Investigation of TCP Protocols in Dynamically Varying Bandwidth Conditions

Fan Zhou[1], Abdulla Al Ali[2], and Kaushik Chowdhury[3]

[1] Northeastern University, Boston MA 02110
zhou.fan1@husky.neu.edu
[2] Qatar University, Qatar
abdulla.alali@qu.edu.qa
[3] Northeastern University, Boston MA 02110
krc@ece.neu.edu

Abstract. Cognitive radio (CR) networks experience fluctuating spectrum availability that impacts the end to end bandwidth of a connection. In this paper, we conduct an extensive simulation study of three different window-based TCP flavors- NewReno, Westwood+, and Compound, each of which has unique methods to determine the available bandwidth and scale the congestion window appropriately. These protocols also differ in their respective sensitivities to the metrics of round trip time, loss rate, residual buffer space, among others. These metrics exhibit divergent behavior in CR networks, as compared to classical wireless networks, owing to the frequent channel switching and spectrum sensing functions, and this influences the choice of the TCP protocol. Our ns-3 based simulation study reveals which specific rate control mechanism in these various TCP protocols are best suited for quickly adapting to varying spectrum and bandwidth conditions, and ensuring the maximum possible throughput for the connection.

Key words: TCP evaluation, dynamic bandwidth, cognitive radio network

1 Introduction

Cognitive radio (CR) networks can potentially overcome the limitations of preassigned and static channelization by identifying unused spectrum, possibly in licensed bands, as well as switch the operation into these channels on a need basis [1]. While the research community has invested heavily in the design of the physical and link layers of the protocol stack, there is need for a systematic study of the end to end operation in the upper layers of the protocol stack, such as the transport layer. This is especially important as a desirable end-user experience, reliability of the data delivery between source and destination, and complete spectrum utilization within the short-time availability of the licensed spectrum can only be achieved through optimizations at the transport layer. In this paper, we report results from a methodical simulation study at the transport layer, focusing on three different variants of TCP that are window-based, i.e., the effective window containing the packets that may be transmitted by the source at any given time is altered depending upon observed delays, losses, bandwidth estimation and buffer space availability.

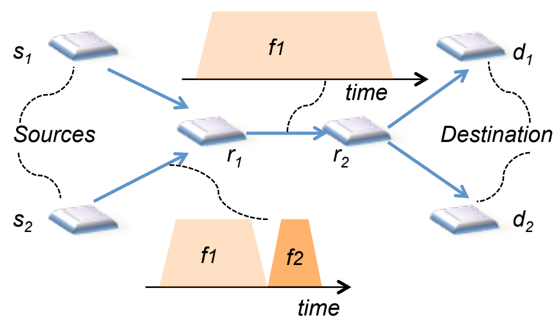

Fig. 1. End to end connection and varying spectrum in intermediate links

TCP has been extensively studied and analyzed over the past decades. Given its widespread adoption, identifying which of its specific rate control mechanisms are best suited in CR networks will yield useful insights in tuning them for these challenging environments. This is the first step towards determining whether minor changes to existing protocol flavors are sufficient to adapt TCP, or if an entirely new class of window-based approaches are needed to achieve satisfactory operation. We conduct the first comprehensive study of three different TCP flavors with this aim, focusing on high-bandwidth varying situations. Our previous works have explored enhancing TCP NewReno by leveraging extensive cross-layer and intermediate-node information [2] as well as extending equation-based protocol called TFRC [3]. However, both these approaches have limitations. First, the transport layer is envisaged to operate end to end from a classical networking viewpoint, and thus, enforcing dependence on the choices made at the lower layers, or mandating feedback from other intermediate nodes, brings about a radical change in the commonly accepted end-to-end paradigm, and results in a loss of generality. Similarly, TFRC is shown to be TCP-friendly in its classical incarnation, while our proposed triggers for the rate control equation in the modified TFRC-CR cause it to lose this important characteristic (though it gives higher network throughput and protection to licensed users). Thus, TFRC-CR can no longer fairly exist with other TCP flows, which limits deployment opportunities. As a result, in this study, we examine other well known and unmodified TCP flavors that are compliant with the fairness criterion, as well as the end-to-end paradigm of the transport layer, and we determine which of these are best suited for CR networks 'as is'.

