Performance improvement of TCP over wireless network

Raja singh

Computer science Department, SRIT, Jabalpur,
M.P.India,
rajasinghpatel@gmail.com

Brajesh patel

Asst. Prof. SRIT,Jabalpur
M.P., India ,

Abstract:
TCP's congestion control suffers from a coarse granularity when applied to the multihop ad hoc environment. To overcome this problem, we propose a rate based wireless congestion control protocol (WCCP). There are two components in WCCP. One is at the transport layer. It replaces the window adjusting algorithm of TCP with a rate control algorithm to regulate the sending rate. The other is between the networking layer and the MAC layer. It monitors and possibly modifies the feedback field in TCP data packets when it passes the outgoing packets from the networking layer to the MAC layer and the incoming packets in the reverse direction.
Keywords : WCCP, TCP's congestion control.

1.INTRODUCTION

1.1 Overview

The increasing popularity of wireless networks indicates that wireless links will play an important role in future internet works. Reliable transport protocols such as TCP have been tuned for traditional networks comprising wired links and stationary hosts. Wireless Ad Hoc networks have found many applications in battlefield, disaster rescue and conventions, where fixed communications infrastructures are not available and quick network configurations are needed. To provide reliable transport service over and hence fully exploits the potential of ad hoc networks, efficient congestion control is of paramount importance.

1.2 What Is TCP?

TCP (Transmission Control Protocol) is a set of rules to send data in the form of message units between computers over the Internet. TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet. TCP provides reliable, in-order delivery of a stream of bytes, making it suitable for applications like file transfer _and e-mail. It is so important in the Internet protocol suite that sometimes the entire suite is referred to as "the TCP/IP protocol suite. . TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end.

1.3 TCP In Wireless Networks

TCP has been optimized for wired networks. Any packet loss is considered to be the result of congestion and the window size is reduced dramatically as a precaution. However, wireless links are known to experience sporadic and usually temporary losses due to fading, shadowing, hand off, etc. that cannot be considered congestion. Erroneous back-off of the window size due to wireless packet loss is followed by a congestion avoidance phase with a conservative decrease in window size which causes the decrement of throughput, inefficiency in network resource utilization, and excessive interruption of data transmissions. In a wireless network, packet losses occur more frequently due to link error because of the unreliability of the physical link. This implies the unnecessary invocation of the congestion algorithm

1.4 TCP Operation

TCP operations have three phases:

i) connection establishment
ii) data transfer
iii) connection termination

2 RELATED WORK &BACKGROUNG:

2.1 WINDOW BASED CONGESTION CONTROL

In Window based congestion control ,the sender window size is determined by the available buffer size in receiver(rwnd).In other words ,we assumed that this is only the receiver that can dictate to the sender the size of sender window. We totally ignored another entity here –the network .If the network cannot deliver the data as fast as they are created by sender , it must tell the sender to slow down.In other word

The network is another entity that determines the size of sender's window.Sender window is also determined by the congestion in the network.
The actual size of window is minimum of these
Actual window size =minimum(rwnd,cwnd)

2.1.1 Congestion Policy

TCP's general policy for handling congestion is based on three phases :

2.1.2 Slow start

- ✓ Initially, the congestion window is one segment large
- ✓ Congestion window increases by one for every newACK
- ✓ Congestion window doubles every RTT
- ✓ Congestion window reset to zero after a loss

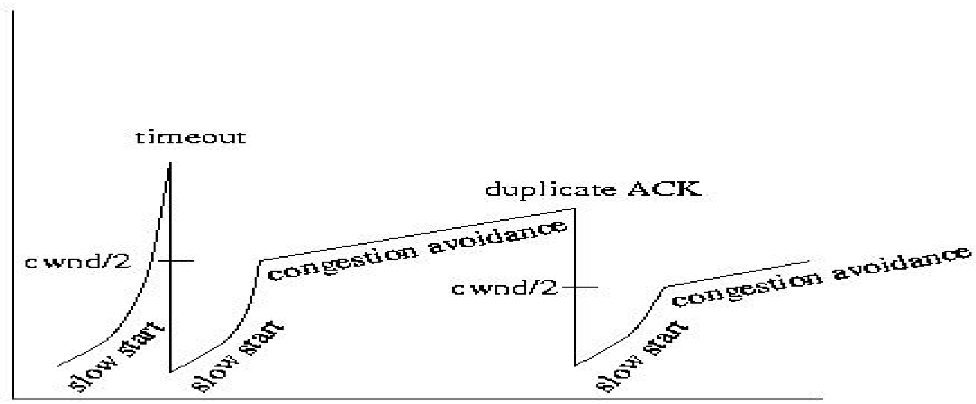
2.1.3Congestion Avoidance

Congestion avoidance is used to grow the congestion window slowly. This is done after a segment has been lost or after ssthresh has been reached.The size of congestion window increases as

$$cwnd = cwnd + MAX (1, (1/cwnd))$$

2.1.4 Congestion Detection

- ✓ ssthresh is set to (cwnd/2)
- ✓ If detection is by RTO, a new *slow start* phase starts
- ✓ If detection is by three ACKs, a new *congestion avoidance* phase starts



