may have dissimilar raw channel bandwidth. The PU activity results in a spectral overlap with the CRU channels, i.e. each PU transmission on a primary user band may affect CRU channels with different extent. We consider the overlap factors as 1, 0.5 and 0.25 for 0, 1 and 2 channel spacings from the PU's central channel frequency, respectively.

The PU activity is modeled by the alternative exponential ON-OFF model [13]. In this model, each PU has two alternative states: ON and OFF. An ON (busy) state represents the period in which the primary band is occupied by PU, while an OFF (idle) state represents the period in which the primary band is idle and can be used by CRUs. The switching between ON/OFF states is regulated by a birth-death Markovian process. Let α_i be the death rate for PU on spectrum i (e.g. PU_i), then the duration of ON state follows an exponential distribution with mean $1/\alpha_i$. Similarly, let β_i represents the birth rate for PU_i, then the duration of OFF state follows an exponential distribution with mean $1/\beta_i$. Each CRU alternately senses the channel and transmits data with observation time equal to t_s and transmission time equal to T_p. The sensing model is based on a maximum a posteriori (MAP) energy detector scheme [13]. Let P_d the probability a CRU correctly detects the activity of PU_i on channel i which has bandwidth W, after sensing the channel for a t_s interval. Let P_f the probability to have a false alarm PU detection. Both P_d and P_f can be expressed as a function of α_i, β_i, W and t_s. The exact formulation of P_d and P_f can be found in [13].

4. SIMULATION OF CRAHNs USING NS-2

The architecture block of the revised NS-2 simulator is shown in Figure 1.

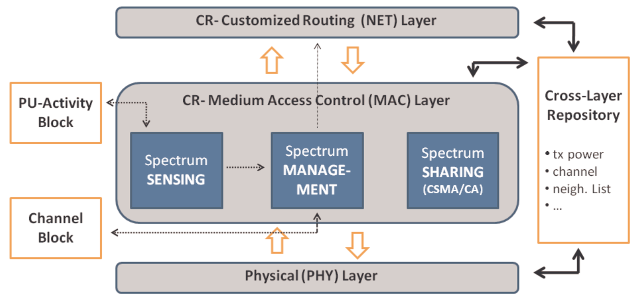

Figure 1. NS-2 Revised Architecture Block.

Compared to the traditional NS-2 architecture [16], we added the following features, implemented as extendible stand-alone C++ modules:

- PU Activity Block. It describes the characteristics of active PUs in the current scenario, i.e. operating primary spectrum band, location, transmitting range. It also contains the description of PU activity in each spectrum band, as a sequence of ON and OFF periods over simulation time. All PUs information are contained in a PU-log file, which is generated off-line. Format of PU-log file is described in Section 4.1.
- Channel Log Block. It contains a channel table information, with the parameters describing the PHY characteristics of each CRU channel in each PU spectrum band, such as operating frequency, channel capacity and amount of noise. Format of Channel-log file is described in Section 4.2.
- Spectrum Sensing Block. It implements the spectrum sensing functionalities for CRUs. It interacts with the PU Activity Block to perform PUs detection. The output of Spectrum Sensing Block is provided to the Spectrum Management Scheme in case a PU is detected on the current CRU channel, and channel switching operations are required.
- Spectrum Management Block. It implements channel decision and channel switching functionalities. If a PU activity is detected on the current CRU channel, the spectrum management scheme chooses the next available channel to be used. Spectrum handoff delay is modeled. Moreover, it notifies the upper layer about the occurrence of a channel switching operation.
- Spectrum Sharing Block. It implements the operations of distributed channel access in wireless ad hoc networks. Current implementation is based on the CSMA/CA MAC scheme. It interacts with the PU Activity Block to model the interference caused by PUs on current ongoing transmissions of CRUs.
- Cross-Layer Repository. It enables information sharing among protocols at different layers of the protocol stack. For example, it may contain information collected at PHY layer (e.g. current transmitting power), MAC layer (e.g. current channel) and Network layer (e.g. current neighbors' list).
- Network Protocol for CRAHNs. Traditional routing protocols for wireless ad hoc networks can be used at network layer, as well as customized network protocols for CRAHNs. We provide the implementation model of the SEARCH routing scheme [3].

In the following, we provide a detailed description of PU-log and Channel-log file formats. Moreover, we give an insight into the rationale of the CRU model.

4.1 PU Activity Log File Format

The PU-log file contains information about (i) PUs location and characteristics and (ii) PUs activity over time. The first part of the file is composed of entries with this format:

$$< i, x, y, spectrum, tpower >$$

where i is the unique identified of a PU, x, y are its location, $spectrum$ and $tpower$ are the PU spectrum band and transmitting power used by PU, respectively. The second part of the file describes the activity of each PU over time, by means of a list of entries with this format:

$$< i, arrival_{time}, departure_{time} >$$

where i is the PU identifier, $arrival_{time}$ is the simulation time when the PU enters the ON period and starts transmitting, and $departure_{time}$ is the simulation time when the PU enters the OFF period and ceases transmitting. Average ON and OFF time are regulated by an exponential distribution with mean $1/\alpha_i$ and $1/\beta_i$, based on the ON/OFF model described in Section III. We develop a script shell program which takes as input the α_i and β_i parameters, and generates the activity over simulation time for PU_i.