The simulation scenario that we will refer to in the subsequent discussion is shown in Fig. 1. Multiple sources (s_1, s_2) inject traffic into a chain network that may have several overlapping intermediate forwarding nodes (r_1, r_2). The connections end at respective destinations (d_1, d_2), which send back the acknowledgements (ACKs). The various links can be susceptible to high priority licensed user activity. All nodes perform synchronized spectrum sensing, and the licensed user coverage range is large enough to impact all the nodes of the chain. Thus, over time, the nodes at either end of a given link may switch to a different channel $(f_1, \ldots, f_n)$ if the current channel is rendered unusable, though the particular choice of the channels for the intermediate nodes is not known to the source. We assume that these channels vary greatly in terms of their upper and lower frequency bounds, as well have different amounts of existing traffic within them.

2 Discussion of TCP Protocols Under Study

In this section, we briefly describe the three different TCP protocols that we investigate in CR scenarios, i.e., (i) TCP NewReno, (ii) TCP Westwood+, and (iii) TCP Compound, mainly stressing on the features that will be useful for the dynamic spectrum environments.

2.1 TCP NewReno

This is the classical TCP version, and we use this to benchmark performance of the other protocols. The congestion window (*cwnd*) doubles in the *slow start* phase, and then increases linearly in the *congestion avoidance* phase. Timeouts caused by missing ACKs reduce the *cwnd* to 1, while triple duplicate ACKs cut the *cwnd* to half of its original value at the time of detecting the congestion. NewReno continuously updates its average round trip time (RTT) by maintaining a moving window over the last few sample RTT values, and thereby adjusts its estimate smoothly over time. We point the reader to [4] for further discussion on this protocol.

2.2 TCP Westwood+

Westwood+ deviates from NewReno in the sense that triple duplicate ACKs don't force the *cwnd* to half, but to $max(\frac{cwnd}{2}, EBW * RTT_{min})$ that allows the source to saturate bottleneck link, while draining the bottleneck queue buffer simultaneously [5]. EBW is the estimated bandwidth and RTT_{min} is the minimum RTT measured. We note that RTT_{min} refers to the time when there is no congestion in the network. Thus, the product of bottleneck link bandwidth and minimum RTT reflects the total capacity of the link. By reducing *cwnd* to $EBW * RTT_{min}$, packets in the buffer will be drained while the link capacity is kept full. Two features of this protocol make it attractive for use in DSA networks: First, for occasional packet losses that are channel-state induced, the estimated bandwidth would not change much. On the other hand, when spectrum changes happen leading to a massive increase or decrease in the available bandwidth, the EBW will change, which in turn allows for a more accurate setting for *cwnd*. Clearly, as the computation of EBW is a critical factor, the protocol uses information about the amount of data received during a certain period of time by passing the samples through a low pass filter using the so called *Tustin* approximation. This gives the protocol robustness against delayed ACKs. This feature is especially useful when licensed or priority users (PUs) of the spectrum cause interruptions in the connection.

2.3 TCP Compound (CTCP)

This protocol diverges from NewReno in the way the *cwnd* changes in the congestion avoidance stage [6]. If $cwnd \geq 38$, a new parameter called as the delay window *dwnd* is added to *cwnd*, i.e., $cwnd = cwnd + dwnd$. The *dwnd* is itself defined in terms of a variable called as $diff$. Here, $diff$ is the difference between estimated current capacity and the theoretical capacity (when there is no congestion). It is computed by estimating all the buffered packets in the network, similar to a different protocol called TCP Vegas. When the latter is less than a threshold r, the *dwnd* increases exponentially to quickly utilize the network resources. In network with high bandwidth-delay product, this increases the efficiency of bandwidth utilization. When $diff$ is larger than r, it suggests increasing accumulation of packets in the buffer. Hence, the *dwnd* is decreased linearly until it reaches zero, and the performance of CTCP is degraded to that of TCP NewReno. This feature is critical in CR networks, as periodic spectrum sensing disconnects the connection temporarily, leading to a build up of packets from the source to the node immediately before the sensing node. Finally, if a packet loss is detected by triple duplicate ACK, *dwnd* is set to a value to make sure the window size is decreased by a factor of $(1-\beta)$. Clearly, the performance of CTCP is closely related with the choice of parameter values. For example, a balance must be struck to prevent the changes from being too aggressive or too conservative in *dwnd*. A set of recommended values of parameters is obtained in [6] through empirical studies.