2.2 Rate Based Congestion Control

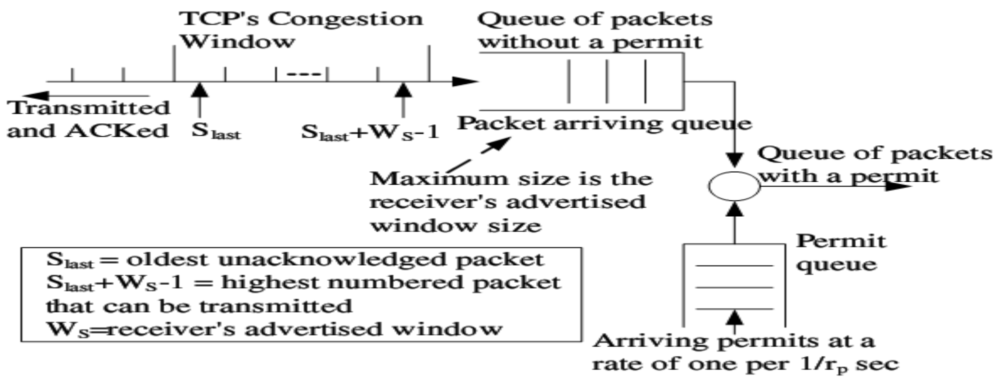


Fig. 4. Rate control mechanism

TCP's congestion avoidance mechanisms are not tuned for request-response traffic like HTTP. One such problem has to do with some TCP implementations forcing slow-start in the middle of a connection that has been idle for a certain amount of time, even if there is no packet loss. Other existing TCP implementations do not treat idle time as a special case and use the prior value of the congestion window to send data.It can lead to degradation of performance.

Rate based algorithms requires the following changes to TCP:

1. Idle time detection and indication that RBP needs to be started.
2. Bandwidth estimation.

3. Calculation of the window that we expect to send in RBP and the timing between segments in that window.
4. A mechanism that clocks the segments sent in RBP.

2.3 Rate Based Algorithms preferred over Window Based

The optimal sending rate per RTT is less than one packet/RTT, we can see that window based congestion control protocols such as TCP tend to overshoot the network capacity as the minimum increase in window size is one packet. In other words, the granularity of window based congestion control mechanism is too coarse. In this sense, window based protocols are not appropriate for supporting stable and reliable transport service in multihop ad hoc networks. Therefore, to provide high throughput, short delay and stable performance with few packet collisions, we opt for an efficient rate-based congestion control algorithm.

3 PROPOSED WORK:

3.1 WIRELESS CONGESTION CONTROL PROTOCOL

In the rate-based congestion control algorithm, to calculate the ideal sending rate, the source is in dire need of a timely and easily measured metric which should satisfy two requirements. First, as mentioned in previous discussion, since MAC contention is tightly coupled with congestion, a candidate of congestion signal should reflect the condition of MAC contention and collision. Second, in order to fully utilize the shared channel without causing severe congestion and packet collision, the candidate should indicate the available bandwidth. The channel busyness ratio *rb*, which is defined as the ratio of time intervals when the channel is busy due to successful transmission or collision to the total time, meets these two requirements.

3.2 Algorithm for Implementing WCCP

- Initially *rp =0,Tc=0, fb=0* in TCP header
- WCCP packet carries a congestion header including three fields: *rp*, *Tc*, *fb*
- When the first ACK arrives sender sets *rp=1/RTT*
- Tc is calculated using the relation max(srtt, β/ *rp*)
- For the next packet *rp Tc* are known and *fb* is calculated at each node as follows
- *rb* , channel business ratio is calculated at each node to determine the available bandwidth at each node
- *rb* is the ratio of total lengths of busy periods to the total time during a time interval
- Available BWa at each node is calculated using *rb* as

$$BW_a = \begin{cases} BW(th_b - r_b)\overline{data}/T_s & ,(r_b < th_b) \\ 0 & ,(r_b \geq th_b) \end{cases}$$