4.2 Channel Log File Format

The Channel(CH)-log file provides information about (i) channel characteristics and (ii) noise effect over time. For each CRU channel, the log file contains an entry with this format:

$$< spectrum, id, bandwidth, noise >$$

where $spectrum$ is the primary band identifier (a number between 1 and B), id is the identifier of CRU channel on the current primary spectrum band (a number between 1 and n), $bandwidth$ is the maximum bandwidth supported by the channel, and $noise$ is the average value of the noise on that channel. Moreover, we allow to model channel configurations where the noise values change over time. In this case, the CH-log file contains the noise values at regular time intervals, for each CRU channel.

4.3 CRU Model Description

Each CRU periodically cycles between two states: a sensing state (for a t_s time interval) and transmitting state (for a T_p time interval). To this aim, each CRU (say node C) is associated a sensing timer and a transmitting timer. When the sensing timer expires, C stops sending data on the current CRU channel (say channel i) and performs sensing operations to detect the presence of a PU. Channel sensing is modeled as a lookup function on the PU-log file, for channel i and simulation time t. In particular, node C checks the PU-log file if there is an entry for a PU (say P) such that:

1. P is transmitting on the same spectrum band of node C for the time interval [t; t+ t_s];
2. The amount of power injected on channel x by node P and received by node C (i.e. P_r^C) is higher than a sensivity threshold for node C (i.e. P_{th}^C), i.e.:

$$P_r^C \geq P_{th}^C \tag{1}$$

Signal propagation is modeled through a generalized free-space model, i.e. P_r^C is computed by node C as follows:

$$P_r^C = \frac{P_t}{d^a} \cdot k \tag{2}$$

where P_t is the transmitting power of P, a is the attenuation factor, d is the actual distance between P and C, and k is the overlap factor between i and the central CRU channel frequency used by P in its primary spectral band. If both conditions (1) and (2) are verified, then the variable *puON_* is set as *true*. However, this is may not be enough to guarantee a correct PU detection, because node C might mis-detect the presence of P, or might incur in a false-positive detection error. The probability of correct detection and false positive are described in [13]. In the current implementation, we used a simplified version of equations for P_d and P_f , as a function of sensing time interval (t_s) only. The Spectrum Sensing Block detects the presence of a PU in the following cases: (i) *puON_= true*, with probability P_d or (i) *puON_ = false*, with probability P_f. In such cases, the Spectrum Management Block is responsible for finding another available channel, based on the current spectrum decision policy. In the current implementation, we used a simple round-robin spectrum policy, so that the $next_{channel}$ to be used is given by:

$$next_{channel} = (current_{channel} + 1)\%(B \cdot n) \tag{3}$$

An additional timer is implemented to model the delay caused by channel switching operations. If $next_{channel}$ is found free of PU activities, then a spectrum handoff notification is sent to the upper layer, and protocol reconfiguration is performed at network layer. Moreover, the characteristics of the new channel are loaded with the CH-log entry information for $next_{channel}$. Then, the CR starts transmitting data on $next_{channel}$ for a time interval regulated by T_p. Many spectrum sharing algorithms have been proposed for CRAHNs [2,3]. In our simulation study, we used a revised version of Carrier Sensing Multiple Access with Collision Avoidance (CSMA-CA) MAC scheme, which also implements acknowledgments (ACK) and frame retransmissions at MAC layer. Moreover, we extended the MAC scheme to take into account the interference caused by PUs on CRUs which are currently transmitting on $current_{channel}$. Each time a CRU (say node C) starts receiving a packet by another CRU (say C2), it checks if a PU (say P) is currently transmitting on the primary spectrum band of $current_{channel}$, or if it will start transmitting during the reception time. In such a case, the amount of power injected on $current_{channel}$ by P is computed at node C, based on the propagation model described by Equation 2. Based on P_r^C, the actual Signal-to-Noise-Ratio (SINR) is computed by node C. If the SINR is lower than a given threshold (which is usually referred in NS-2 as *CPTresh*), then the current packet transmitted by node C2 is discarded by node C at the MAC layer. The algorithm describing the CRU model is shown in the pseudocode below.

```
1 puON_=lookup(channel, t_s, switching)
2 if (puON_ && Pd ) || (!puON_ && Pf )
3     channel=(channel + 1) % (B*n)
4     defer for CHANNEL_SWITCHING_DELAY
5     switching=true
6     go back to 1
7 if (switching)
```

notify upper layers about spectrum handoff

load channel parameters

start T_p timer

transmit data till T_p expires, then go back to 1

5. TCP PERFORMANCE EVALUATION

Before presenting the simulation results, we describe the simulation setup used for the evaluation of TCP over CRAHNs. Unless stated otherwise, the parameters used in the simulation are shown in Table 1.