3 Network Setup for The Evaluation Study

In this section, we describe the topology and simulation setup for CR network. All simulations are conducted through a packet level simulation in the open source ns3 simulator.

- *Traffic:* We assume a saturated scenario, where senders always have data to send. We inject new packets in the sending buffer as long as it is not completely full.
- *Topology:* We use the network topology as shown in Fig. 1. There are two senders $\{s_1, s_2\}$, two destinations $\{d_1, d_2\}$, and two intermediate routing nodes $\{r_1, r_2\}$, which we refer to as 'routers' to simplify the discussion. The link between first and second router is the bottleneck link with longer delay and lower bandwidth.

– *Bandwidth:* The bandwidth in the CR network is influenced by two factors: spectrum sensing and the activity of PUs. The transmission in either case is paused for a short period of time. Following this pause, the node might choose a new channel with different bandwidth. This may cause a sudden transition of the nodes into either a new channel with the same bandwidth (we call this as the *fixed* scenario where all channels are similar) or widely different bandwidth (we call this as the *varying* scenario where channels can be of unequal width that we select uniformly from a pre-decided range). Note that we assume only the bottleneck router performs spectrum sensing and switching functions to clearly demonstrate the behavior of the TCP protocols.
– *Spectrum sensing:* We assume the bottleneck router senses the spectrum for 0.1s, in intervals of 5s. During spectrum sensing, the transmission is temporarily interrupted, and the router cannot receive nor send any packet. After spectrum sensing, the node might stay on original channel or use a new channel with different bandwidth, as we have discussed above.
– *Influence of PUs:* In a CR network, the transmission might be interrupted by sudden occurrence of the PU. We model the arrival of PUs as a Poisson process, and hence, the time interval between successive arrivals follows an Exponential distribution. We vary the mean time to model different extents of the interference and consequently, the network interruption.

4 Simulation Results and Observations

4.1 Impact of fixed and varying bandwidth

Fixed bandwidth In this simulation, we study the performance of TCP in fixed bandwidth scenarios. We begin by considering a classical wireless network without CR functionality (classical) and plot real-time throughput of flow 1 (s_1, r_1, r_2, d_1) using TCP NewReno in Fig. 2. We see that the interruption caused due to periodic sensing in the same connection (CR with sensing) severely reduces the network throughput. The maximum throughput here is about 1.8Mbps, which is less than half of the minimum throughput classical case. We see a similar trend in Westwood+ and CTCP. The implication here is that even if the interruption time is 2% of the transmission time, the frequent timeout retransmit events continue to cause severe degradation to the performance to TCP.

Fig. 3 presents the throughput of three protocols for different values of the bottleneck channel bandwidth (though all channels have the same bandwidth) . We notice that the throughput of NewReno does not increase much with higher bandwidth availability. This is because NewReno's performance is limited by the slow rate of increase of the *cwnd* in the congestion avoidance stage. From Fig. 4, we see the three traces of *cwnd* are similar in trend, regardless of different bottleneck bandwidth selections.