- data is the average lengths (in seconds) for the successful transmission of the data packets *Ts* is the average time of a successful transmission at the MAC layer
- channel resource *ΔS* for each node proportionally to its current traffic load S
  When *rb* < *thb*, *ΔS* is positive
  Increase the traffic
  When *rb* >= *thb*, *ΔS* is negative
  Decrease the traffic

$$\Delta S = \frac{th_b - r_b}{r_b} \times S.$$

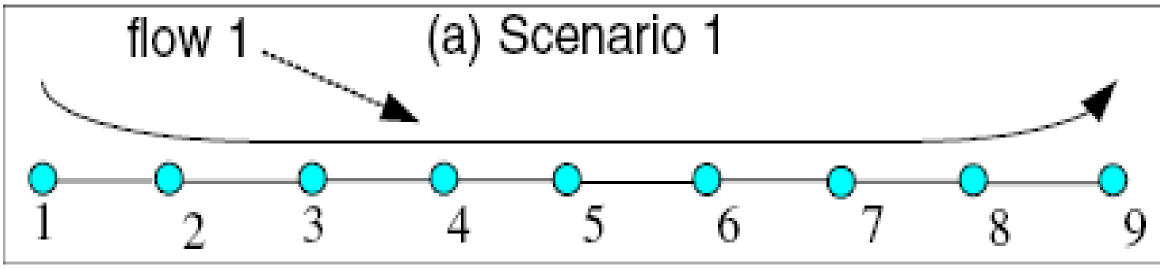
$$C_p = \frac{\Delta S}{\overline{T_c} J}$$

- the increasing amount of traffic rate for each flow(bytes/s)
- This increase in traffic rate divided by payload size gives us the increase in senders arrival rate *fb*
- *rp* ,*Tc*, and the newly calculated *fb* a are sent from receiver back to sender through ACK .
- Now the updated arrival rate at sender becomes

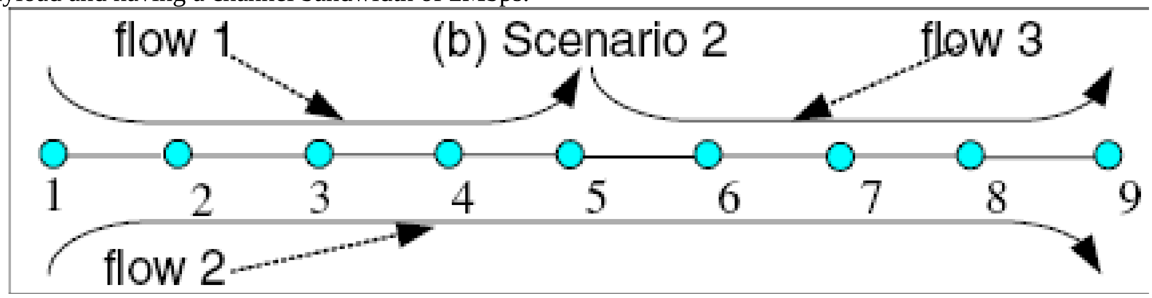
  rp= rp + fb
- Above algorithm is iteratively applied to update the value of sending rate *rp*

3.3 Scenarios Implementation and simulation result:

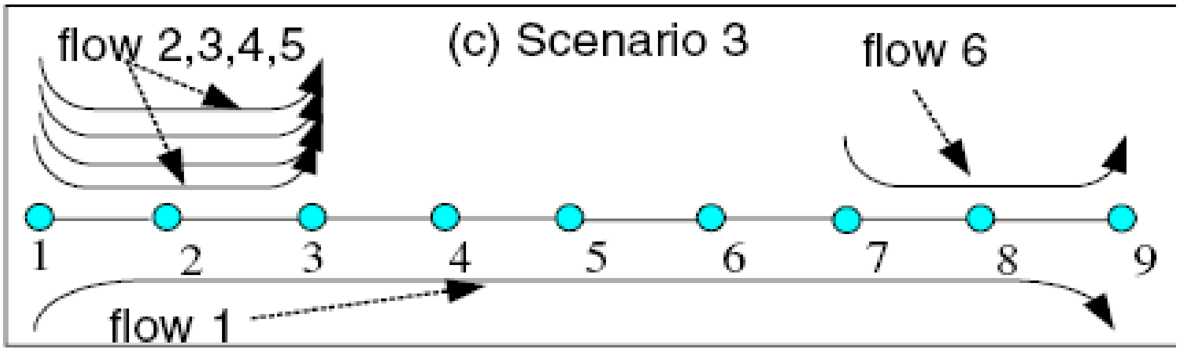
To illustrate the coupling of congestion and medium contention, we use NS 2.27 to conduct a set of simulations over a 9-node chain topology as shown.One or more TCP flows with 1000 bytes payload traverse from node 1 to node 9. The pre-computed shortest path is used, so there is no routing overhead. The channel bandwidth (channel transmission rate) is 2 Mbps. Simulations run for 300 seconds.