Table 1. Simulation Environment Specifications

Number of primary bands	5
CRUs channels on each primary band	1
Traffic Type	TCP-FTP
Packet Size	1000 byte
CRU Transmission range	250m
PU Transmission range	500m
PU parameters α - β	Variable
Sensing Time t_s	Variable
Transmitting Time T_p	Variable

We used the NS-2 simulation tool [16], with the extensions described in Section 4. We investigated the performance of different TCP variants over CRAHNs, i.e. TCP Reno, TCP NewReno, TCP with Selective Acknowledgment (TCP SACK) and TCP Vegas. The CRAHN environment is constructed as follows. We assume that *5* PU spectrum bands are present (i.e. *B*=5). Moreover, we assume that the number of CRU channels is equal to the number of primary bands (i.e. *n*=1), i.e. there is complete overlapping between CRU channels and PU spectrum bands. This assumption makes easier to investigate the impact of PUs interference on CRUs using the same spectrum band. Individual spectrum bands are occupied randomly and independently of each other by PUs, according to the ON/OFF model described in Section 2. The average duration of ON/OFF state is the same, in all the spectrum bands. We let $1/\alpha$ and $1/\beta$ be the average ON and OFF period of a spectrum band, respectively. We considered two different network scenarios: a static single-hop CRAHN scenario, and a mobile multi-hop CRAHN scenario. In the first case, the network is composed of 2 CRUs. A TCP/FTP connection is established between the static CRUs. No mobility effect is considered at this stage. We studied the performance of TCP under different CRAHNs characteristics, e.g. the sensing time interval of CRUs, interference caused by PUs activity and bandwidth variation in an heterogeneous channel environment. The choice of the single-hop scenario can be motivated as follow. First, the single-hop scenario is simple enough to understand the impact of CRAHNs characteristics on the dynamics inside the TCP, while this might be difficult to investigate in multi-hop topologies. Second, the single-hop topology constitutes a "base case", from the point of view of protocol performance. If we discover that a single parameter, e.g. the sensing time interval, has a strong impact on TCP performance, then this effect would be emphasized in a multi-hop environment by the presence of multiple intermediate nodes between the source and the destination pair. Moreover, although very simple, the single-hop topology constitutes a realistic model for the evaluation of infrastructure-based CR networks, where the mobile CRUs are attached to a fixed Cognitive Base Station (CBU). TCP performance over infrastructure-based CR networks is also investigated in [7]. In the second scenario, we considered multi-hop CRAHNs topologies, composed of 25 mobile CRUs. We varied the number of active TCP/FTP connections. The multi-hop scenario is used to evaluate end-to-end TCP performance when all the CRAHNs characteristics are considered. We considered two metrics in the performance analysis:

- TCP Throughput: this is the end-to-end TCP throughput at the application layer, i.e. the amount of bits for seconds received by the upper layer FTP application at the destination node, without considering out-of-order, duplicated and TCP-ACK packets.
- TCP Efficiency: this is an estimation of bandwidth resource utilization by TCP. It is defined as follows [7]:

$$\varepsilon = \frac{\mathrm{TCP_{THR}}(t_1, t_2)}{\int_{t1}^{t2} C(t)dt}, 0 \leq \varepsilon \leq 1 \tag{4}$$

where $\mathrm{TCP_{THR}}(t_1, t_2)$ is the average TCP throughput computed over the measurement period from t_1 to t_2, and $C(t)$ is the available channel capacity at time t [7].

In Sections 5.1, 5.2 and 5.3 we show the performance results of TCP in the single-hop topology, when we varied the sensing time interval of CRUs (Section 5.1), the PU activity (Section 5.2) and the difference in bandwidth between adjacent CRU channels (Section 5.3). For each test, we describe the simulation setup, we show the simulation results and we discuss the design issues to enhance TCP performance over CRAHNs. In Section 5.4 we show the simulation results of TCP over generic multi-hop multi-flow random topologies.

5.1 Sensing Cycle Analysis

Simulation Setup. In the Sensing Cycle Analysis, we considered the basic single-hop topology, and we varied the sensing interval (t_s) and transmitting interval (T_p) of CRUs. All the primary bands have capacity equal to 1 Mb/s. We set the α and β parameters equal to 0.5, so that the average ON and OFF time of a spectrum band is equal to 1s in both cases. We consider the TCP throughput at the application layer as defined in Section 5.

Simulation Results. Figures 2-5 show the TCP performance for the Sensing Cycle Analysis. Figure 2 shows the TCP NewReno throughput as a function of sensing interval (t_s, on the *x*-axis) and transmitting interval (T_p, on the *y*-axis). Figure 2 confirms that the selection of < t_s, T_p > parameters by CRUs has a remarkable impact on the TCP performance. In particular, given parameters α and β which describe the PUs activity, there exists an optimal selection of <t_s, T_p> parameters which maximizes the TCP performance. In Figure 2, the maximum throughput is reached when t_s and T_p are equal to 0.2s and 1s, respectively. This is in accordance with the theoretical analysis described in [13]. Figure 3 shows the throughput of NewReno TCP in the same scenario, where T_p is fixed to 0.6s, and t_s varies between 0.025s and 3s. In Figure 3, we also show different curves for throughput experienced by different TCP versions.