Interestingly, Westwood+ outperforms NewReno and CTCP when bandwidth is less than 10Mbps. This is mainly because Westwood+ adjusts its slow

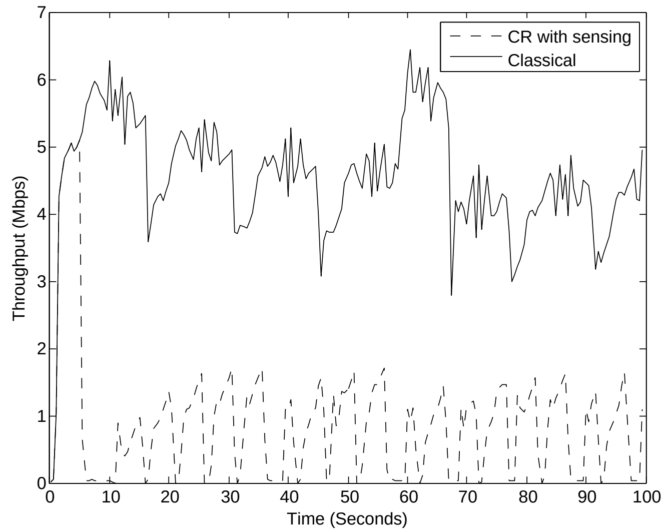

Fig. 2. Real-time throughput of NewReno in cognitive radio network and none-cognitive radio network

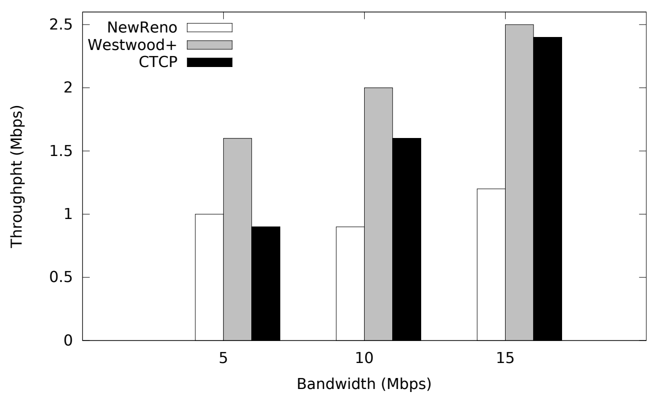

Fig. 3. Average throughput of NewReno, Westwood+, CTCP in fixed bandwidth model

start *cwnd* according to the product of estimated bandwidth and RTT_{min}. This is very effective in a CR network, as the estimated bandwidth is not immediately impacted by an occasional packet loss, and the *cwnd* is not directly halved. Moreover, after a channel switch, especially, if the new channel has lesser bandwidth than the previous one, the *cwnd* of TCP Westwood+ quickly sets itself to the new bandwidth. Unlike NewReno, it does not have to ramp up back from $\frac{cwnd}{2}$.

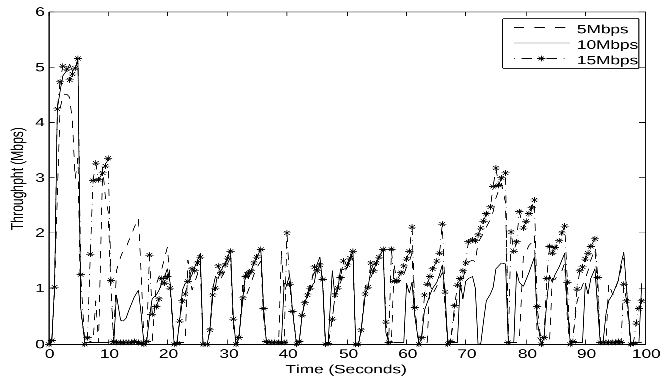

Fig. 4. Real-time throughput of NewReno for different channel bandwidths

We observe that CTCP has comparable throughput with Westwood+ when the channel bandwidth availability increases to 15Mbps. This is because the improvement of throughput in CTCP mainly comes from the addition of the *dwnd* in congestion avoidance stage. This mechanism is less effective in small bandwidth connections, as *cwnd* needs to be greater than the threshold of 38 to trigger inclusion of *dwnd*. Therefore, when spectrum switching allows the connection to enter into channels with much higher bandwidth, it is better to use CTCP as this results in faster rise of the end to end throughput. However, in low bandwidth conditions, CTCP performs similar to NewReno.

