Wireless TCP Enhancements Using Multi-level ECN

Mukundan Sridharan, Arjan Durresi, Chunlei Liu, Raj Jain

Raj Jain is now at Washington University in Saint Louis, jain@cse.wustl.edu http://www.cse.wustl.edu/~jain/

ABSTRACT

Due to the increasing demand for mobility among the Internet users, there is a urgent requirement to identify and solve the deficiencies in the wireless domain. One such urgent problem is the poor performance of TCP over wireless links. TCP still being the only protocol used in the Internet for reliable transfers, the assumptions made by TCP in the wired domain are not valid in the wireless domain. To enhance the performace of the TCP in the wireless domain, we need to differentiate the 'congestion loss' from the 'wireless loss'. We find that the previous attempts in this direction make unjustified demands from the network or the accuracy of the schemes are inadequate. We are convinced that reliable transport is a end-to-end semantic and other network components should not be burdened with this work. In this paper we propose a scheme called the 'Source-Centric Congestion Filtering', based on the MECN protocol, which tries to differentiate the losses based on the network feedback. Our simulations using the NS-2 simulator shows that our protocol has very less percentage of error and performs better than most of the other end-to-end TCP variants.

Keywords: Wireless, MECN, Wireless TCP, ECN, Filtering, Congestion Control

1. INTRODUCTION

The Transmission Control Protocol (TCP) is the dominant transport mechanism for reliable data transfer over the Internet. Internet is growing in size and becoming increasingly heterogeneous and the assumption that packet drops in network occur mainly due to congestion is no longer valid, as the wireless and satellite link are prone to transmission errors. Since TCP treats all packet loss as congestion loss and decreases the 'congestion window', the performance of TCP suffers heavily over wireless link. Studies have shown that the performance of TCP can be improved if the cause of a packet loss is identified [1] .In connections over a wireless links, packet losses are mainly due transmission errors or handoffs. Hence by differentiating a 'congestion loss' from a 'wireless transmission loss', TCP source can be made to refrain from the costly 'congestion control' measures. When wireless losses are retransmitted end-to-end, regular TCP algorithm treats the packet loss as an indication of congestion and thus reduces the congestion windows size. Unnecessary reductions cause big performance degradation because the effective transmission rate is cut to half.

One class of solutions [2,3] , requires the base stations to do some house keeping and retransmit or identify, wireless drops. Another solution, is to implement some kind of a link layer retransmission sceheme, which would be aware of transport layer losses [3,4] . Split connection or I-TCP [5]splits the connection into two TCP one in wired domain and one for wireless. Explicit Loss Notification (ELN) [6] is a general mechanism by which the cause of the packet loss can be communicated to the TCP sent doesnt specify how the cause of the loss is identified. It is generally used with snoop [2] .Other schemes [7]have been suggested in the past for this problem, using the end-to-end sematic, but the accurracy of such schemes are found to be unsatisfactory . The schemes that have been suggested either place some burden on other network components or voilate the end-to-semantic. Hence the search for a better scheme continues. A survey of different schemes proposed and their pitfalls are discussed in .[1] We believe that reliable transport, is a end-to-end semantic and the base stations should not be burdened

Further author information: (Send correspondence to Arjan Durresi.)
Mukundan Sridharan: E-mail: sridhara@cis.ohio-state.edu
Arjan Durresi: E-mail: durresi@cis.ohio-state.edu, Telephone: (614)6885610, Fax:(614) 2922911
Chunlei Liu: Email: cliu@cis.ohio-state.edu
Raj Jain: Email: jain@cis.ohio-state.edu

with this responsibility. Hence in our approach no special functionality of the base station is assumed and we try to solve the problem, based on a end-to-end approach.

The scheme presented here is based on MECN[8] , which is an extension to ECN [9]and RED [10] . The idea is to differentiate the 'congestion loss' and 'wireless loss', based on the feedback received from the network, in the form of congestion marks. If a loss is found out to be a 'wireless loss', the packet is retransmitted and the congestion window is not reduced.If a loss is found out to be a 'congestion loss', the level of congestion is estimated and based on the congestion level, the amount by which the 'congestion window' should be reduced is calculated and the lost packet is retransmitted. The rest of the paper is organized as follows. In Section 2, we give a brief introduction to the MECN scheme. The proposed solution is explain in Section 3. The possible mistake scenarios and error percentages are discussed in Section 4. In Section 5, the simulations done using *ns* [11] simulator, the simulation configuration, results and analysis are presented. In Section 6, the conclusion of study are presented.