In this scenario there is a single TCP flow from node 1 to node 9 with 1000 bytes of payload and having a channel bandwidth of 2Mbps.

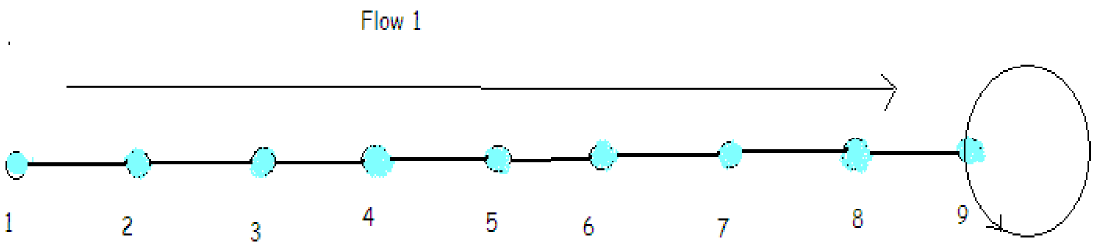


In this scenario there are 3 TCP flow, flow 1 from node 1 to node 5, flow 2 from node 1 to node 9,flow 3 from node 5 to node 9.All other channel characteristics remain same as in scenario 1.



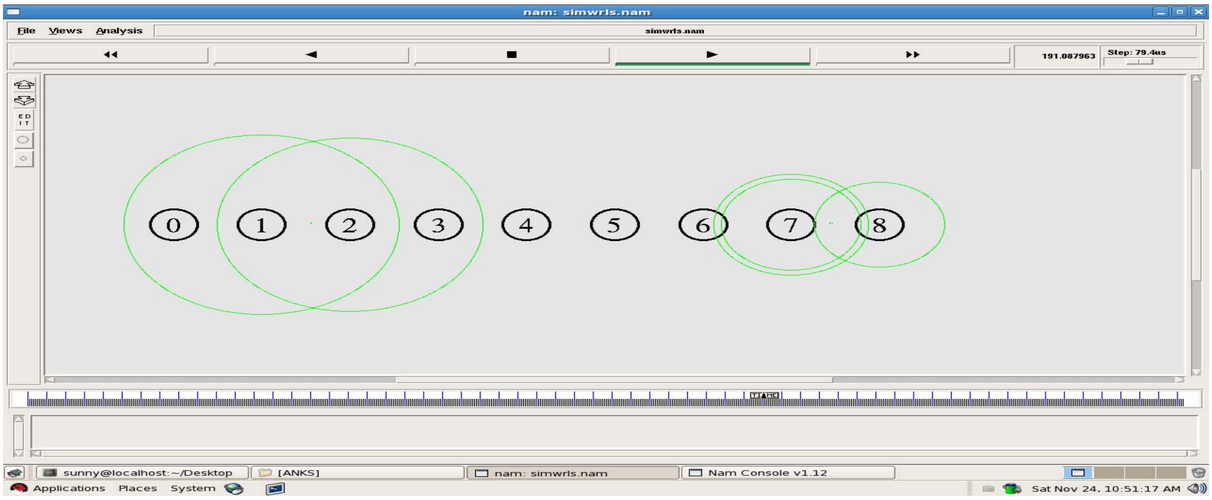
In this scenario there are multiple TCP flows from node 1 to node 3. One flow from node 1 to node 9 and one flow from node 1 to node 9.

Finally scenarios were implemented having mode nodes. In this a 9 node chain topology is Simulated having its last node mobile.