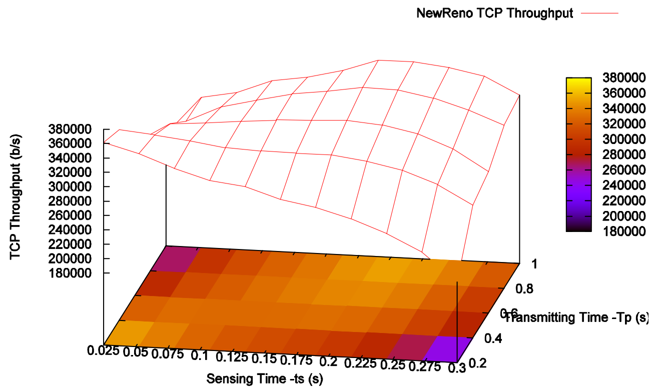

Figure 2. Sensing Cycle Analysis: NewReno TCP Throughput, variable transmitting and sensing time intervals.

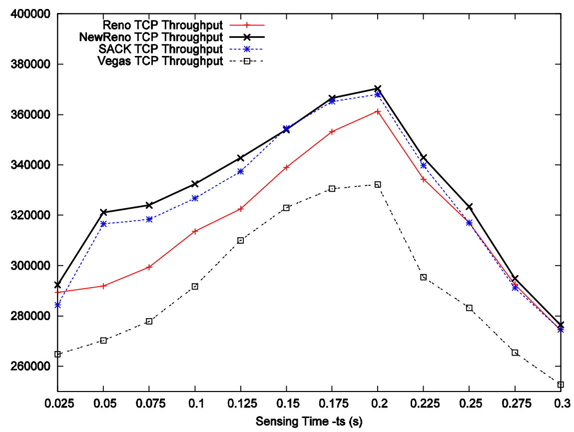

Figure 3. Sensing Cycle Analysis: TCPs Throughput.

From Figure 3, we can see that the throughput increases with t_s till a critical threshold is reached. When $t_s > 0.2$s, all the protocols experience throughput degradation. Such a behavior is independent of the TCP version used, and can be explained by the fact that the selection of the sensing interval constitutes a trade-off between (i) accurate PU detection and (ii) efficient channel utilization. In Section 3, we introduced the probability of correct detection (P_d), as a function of t_s [13]. When t_s is short, the probability of correct detection (P_d) is low. As a result, a CRU might start transmitting on a channel which is already used by a PU, because of PU mis-detection. Interference caused by the transmission of PU on the current channel may cause TCP-DATA and TCP-ACK packet losses for CRUs, and thus poor TCP performance. When t_s is longer than 0.2s, the throughput decreases because CRU consumes time in monitoring the channel rather than in data transmitting. Generally speaking, longer sensing interval translates in longer RTT of transmitted segments at transport layer, and thus in reduced TCP rate at sender side. Moreover, Retransmission Timeout (RTO) events may be triggered at sender side because of ACK packets which are delayed at MAC layer, even in the absence of true congestion. RTO events are interpreted as indicators of network congestion, and force TCP sender to shrink the CW size. As a results, TCP experiences performance degradation, when the sensing time interval is longer than 0.2s. Moreover, Figure 3 shows that TCP SACK and TCP NewReno overtake TCP Reno and TCP Vegas in terms of throughput. Surprisingly, TCP Vegas provides the lowest performance for all the configurations of t_s, but we found that this might be caused by an incorrect protocol implementation in NS-2. We are currently investigating this phenomenon.

In Figures 4 and 5, we analyze the dynamics inside TCP at sender side, i.e. we monitor the behavior of the CW size and RTT index under different configuration of t_s.

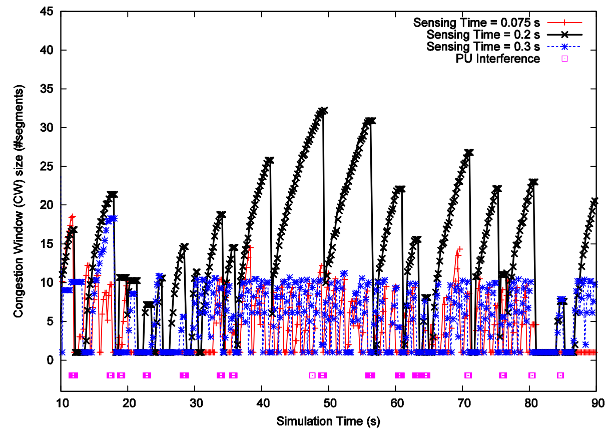

Figure 4. Sensing Cycle Analysis: CW size over Time.