2. BRIEF INTRODUCTION TO MECN

2.1. Marking bits at the router for MECN

MECN [8] uses the two bits that is being specified for the use of ECN [12] , in the IP header (bits 6 and 7 in the TOS octet in Ipv4, or the Traffic class octet in Ipv6), to indicate four different levels of congestion, instead of the binary feedback provided by ECN. The non ECN-capable packets are identified by '00', and it uses the other combinations to indicate three different levels of congestion and with packet-drop, four different levels of congestion is indicated and appropriate action could be taken by the source TCP depending on the level of congestion. The MECN packet marking/dropping policy is shown in Figure 1. If the size of the average queue is in between min_{th} and min_{th}, there is incipient congestion and the ECN bits are marked as '10' with a probability p_1. If the average queue is in between mid_{th} and max_{th} , there is moderate congestion and the ECN bits are marked as '11' with a probability p_2. If the average queue is above the maxthresh all packets are marked.

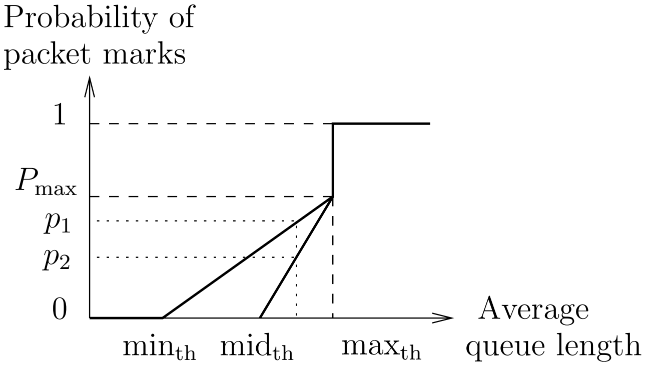

Figure 1. Packets marking scheme for MECN

2.2. Feedback from Receiver to Sender

The receiver reflects the bit marking in the IP header, to the TCP ACK. Since we have three levels of marking instead of 2-level marking in the traditional ECN, we make use of 3 combination of the 2 bits 8, 9 in the reserved field of the TCP header and the other combination used by the source has to indicate that the congestion window reduced.

2.3. Response of TCP Source

The MECN source reaction can be summarized as:

- When there is a packet-drop the cwnd is reduced by $\beta_3 = 50\%$. This done for two reasons: First, a packet-drop means severe congestion and buffer overflow and some severe actions need to be taken. Second, to maintain backward compatibility with routers which don t implement ECN.
- For other levels of congestion, such a drastic step as reducing the cwnd as half is not necessary and might make the flow less vigorous. When there is no congestion, the cwnd is allowed to grow additively as usual. When the marking is 10 (incipient congestion), cwnd is decreased by β_1 %. When the marking is 11 (moderate congestion) the cwnd is decreased multiplicatively not by a factor of 50% (as for a packet drop), but by a factor β_2% less than 50% but more than 1.

3. SOURCE-CENTRIC CONGESTION FILTERING USING WMECN

The scheme proposed here, called the 'Wireless MECN' or WMECN, is based on a modification of the Multi-level Explicit Congestion Notification. The basic idea is to keep a history of the congestion feedbacks, received from the network at each source. Whenever a packet loss is detected at the source, either by dupacks or timeout, a procedure is triggered, which determines, wheather the loss is a 'congestion loss' or a 'wireless loss', based on the information about the past congestion feedbacks, available at the source. Even though the idea is not new and several schemes use, the idea of 'explicit feedback' to differentiate the losses, the implementation of the idea is what gives this scheme, the advantage.