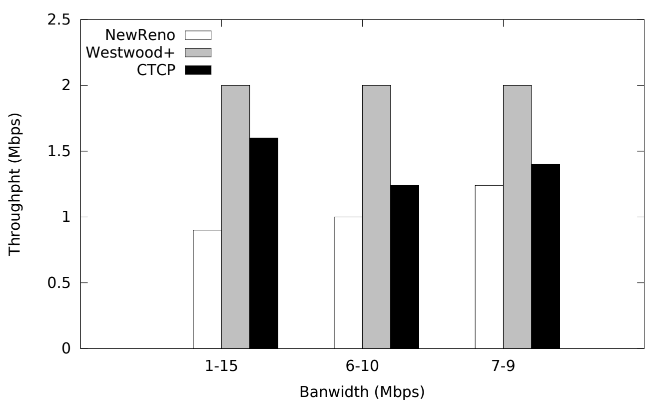

Fig. 5. Average throughput of NewReno, Westwood+, CTCP in varying bandwidth channels

Varying bandwidth In this scenario, we study the ability of TCP protocols to get adapt to different channels that have varying bandwidth. We assume that after each spectrum sensing stage, the node randomly picks a new channel. The bandwidth of this new channel is uniformly distributed between three ranges: [1Mbps,15Mbps], [6Mbps, 10Mbps] and [7Mbps, 9Mbps]. While the average bandwidth in these three ranges is the same, the wider the range, more is the diversity within the available bandwidth set.

Figure 5 shows the first flow's, i.e., (s_1, r_1, r_2, D_1), average throughput using the three TCP protocols in a multichannel environment where channels are picked from different varying ranges of bandwidth. We observe that the performance of three protocols is surprisingly stable. The throughput does not change much even if the available bandwidth fluctuates drastically. However, this stability is a direct outcome of their low efficiency in utilizing bandwidth in a CR network. It can be intuitively seen in Fig. 6 for the best performing protocol (Westwood+).

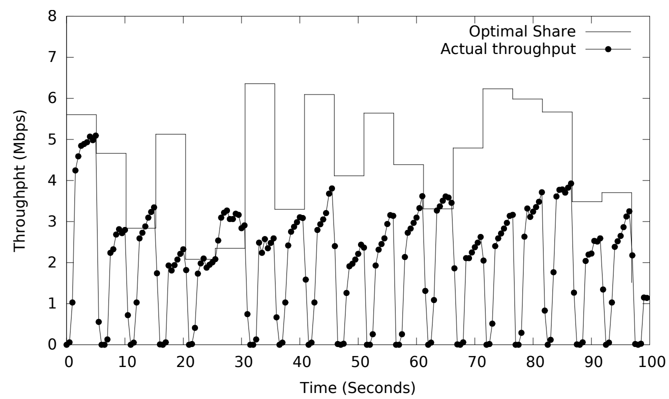

Fig. 6. Comparison of real-time throughput of Westwood+ and its optimal share (optimal share is Total Bandwidth/2)

Note that the recovery of *cwnd* after a timeout event in Westwood+ is very fast (i.e., much faster than NewReno and Compound). However there still exists an upper limit on the throughput that Westwood+ can achieve. This large gap between the optimal throughput and actual throughput shows that there is a large proportion of bandwidth wasted. This implies that for a CR network, the bottleneck in the performance may lie not in the bandwidth being unavailable for the connection in the chosen channel, but in the less-than-optimal rate of increase of *cwnd* in the period between two transmission interruption events.

4.2 Spectrum sensing pattern

In this simulation, we investigate the impact of different spectrum sensing durations. The transmission (On) period time are set as 5s, 10s, 15s and spectrum sensing time (Off) durations are 0.05s, 0.1s, 0.2s and 0.4s. We perform an ex-

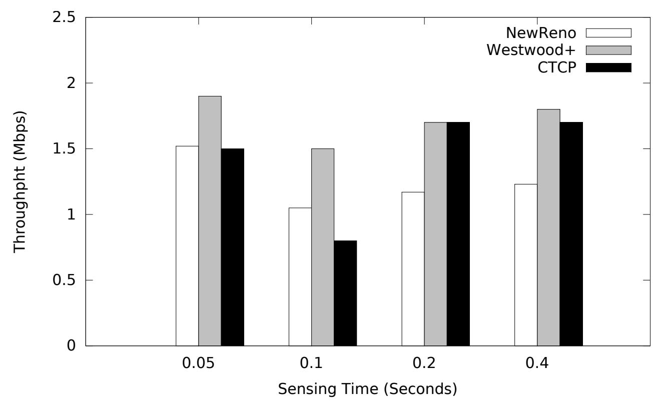

Fig. 7. Average throughput of NewReno, Westwood+, CTCP with different sensing times

haustive set of simulation trials matching for every pair of On and Off time instances drawn from the above set.