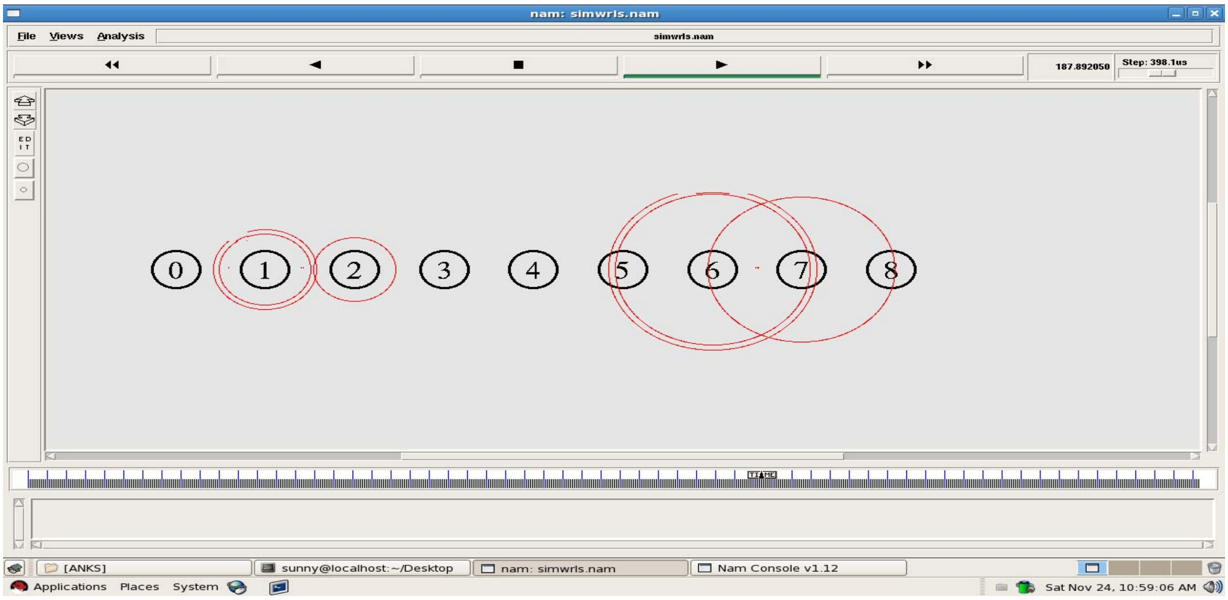


In this scenario last node is mobile, so here introducing mobility in wireless networks. It is 9 node chain topology having 1000 bytes of payload and channel bandwidth of 2Mbps. Node 9 is mobile having movement in a circle.