3.1. Marking of bits at the Router WMECN

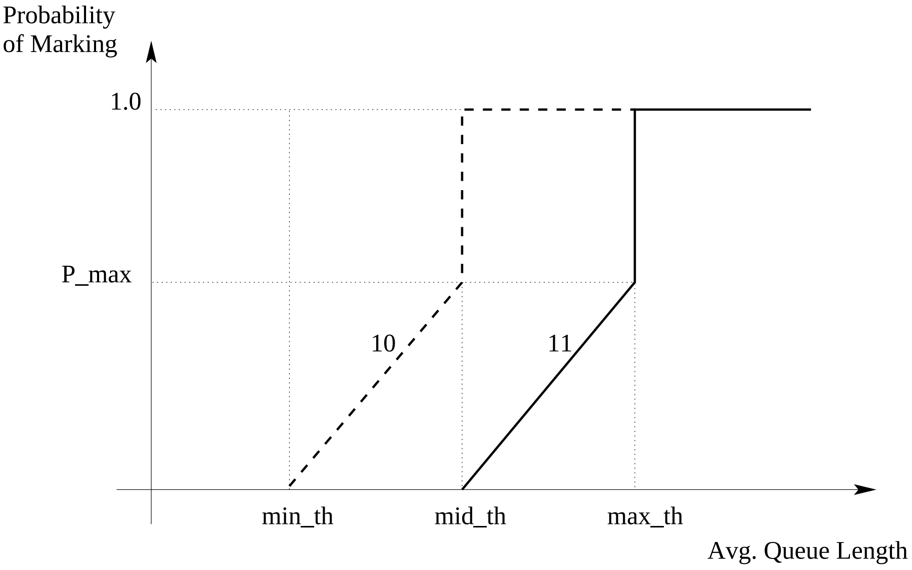

Figure 2. Packets marking scheme for WMECN

Figure 2 shows the modified MECN packet marking scheme, at the router. If the average queue is between min_{th} and mid_{th}, then the packets are marked with '10' with a maximum of probability of P_{max}. If the average queue is between mid_{th} and max_{th}, then the packets are marked with '11' with a maximum of probability of P_{max} and the packets that are not marked with '11' are deterministically with '10'. And if the average queue is greater than max_{th} all packets are marked with '11'. This is done, so that all packets that reach the router after mid_{th} get some kind of mark for sure. This increases the accuracy of the scheme. Even though on the outlook, this may seem 'very strict', the fact that the sources go down only once every RTT and they go down only by 80% or 60%, instead of 50% as in a traditional case, helps in making this scheme a high performance system.

Congestion Feedback	Meaning	Weights
01	No Congestion	0
10	Mild Congestion	1
11	Severe Congestion	2

Table 1. Congestion bits, Their Meaning And Weights

3.2. Filtering Context And Source Response

The reflection of the congestion bits at the receiver is done, in the same way as in MECN. As in MECN, the congestion feedback bits can have three values. '01' meaning 'no congestion', '10' meaning 'mild congestion' and '11' meaning 'severe congestion'. The sources when they get the acknowledgment back from the receiver, stores these, congestion feedbacks with relative weights. The congestion feedback '01', has a weight of '0', '10' has a weight of '1' and '11' has a weight of '2'. The Table 1, shows the bit markings, there meanings and their corresponding weights. At any point of time, the last ten congestion feedback values are stored in a table. These ten values are called 'Filtering Context', using which the decisions are made.

The weights of all ten feedbacks are added to get a 'summed weighted feedback'(SWF). This SWF cannot have a value more than 20, since the maximum weight of a single feedback is two. Decisions are taken based on SWF. The Algorithm for the source response is shown below:

1. If $SWF \leq 10$, then no congestion. Do not decrease 'cwnd'.
2. If $11 \leq SWF \leq 13$, then mild congestion. Decrease 'cwnd' by 80%
3. If $14 \leq SWF \leq 16$, then moderate congestion. Decrease 'cwnd' by 60%
4. If $17 \leq SWF \leq 20$, then severe congestion. Decrease 'cwnd' by 50%

In all the above four conditions the packet, which was lost is retransmitted. Again in all the above conditions, 'cwnd' is decreased only if it had not been decreased once before in the same RTT, else, the packets are just retransmitted and the window is not reduced. Each time a packet loss is identified as a 'wireless loss', it is treated as a congestion feedback with a '01' and a entry with a weight '0' is added to the table and similarly each time a packet loss is identified as congestion loss by condition 3 or 4, it is treated as a feedback with '10' and '11' respectively and a entry is added to the feedback table, with appropriate weight. This helps when we have bursty wireless losses or congestion losses. Instead of taking action based on old information, the new decision is added to the feedback table. The various thresholds in this scheme are determined using simulations. It was found that for these thresholds the system gives the minimum error.

The increased congestion feedback information provided by the MECN, the graded response to the congestion feedback, the graded decision making, in deciding wheather a drop is 'wireless loss' or 'congestion loss', all of these make this scheme, very accurate and increases the performace TCP.