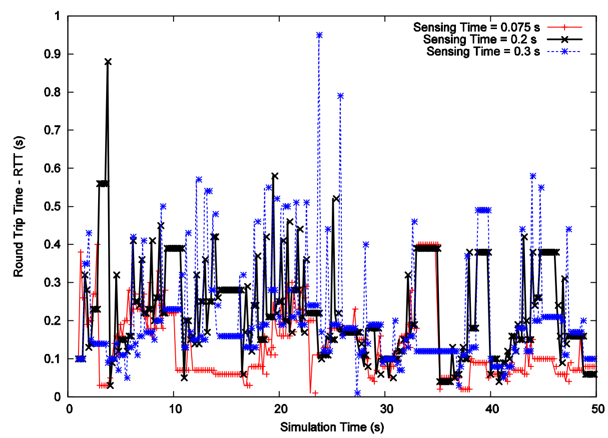

Figure 5. Sensing Cycle Analysis: RTT over Time.

Fig. 4 shows the CW size over simulation time, for three different settings of t_s, when T_p is fixed to 0.6s. When t_s is short (i.e. equal to 0.025s), the CW experiences frequent shrinking because of RTO events caused by PU interference. In the same way, frequent RTO events occur when t_s is long (i.e. equal to 0.3s), due to the inefficiency in channel utilization. In Figure 4, we also show the events of PU interference for t_s equal to 0.2s, i.e. the events where a PU starts transmitting on the channel currently used by CRU. As stated before, PU interference may cause packet losses at both sender and receiver side, so that RTO events are triggered. This is shown in Figure 4 by the fact that CW is collapsed to the value of 1 segment in proximity of PU mis-detection and interference events. Figure 5 shows the RTT index over simulation time, for the three configurations of t_s. As expected, the configurations with longer t_s experience higher variations of the RTT, due to the fact that a CRU consumes more time in sensing the channel than transmitting.

Considerations. The sensing interval plays a critical role in deciding the optimal end-to-end TCP throughput. A short sensing interval increases the risk of interfering with PU activities, while a long sensing interval increases the PU protection but reduces the transmission opportunities for CRUs. The transport layer should balance the tradeoff so that the throughput is maintained at the desired level while the PU interference is minimized. Since the sensing cycle is implemented at MAC/PHY layers, explicit feedback should be used to inform the transport layer when a channel sensing event is performed. Notifications of sensing events might be exploited to adapt the flow control mechanism in TCP, by freezing the current TCP state during sensing time intervals, or by reducing the sending rate, as proposed in [4].

5.2 PU Activity Analysis

Simulation Setup. In the PU Activity Analysis, we considered the single-hop topology, and we varied the α and β parameters regulating the average PU activity on each primary band/CRU channel. The sensing and transmitting time interval are equal to 0.2s and 0.6s, respectively. We considered the TCP throughput at application layer, defined in Section 5.

Simulation Results. Figure 6 shows the throughput of TCP NewReno as a function of the α (*x*-axis) and β parameters (*y*-axis). Based on values of α and β, we can distinguish among 4 different "regions" of PU activity:

- $\alpha \leq 1, \beta \leq 1$, (*Long-Term Activity Region*): in each spectrum band, there are long ON periods followed by long OFF periods.
- $\alpha \leq 1, \beta > 1$, (*High Activity Region*): in each spectrum band, there are long ON periods followed by short OFF periods.
- $\alpha > 1, \beta \leq 1$, (*Low Activity Region*): in each spectrum band, there are short ON periods followed by long OFF periods
- $\alpha > 1, \beta > 1$, (*Intermittent Activity Region*): in each spectrum band, there are short ON periods followed by short OFF periods.

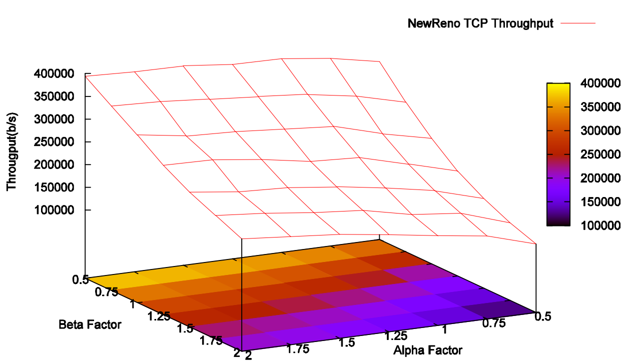

Figure 6. PU Activity Analysis: NewReno TCP Throughput

Not surprisingly, TCP performance are maximized when the CRUs have more possibility to access the licensed spectrum without interfering with the PU activity, i.e. in the "*Low Activity Region*" in Figure 6. When the PU activity is high in each spectrum band, the CRUs might need to frequently switch among the available channels, before finding a different vacant portion of the spectrum. Additional delay introduced by spectrum handoff might trigger RTO events and adversely affect the end-to-end performance, for the same reasons described in Section 5.1. As a result, TCP experiences the lowest throughput in the "*High Activity Region*". Moreover, Figure 6 shows that TCP performs better when there are long ON periods followed by long OFF periods (i.e. in the "*Log-Term Activity Region*") rather than when frequent ON/OFF switches occur (i.e. in the "*Intermittent Activity Region*"), in each primary band. This is due to the fact that when ON/OFF periods are very short a PU might arrive during transmitting period of CRUs, even if the channel was found free during sensing.