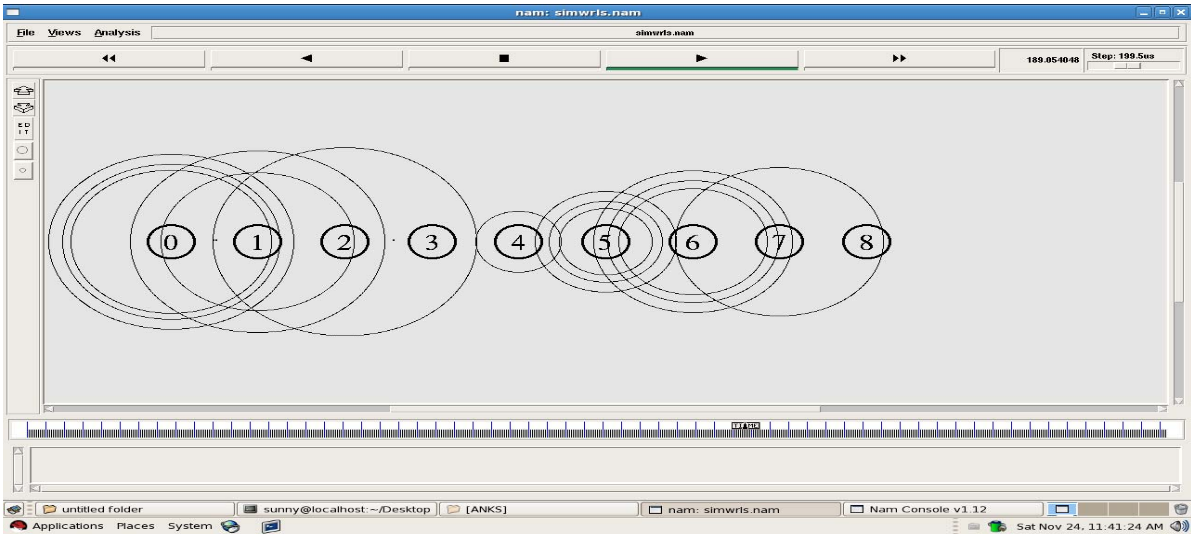
3.3.1 Implementation of Scenario 1



3.3.2 Implementation of Scenario 2



3.3.4 Implementation of Scenario 3



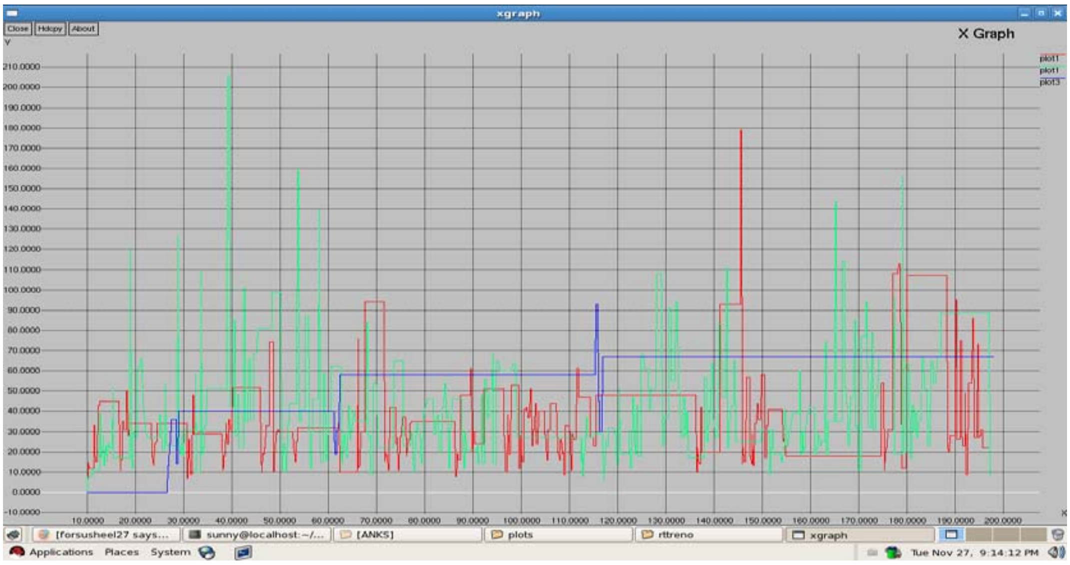
4 Results:

Based on the modifications done to implement the Wireless Congestion Control Protocol (WCCP) the following results were obtained:

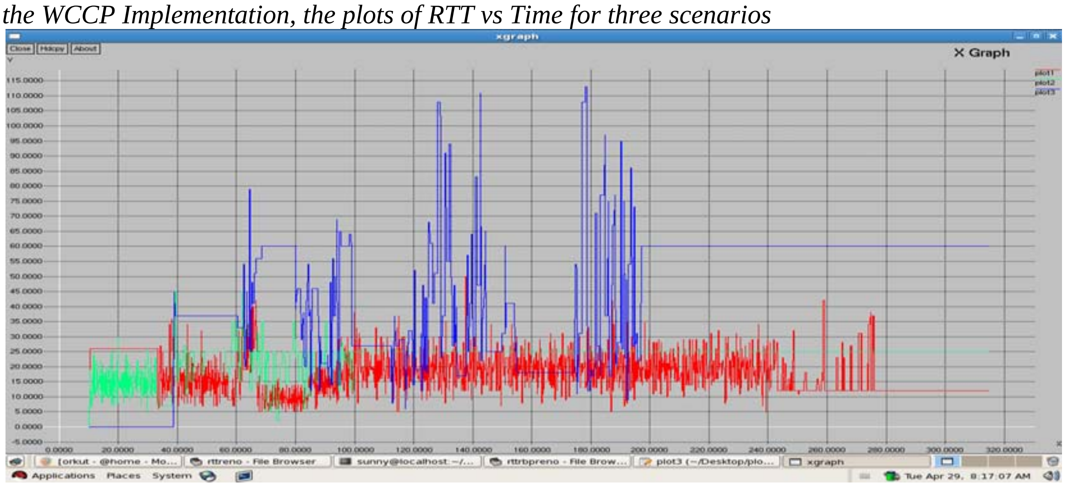
- Initially rtt and srtt are ZERO when sender sends first packet
  When first ACK is received then rtt ans srtt is 1µs and Tc = max($srtt$, β/rp).
- Initially Busyness Ratio is 0.006048 and it keeps on fluctuating between 0.001471 and 0.547434
- Rtt keeps on fluctuating between 1.0µs and 124.0µs
- ΔS varies from =-2.970333 to 22329.470132
- Available Bandwidth varies from 0.08 Mbps to 1.24Mbps
- Sending rate rp varies from 2.24 pkts/s to 1042.94 pkts/s

Total Number of Dropped Packets For Window Based in Scenario 1: 2773
Total Number of Dropped Packets For WCCP in Scenario 1   :1245

Following graphs show the RTT of the three scenarios implemented with TCP Agent used as Tahoe