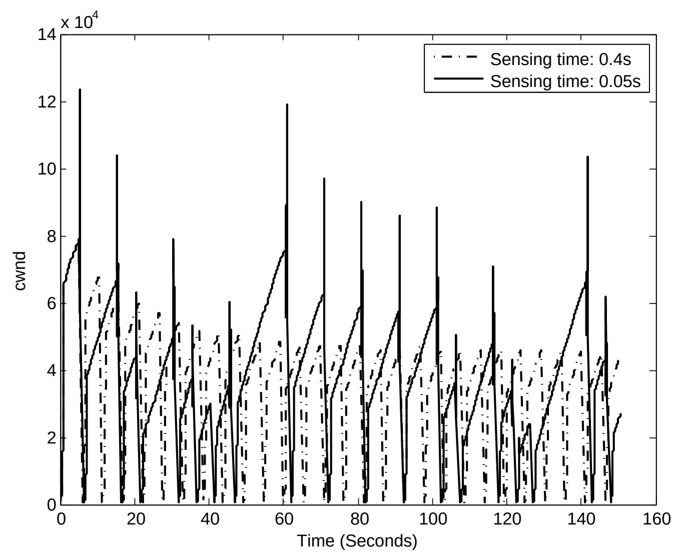

Fig. 8. *cwnd* of Westwood+ with different sensing times

Figure 7 shows the performance of each protocol with different Off time and fixed On time. Intuitively, increasing the sensing time will prolong the disconnection period of connection, and thus, we expect that the throughput will dramatically fall. However from Fig. 7, we see that this is not always the case. While the throughput drops sharply as sensing time increases from 0.05s to 0.1s, this trend is reversed when sensing time increases from 0.1s to 0.4s. To understand this phenomenon, we trace the behavior of *cwnd* in Westwood+, as shown in Fig. 8.

We find that transmission is not affected by the disconnection in the network when sensing time is small, say around 0.05s. As a result, there is no duplicate ACKs or timeout event caused by packet loss during 45s to 60s in the simulation time. In other words, the interruption of the transmission at the affected link is completed hidden from the source node. Here, the bottleneck router alternatively inserts several packets from sender 1 and sender 2 into its own sending buffer queue. So correspondingly the senders will experience alternate bursty incoming ACKs and then this phase is followed by a phase of no ACKs. When there is no ACK coming in, the sender does not send any packet. When the periodic spectrum sensing coincides within the 'no ACK' periods, the sender remains oblivious to the reason why the ACKs are sparse.

When the sensing time crosses a certain threshold, say 0.1s, then timeouts happen after the spectrum sensing. In this situation, the throughput depends upon how fast *cwnd* can recover from the impact of timeout. Next, we study the performance of protocols with different On time between two spectrum sensing events. We see that there is a jump in the throughput for all three protocols when the On time increases from 5s to 10s. However, the throughput does not change to a similar extent when On time increases from 10s to 15s. We analyze the reason of this behavior in the next scenario.

4.3 Influence of PU

Here, we simulate the influence of random interference of the PU on the performance of TCP. After the PU appears, the affected router must do a spectrum switch. For simplicity, we assume that the time required to do this is fixed (0.1s). Also, to demonstrate the results with clarity, we use the fixed bandwidth model. At a high level, the influence of the PU is similar to that of spectrum sensing: the only difference is that the On time between two transmission interruptions is now an exponential random variable rather than a fixed constant.