4. MISTAKE SCENARIOS

As discussed before, if a mechanism is put in place, to differentiate the 'wireless loss' from a 'congestion loss', huge improvements can be made to the performance of the TCP over wireless. The performance of such schemes, depends mainly on the accuracy with which, they can separate the two kind of losses. There are two kinds of errors that can happen, in such schemes

1. A congestion loss identified as a wireless loss: In this case, the congestion window is not reduced, thinking it is a wireless loss and hence the sources sends more packets, which would contribute to more congestion
2. A wireless loss identified as congestion loss: In this case, the window is reduced, when there is no congestion and the performance of the flow suffers

Both these cases are bad and we want to eliminate these cases as much as possible. While case 2, results in decreased performance, case 1 is more dangerous, since that would result in more congestion than actually decreasing it. From our simulations we found that type 1 error, never happens in our scheme, there by making it a secure scheme to use. That is a congestion loss is never identified as a wireless loss and hence the scheme never contributes to more congestion, while trying to improve the performance of the wireless TCP. The type 2 error occurs, with a very minimum probability. This probability is found using simulations, to be less than 1.0%. As the packet error rate on the wireless link increases, the probability of making a wrong decision decreases. Also it was found that the 'cwnd' is never decreased by half , when a wireless loss is wrongly identified as a congestion loss. Hence the scheme is highly accurate and this scheme performs better than all other schemes, which doesn't make any assumptions about the network.

5. SIMULATIONS AND RESULTS

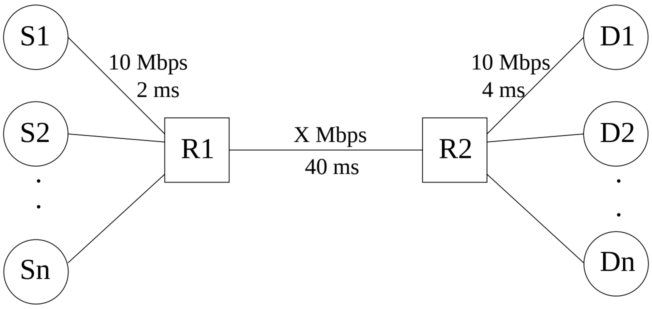

Figure 3. Dumb-bell Network Configuration

5.1. Simulation Configuration

The simulation configuration is shown in Figure 3. A Number of sources S_1, S_2, S_3,..., S_n are connected to a router R_1 through 10Mbps, 2ms delay links. Router R_1 is connected to R_2 through a 1Mbps, 30ms delay link and a number of destinations D_1, D_2, D_3,..., D_n are connected to the router R_2 via 10Mbps 4ms delay links. The bottleneck link has a capacity of 1.0 Mbps and a delay of 30ms. The link speeds are chosen so that congestion will happen only between routers R_1 and R_2 where our scheme is tested. The weight used for queue averaging is $\alpha = 0.002$.

Some of the links between the router R2 and destinations are made wireless links. Some sources (equal to the number of wireless destinations) run a FTP application. Reno TCP is used as the transport agent. (The modifications were made to the Reno TCP). The packet size is 1000 bytes and the acknowledgment size is 40 bytes. The number of sources is varied to alter the congestion level. The background traffic is a UDP flows generated by an exponential on-off model, which runs on other sources. The mean burst period and the mean silence period are 100ms and the burst data rate is 500 kbps. The packet error rate of the wireless link is varied to test the performance of various proposals under different loss scenarios.

To reflect the steady state measurement, the simulation time should be long enough to minimize the effect of the initial transient state. Longer simulation normally generates smoother aggregate results. We tried various simulation time and found the results of 100 seconds show the essential features without noticeable difference from longer simulations. So all aggregate measurements are collected from 100 second simulations.

The proposals we compared include base TCP, ECN, TCP WestWood [7] , Snoop, Wireless MECN (WMECN) and Wireless MECN with link layer retransmissions.