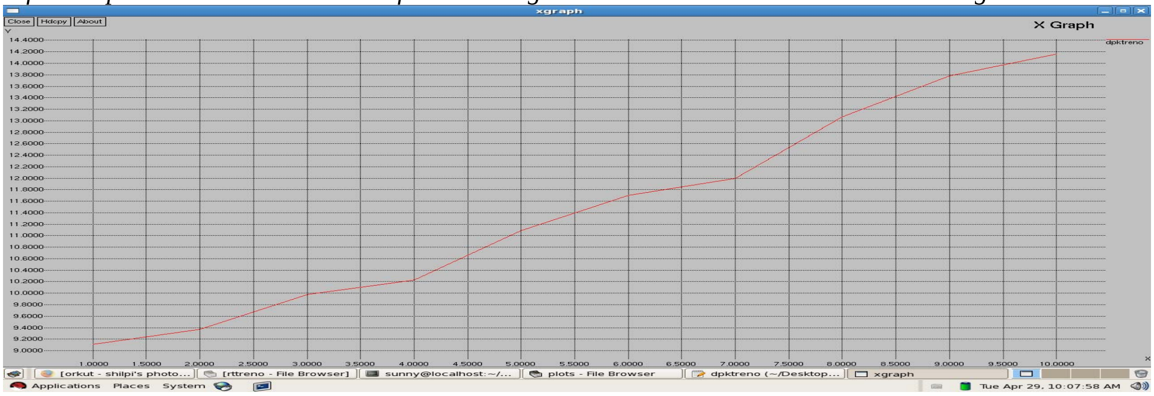


Plot 3 shows the variation of RTT vs Time for 300s simulation for the scenario 1 for flow 1 having 6 flows in total
Plot 2 shows the variation of RTT vs Time for 300s simulation for the scenario 2 for flow 2 having 3 flows in total
Plot 1 shows the variation of RTT vs Time for 300s simulation for the scenario 3 for single flow
On comparing the above plots we can see that as the number of flows increases RTT variations also increases which ultimately shows that throughput decreases.
Using the WCCP Implementation, the plots of RTT vs Time for three scenarios



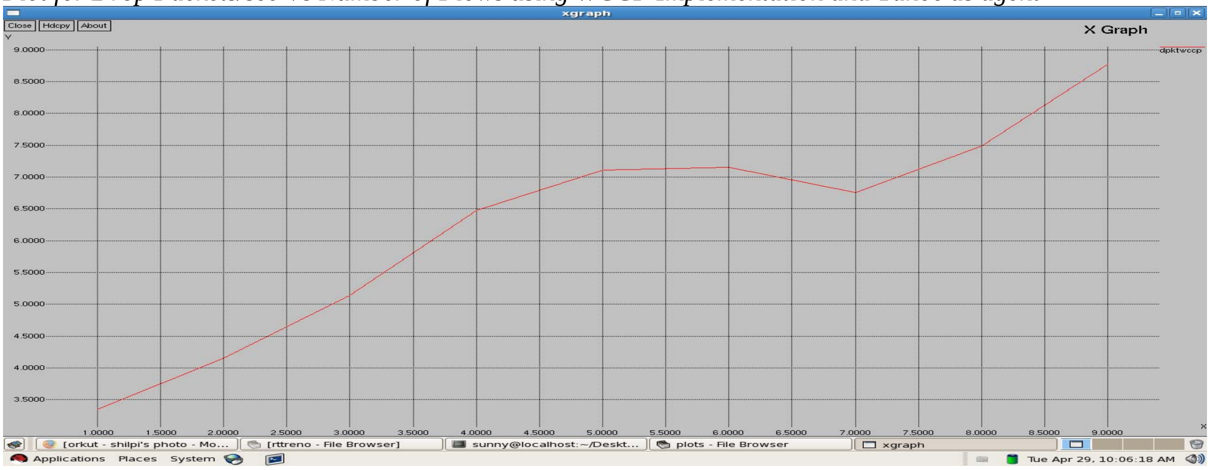
Plot 1 for scenario 1
Plot 2 for scenario 2
Plot 3 for scenario 3
The Above plots shows that by using rate based protocol for implementing the scenarios the variation in RTT reduces considerably, hence an increase in throughput which is desired for the performance improvement.

Plot for Drop Packets/sec Vs Number of Flows using Window based Protocol and Tahoe as agent