Fig. 9 shows average throughput of flow 1. Overall, all three TCP protocols benefit from larger mean inter-arrival time λ. In addition, here we observe the similar phenomenon as in experiment B. That is for NewReno and Westwood, there is a sharp increase in throughput when the λ increase from 5s to 15s. However the throughput does not change much when λ increases from 15s to 20s. Note that for TCP Compound, the throughput has a stable increase with larger values of λ.

The increase of throughput between 5s to 15s correlates with fewer timeout events as these are PU-free durations. However for NewReno and Westwood+, as long as the transmission time is long enough for them to get out of slow start stage, further increase in throughput is throttled by the slow rate of change of *cwnd* (i.e., linear change) in the congestion avoidance stage. On the other hand, CTCP can benefit more from the longer uninterrupted transmission durations

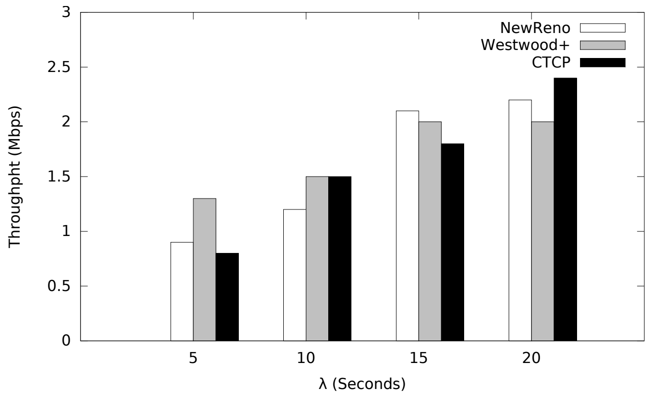

Fig. 9. Average throughput of NewReno, Westwood+, CTCP for various λ

as it's *cwnd* grows faster in the congestion avoidance stage than NewReno and Westwood.

5 Discussion and Conclusion

While Westwood+ gives quick recovery of the *cwnd* after a timeout, its spectrum utilization is severely impacted by the slow increase of the *cwnd* in the congestion avoidance stage. In contrast, CTCP can utilize bandwidth better through the additive effect of the variable *dwnd* in the congestion avoidance stage. Also if the *cwnd* is not high enough, which may occur in low bandwidth channels or high PU activity channels, CTCP will degrade to NewReno, as it cannot reach the threshold to activate the addition of *dwnd*.

From the simulation study, we identify two ways to boost the performance of TCP protocol in a CR network. The first is to increase *cwnd* aggressively during the limited continuous transmission time. For example, combining the behavior of Westwood+ to set the slow start threshold, and then switching over to CTCP's fast window size increase during congestion avoidance stage could enhance the utilization. A different approach is to completely hide the CR-related activity inside the connection from the source node without allowing any intermediate node feedback. We shall pursue both these directions as part of our future work.

Acknowledgment

This work was supported in part by the Office of Naval Research under grant number N000141410192.

References

1. I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, CRAHNs: Cognitive radio ad hoc networks. In *Elsevier Ad Hoc Networks*, vol. 7, no. 5, Jul. 2009.
2. K. R. Chowdhury, M. Di Felice and I. F. Akyildiz, TCP CRAHN: A Transport Control Protocol for Cognitive Radio Ad Hoc Networks. In *IEEE Trans. on Mobile Computing*, vol. 12, no. 4, pp. 790-803 Apr. 2013.
3. A. Al-Ali and K. R. Chowdhury, An Equation-based Transport Protocol for Cognitive Radio Networks. In *Elsevier Ad Hoc Networks*, vol. 11, no. 6, pp. 1836-1847, Aug. 2013.
4. S. Floyd and T. Henderson, The NewReno Modification to TCP's Fast Recovery Algorithm. Internet Engineering Task Force , Request for Comments (Experimental 2582), Apr. 1999.
5. 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 the ACM Conf. on Mobile Computing and Networking*, pp. 287-297, 2001.
6. K. Tan, J. Song, Q. Zhang, and M. Sridharan, A Compound TCP Approach for High-speed and Long Distance Networks . In *MSR-TR-2005-86 (techreport) Microsoft Research*, Jul. 2005.