Considerations. When a PU is detected, the CRU should cease the current activity and move to another vacant portion of the spectrum. However, while spectrum sensing is periodic and has a well defined interval, the time taken to (i) search for a new channel on a different spectrum band, (ii) coordinate with the next hop neighbor to find a mutually acceptable channel and (iii) notify such information at upper layer, is of an uncertain duration, and depends on the parameters describing the PU activity (e.g. α, β). Frequent or long spectrum handoff events adversely affect the end-to-end performance. Again, network feedback might be used to notify the TCP source of the occurrence of spectrum handoff events.

5.3 Channel Heterogeneity Analysis

Simulation Setup. In the Channel Heterogeneity Analysis, we considered the basic single-hop topology, but we removed the assumption all CRU channels/primary spectrum bands have the same capacity. Without loss of generality, we will use channel *i* to refer to both spectrum band *i* and CRU channel *i*. To evaluate the impact of channel heterogeneity on TCP performance, we built a channel environment where the capacity of channel *i* is defined as follows:

$$C(i) = \begin{cases} \gamma\left(1+\frac{M}{2}\right) \ if \ (i\%2)=0 \\ \gamma\left(1-\frac{M}{2}\right) \ if \ (i\%2)=1 \end{cases} \quad (5)$$

for $0 \leq i < 5$. From Equation 5, it is easy to see that a switching operation from channel i to channel $i+1$ ($0 \leq i < 4$), produces an increment/decrement of capacity equal to γM. In the following, we define M as "*Channel Bandwdith Differentiation Factor*", because higher values of M correspond to higher difference in raw bandwidth between adjacent channels. We set γ equal to 1 Mb/s. The α and β parameters are both equal to 0.5. The sensing and transmitting time intervals are equal to 0.025s and 1.0s, respectively. We considered the TCP Efficiency, defined by Equation 4.

Simulation Results. Figures 7-8 show the TCP performance for the Channel Heterogeneity Analysis. Figure 7 shows the TCP Efficiency as a function of the M factor, for different TCP variants. All TCPs experience the highest efficiency for $M = 0.2$. Classical TCP is not able to dynamically adapt to bandwidth variation. As a result, the TCP Efficiency of all the TCP variants degrades when M increases. Simulation results presented here are also in accordance with results described in [7]. From Figure 7, we can also see that TCP SACK enhances TCP NewReno and TCP Reno, for all the value of M. Performance results (not shown here) demonstrate that TCP Vegas has the lowest TCP efficiency, for all the value of M. Figure 8 shows the CW size over simulation time on the *x*-*y* axes for the configuration with $M = 1.6$, i.e. when large bandwidth variation is experienced by CRUs after switching to adjacent channels. Moreover, we also show the available bandwidth for CRUs over simulation time on the *x-y2* axes. Figure 8 confirms that the CW of classical TCP is unable to correctly track the available bandwidth. This is caused by the scheme which regulates the CW size, which is based on the additive increase/multiplicative-decrease algorithm (AIMD). The AIMD algorithm combines linear growth of the CW size during the congestion avoidance phase, with an exponential reduction when a congestion takes place. In congestion avoidance phase, the CW is increased by one segment every RTT. Such a conservative approach reveals to be inadequate when a brief bandwidth increase is experienced by CRUs, because the spectrum opportunity is often lost before the CW has increased to half the segments that may be supported on the new channel (Figure 8). The AIMD algorithm does not take into account the events of bandwidth decrease too. In such cases, the CW is not reduced, but keeps growing till a RTO event occurs. Figure 8 shows that RTO events are triggered in correspondence of bandwidth reduction.

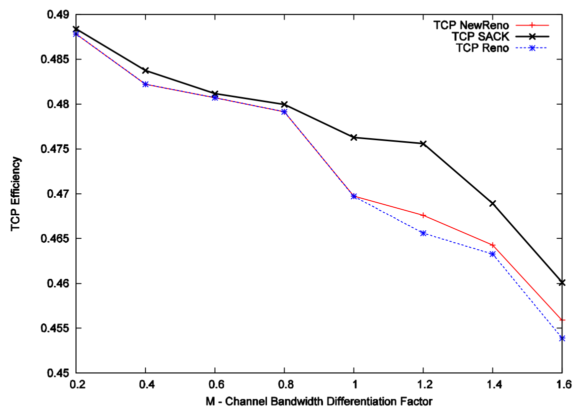

Figure 7. Channel Heterogeneity Analysis: TCP Efficiency.