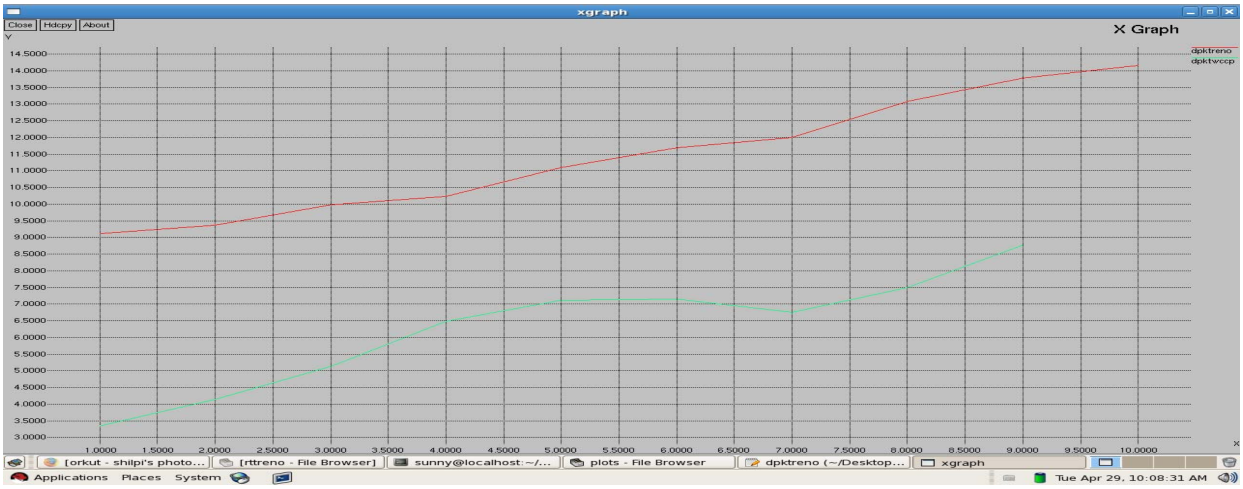


The plots shows that as the number of flows increases the dropped packets/sec also increases

Plot for Drop Packets/sec Vs Number of Flows using WCCP Implementation and Tahoe as agent



Comparative Analysis of Window based and Rate based(WCCP)



The comparative analysis shows that in WCCP implementation has improved performance over Window based approach. The above plots shows that Dropped Packets/sec in WCCP is always smaller than in Window based

5 Conclusion

Congestion control is critical to reliable transport service in wireless multihop ad hoc networks. Unfortunately, traditional TCP suffers severe performance degradation. We propose a systematic solution named Wireless Congestion Control Protocol to address this problem We Implement the algorithm and evaluate WCCP in comparison with TCP in various scenarios. The results show that our scheme outperforms traditional TCP in terms of throughput and number of dropped packets.

6 References

[1] V. Jacobson, “Congestion avoidance and control,” the ACM SIGCOMM’88 pp. 314-328.
[2] Tomoya Hatano, Hiroshi Shigeno and Ken-ichi Okada, “TCP friendly congestion control for highSpeed network”, IEEE, 2007. pp. 10-10.
[3] David X. Wei, Cheng Jin, Steven H. Low, and Sanjay Hedge, “Fast TCP: Motivation, Architecture,
[4] Algorithms, Performance”, IEEE/ACM transactions on networking, 2006
[5] W. Stevens, “TCP Slow Start, Congestion Avoidance,Fast Retransmit, and Fast Recovery Algorithms”, RFC2001, Jan. 1997
[6] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control”, RFC 2581, Apr. 1999.pp.1-8.
[7] S. Floyd and T. Henderson, “The NewReno modification to TCP’s fast recovery algorithm”, RFC 2582, Apr. 1999. Pp.2-15.
[8] V. Jacobson, R. Braden, and D. Borman, “TCP extensions for high performance,” RFC 1323, May 1992 .pp.7-10.
[9] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP selective acknowledgment options,” RFC 2018, Oct. 1996.pp.6-9.
[10] Jon Postel, “Transmission Control Protocol,” September 1981, RFC 793 .pp.1-79.
[11] Allman, M., Balakrishnan, H. and S. Floyd, “Enhancing TCP’s Loss Recovery using Limited Transmit:, RFC 3042, January 2001.pp. 1-4.
[12] S. Floyd: “HighSpeed TCP for Large Congestion Windows”, RFC 3649, December 2003 .pp.5-25.
[13] Lisong Xu, Khaled Harfoush, and Injong Rhee: “Binary Increase Congestion Control for Fast, Long Distance Networks”, 2003.pp.1-12
[14] Tom Kelly: “Scalable TCP: Improving performance in high-speed wide area networks”; ACM SIGCOMM
[15] Behrous A Forouzan , Data Communication And Networking, Tata Mcgraw Hills 4th Edition
[16] Hongqiang Zhai, Xiang Chen, and Yuguang Fang “Improving Transport Layer Performance in Multihop Ad Hoc Networks by Exploiting MAC Layer Information” IEEE transactions on wireless communications, vol. 6, no. 5, may 2007
[17] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz “A Comparison of Mechanisms for Improving TCP Performance over Wireless Links” To appear, Proc. ACM SIGCOMM ’96, Stanford, CA, August 1996.
[18] http://www.isi.edu/nsnam/ns
[19] http://www.wikipedia.org