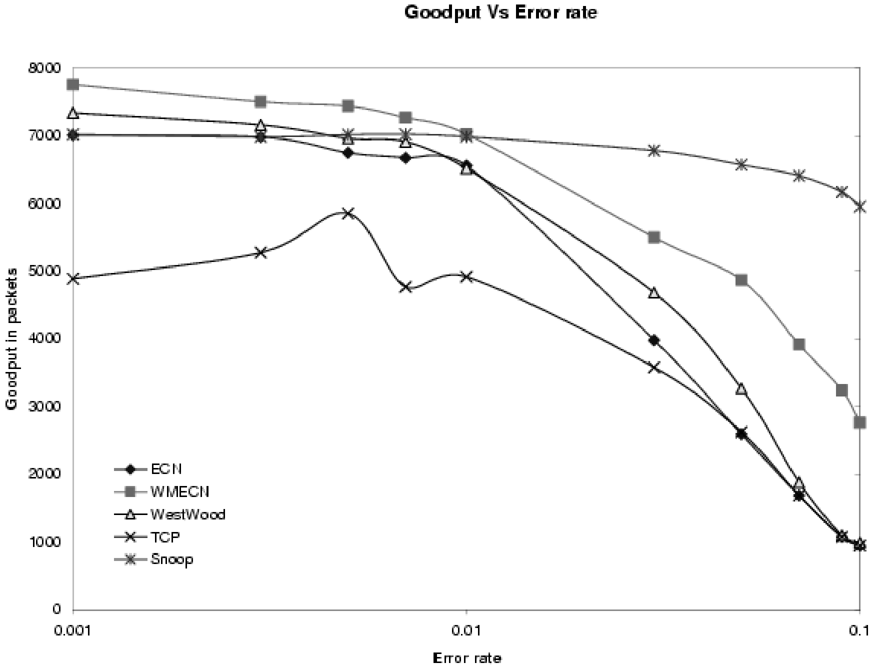

Figure 4. Comparison of Goodput for various schemes

5.2. Results And Analysis

The major metric we used to evaluate the proposals is the *goodput*, which is defined as the number of packets successfully received and acknowledged by the mobile host, excluding the retransmitted packets. The goodput of five proposals under different packet error rates is shown in Figure 4. As we can see that the performance of base TCP is acceptable under light error rates, while its performance degrades fast under heavy error rates. ECN and TCP WestWood performs very well under light error rates, but again the rate of degradation increases as the error rate increases. WMECN performs better than all other proposals compared except Snoop, and its rate of degradation is much slower compared to the other schemes. The reason why snoop performance is stable throughout, is because snoop is the only protocol where the link layer retransmissions done. Hence WMECN performs better than all other protocols of its type (protocols which doesnot demand any additional service from the link layer), including the TCP WestWood.

To make the comparison fair with Snoop, we implemented a link layer retransmission (LRT) protocol similar to DDA [3] and used this along with the WMECN. The results for this scheme (called WMECN + LRT), along with all other schemes is shown in Figure 5. As we can clearly see here WMECN+LRT performs better than all other schemes.

Figure 6, shows the average CWND of various schemes, under various error rates. We can see that the results follow a similar pattern to goodput results.

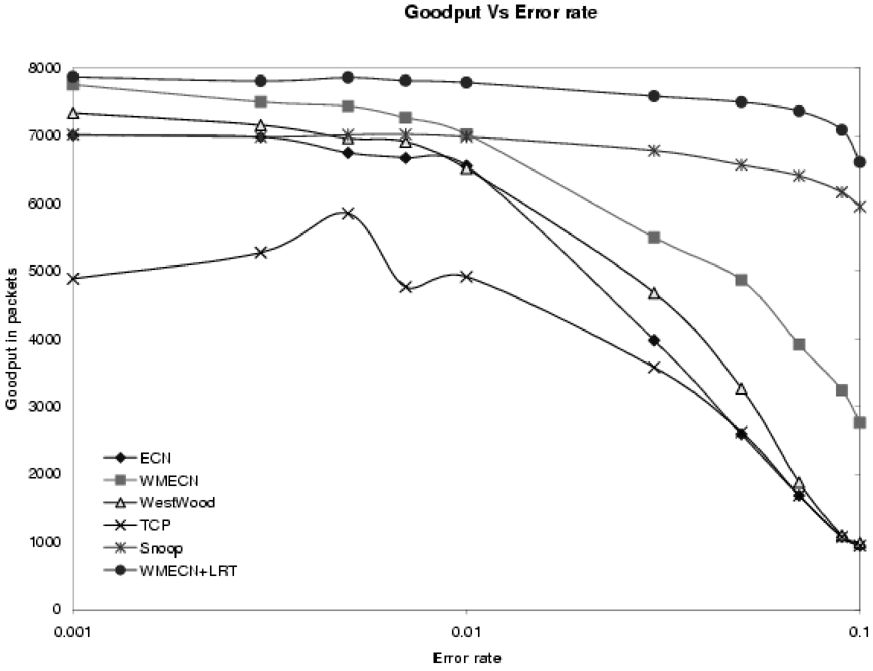

Figure 5. Comparison of Goodput for various schemes

6. CONCLUSION

In this paper a modified version of MECN, called the Wireless MECN (WMECN) was proposed, to improve the performance of the TCP protocol, in wireless networks. A histroy of congestion feedbacks from the network is maintained at each source and the sources try to make a eduacated guess of wheather a loss is a 'wireless loss' or a 'congestion loss'. The scheme was found to be very accurate and was shown to perform better the base TCP, ECN, TCP WestWood and snoop, using simulations in the ns-2 simulator. WMECN always performs better than protocols, which doesn't implement link layer retransmissions and WMECN along with link layer retransmissions, performs significant better than snoop.