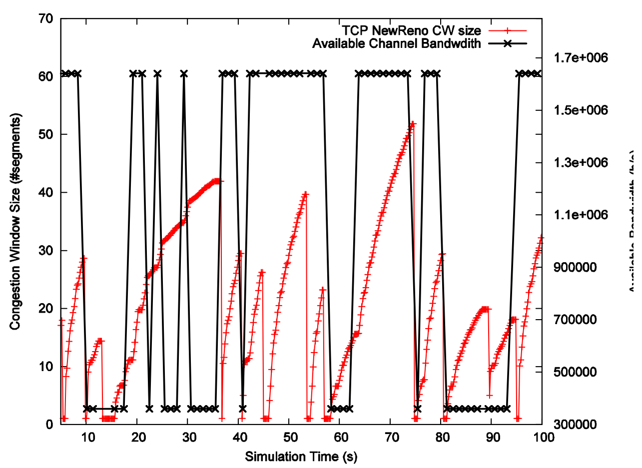

Figure 8. Channel Heterogeneity Analysis: CW size vs Available Bandwidth.

Considerations. TCP cannot effectively adapt to brief reductions/increases in channel capacity, due to the drawbacks of the AIMD algorithm. We believe that the CW size in TCP must be scaled appropriately to meet the new channel conditions. To this aim, TCP should be notified of the occurrence of a channel switching event by the link layer. Moreover, link layer metrics should be used to estimate the raw bandwidth of the new channel. In [4] a scale-bandwidth algorithm is proposed so that CW and RTT of the TCP source are adjusted after each channel bandwidth variation, in order to provide efficient utilization of the available spectrum resources.

5.4 Multi-hop Multi-flow Random Topology

In Section 5.1, 5.2, 5.3 we evaluated the impact of different CRAHNs factors on TCP performance, over a single-hop topology. In this section, we evaluate TCP performance over a generic random multi-hop topology. The goals of this analysis are twofold. First, we show the combined effect of all the CRAHNs characteristics described above on TCP performance. Second, we underline how such characteristics might produce different results than traditional characteristics of ad hoc networks (e.g. nodes' mobility), motivating the need for considering novel transport layer solutions for CRAHNs. We considered random topologies CRAHNs composed of 25 mobile CRUs over a simulation area of 1000x1000 m^2. All the CRUs move with uniform speed v (2.5 m/s<v<4.0 m/s), and an average pause time of 3s. We varied the number of active FTP-New Reno TCP connections, from 2 up to 8. We used the following parameters for the cognitive environment: α =0.5, β =0.5, t_s =0.2, T_p =1s, M =0.2. In Figure 9 and 10, we evaluated the performance of TCP by using two different simulation models:

- CRAHN Model ON: We used the CRAHN model described in Section 4, with the cognitive parameters stated above. We considered the impact of sensing time, PU activity and channel bandwidth variation on the end-to-end performance.
- CRAHN Model OFF: We did not simulate the impact of CRAHN characteristics. At MAC Layer, we used the traditional 802.11 MAC DCF scheme, without the modification described in Section 4. Basically, we performed an evaluation of TCP in a traditional mobile ad hoc scenario, as addressed in [6].

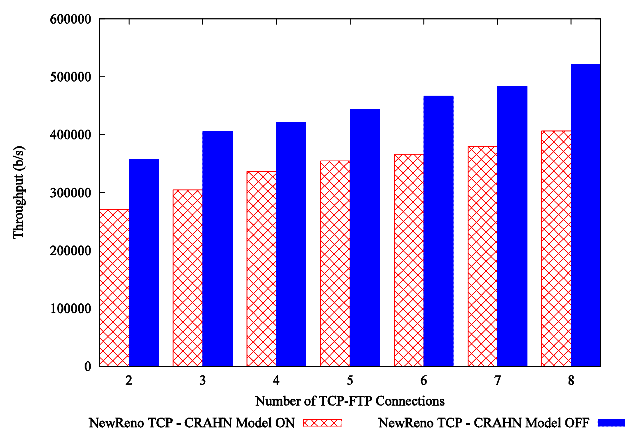

Figure 9. Multi-hop Topology: End-to-end Throughput.

Figure 9 shows the TCP throughput as a function of variable number of active connections. As expected, the throughput increases when more connections are added in the simulation scenario, for both the configurations. However, the configuration "CRAHN model OFF" experiences higher throughput than the configuration "CRAHN model ON", and the performance difference increases with the number of active connections. Figure 10 provides an orthogonal view of system performance, by showing the average number of RTO events for seconds as a function of the number of active connections. As expected, RTO events increase with congestion, when more flows are added into the network. However, additional RTO events are also triggered by link failures and route reconfiguration caused by the nodes' mobility [6]. To this aim, some TCP-variants proposed for ad hoc networks exploit network feedback, which allows to detect packet losses caused by mobility handoff or by network congestion [8]. This approach might not work well over CRAHNs, where additional RTO events might be triggered by the delay caused by spectrum handoff and by the sensing cycle, and by brief variations in channel capacity (Figure 10). As a result, novel transport solutions should be designed for CRAHNs.

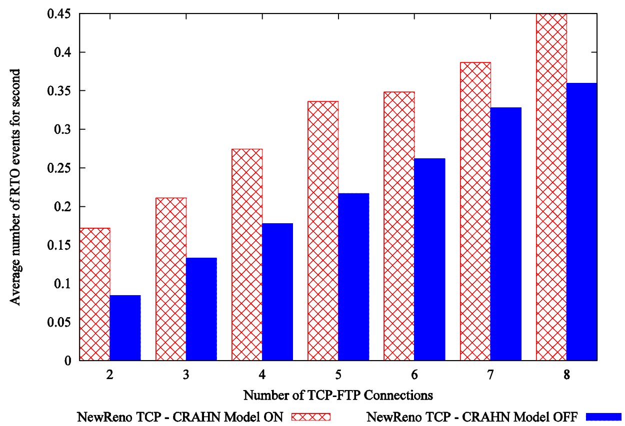

Figure 10. Multi-Hop Topology: Avg. RTO events for seconds.

6. CONCLUSIONS

In this paper, we addressed performance evaluation of TCP over Cognitive Radio Ad Hoc Networks (CRAHNs). To this aim, we presented an extension of the NS-2 tool for the modeling and simulation of CRAHNs. Our extension provides accurate modeling of PUs activities and of CRUs spectrum management functionalities, including spectrum sensing, decision and mobility schemes. Moreover, it provides facilities to support cross-layer information sharing among network protocols at different layers of the protocol stack. By using the revised NS-2 tool, we evaluated the performance of TCP over CRAHNs. We studied the impact of specific characteristics of CRAHNs over end-to-end performance, such as the sensing cycle, the interference caused by PUs and the channel heterogeneity. Simulation results show that sensing time interval and type of PU activity play a critical role in deciding the TCP performance. Moreover, simulation results highlight that transport protocols proposed for traditional wireless ad hoc networks might not work well over CRAHNs. Since transport layer is still an explored research area for CRAHNs, studying and understanding problems of classical TCPs in the cognitive environment is fundamental to design novel transport protocol solutions for CRAHNs. Future works in this direction will involve the extension of the revised NS-2 simulation tool, the implementation of MAC and routing protocols for CRAHNs in the current framework, and further evaluation of transport layer issues over CRAHNs.

7. ACKNOWLEDGMENTS

This work is supported by Italian MIUR funds (project PRIN-2006 NADIR: Design and Analysis of distributed, and QoS-aware protocols and algorithms for Wireless Mesh Networks).

8. REFERENCES

[1] FCC, ET Docket No. 03-222 Notice of proposed rule making and order, December 2003.

[2] 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.

[3] I. Akyildiz, W.Y. Lee, K. R. Chowdhury. CRAHNs: Cognitive Radio Ad Hoc Networks, to appear in Ad Hoc Networking, 2009.

[4] K. R. Chowdhury, M. Di Felice, I. Akyildiz. TP-CRAHN: A Transport Protocol for Mobile Cognitive Radio Ad Hoc Networks. in Proc. of INFOCOM'09, Rio de Janeiro, pp. 2482-2490.

[5] 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.

[6] G. Holland and N. H. Vaidya. Analysis of TCP Performance over Mobile Ad Hoc Networks. in Proc. of ACM MOBICOM'99, Seattle, pp. 219-230.

[7] 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'07, Dublin, pp. 486–495.

[8] J. Liu and S. Singh. ATCP: TCP for mobile ad hoc networks. IEEE Journal on Sel. Areas of Comm., 19(7):1300–1315, 2001.

[9] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. Internet Engeneering Task Force, Request for Comments (Experimental) 2582, Apr. 1999.

[10] L. Brakmo and L. Peterson, TCP Vegas: end to end congestion avoidance on a global Internet, IEEE J. Selected Areas Communication, 13(8), pp. 1465-1480, 1995.

[11] M. Mathis, J. Mahadavi, S. Floyd and A. Romanow. TCP selective acknowledgment options. Internet Engeneering Task Force, Request for Comments (Proposed Standard) 2018, Oct. 2006.

[12] H, Abrahamsson, O. Hangsand and I. Marsh. TCP over high speed variable capacity links: A simulation study for bandwidth allocation. in Proc. of PIHSN'02, Salern, pp. 117-129.

[13] W. Y. Lee and I. Akyildiz, Optimal Spetrum Sensing Framework for Cognitive Radio Networks. in IEEE Transactions on Wireless Communications, 7(10) , pp. 3845-3857, 2008.

[14] K. Xu, M. Gerla, L. Qi and Y. Shu. Enhancing TCP fairness in ad hoc wireless networks using neighborhood red. in Proc. of MOBICOM'03. San Diego, pp. 16-28.

[15] S. E. ElRakabawy, A. Klemm and C. Lindemann. TCP with adaptive pacing for multihop wireless networks. in Proc. of MobiHoc'05, Urbana-Champaign, pp 288-299.

[16] The network simulator - ns-2. http://www.isi.edu/nsnam/ns