REFERENCES

1. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Transactions on Networking* **5**(6), pp. 756–769, 1997.
2. H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks* **1**(4), 1995.
3. N. Vaidya, M. Mehta, C. Perkins, and G. Montenegro, "Delayed duplicate acknowledgements: A TCP-unaware approach to improve performance of TCP over wireless," *Technical Report, Computer Science Dept., Texas A&M University* , 1999.

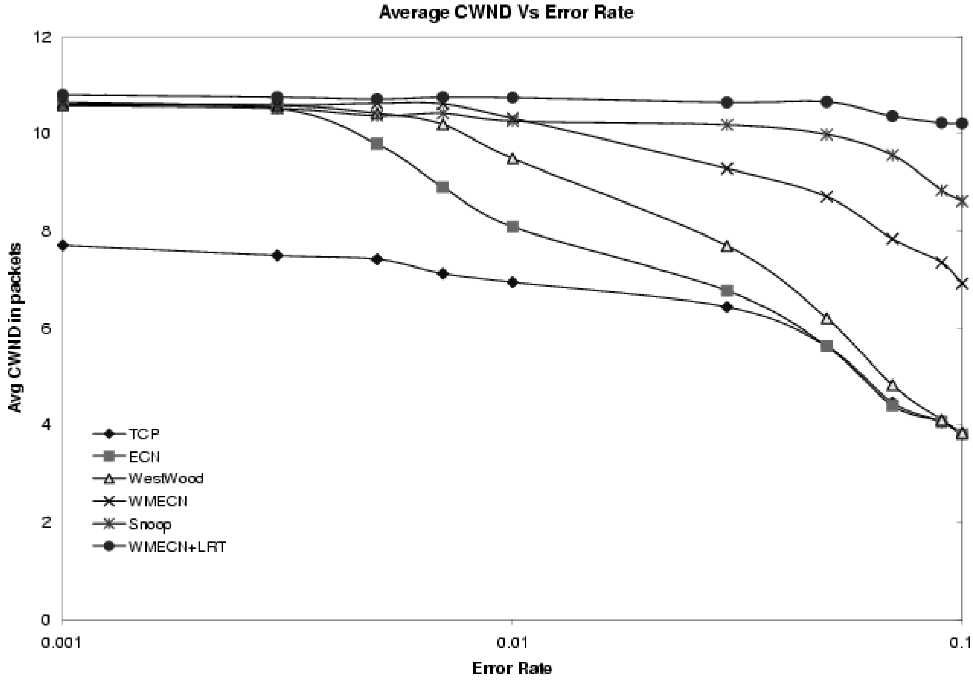

Figure 6. Comparison of Average CWND for various schemes

4. G. Xylomenos, “Multi service link layers: An approach to enhancing internet performance over wireless links,” 1999.

5. A. Bakre and B. R. Badrinath, “I-tcp: Indeirect tcp for mobile hosts,” *15 International Conference on distributed computing systems* , 1995.

6. H. Balakrishnan and R. Katz, “Explicit loss notification and wireless web performance,” in *Proc. IEEE lobecom Internet MiniConference*, (Sydney, Australia), November 1998.

7. C. Casetti, M. Gerla, S. Mascolo, M. Sansadidi, and R. Wang, “Tcp westwood: End-to-end congestion control for wired/wireless networks,” *Wireless Networks Journal* **8**(4), pp. 467–479, 2002.

8. A. Durresi, M. Sridharan, C. Liu, M. Goyal, and R. jain, “Traffic management using multilevel explicit congestion notification,” in *Proc. of the 5th World MultiConference on Systemics, Cybernetics and Informatics SCI’2001, ABR over the Internet*, pp. 12–17, (Orlando, FL), July 22-25 2001.

9. S. Floyd, “TCP and explicit congestion notification,” *ACM Computer Communication Review* **24**, pp. 10–23, October 1994.

10. S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking* **1**, pp. 397–413, August 1993.

11. “ns network simulator.” http://www.isi.edu/nsnam/ns/.

12. K. Ramakrishnan, S. Floyd, and D. Black, “A proposal to add explicit congestion notification (ECN) to IP.” RFC 3168, September 2001.