Enhancing TCP Performance over Heterogeneous Networks

E.S. Saad., J.R. Elbergali., M.A. Ashibani

Abstract—The all-IP wired and wireless hybrid network is becoming a reality and the wireless network is getting more involved in our daily communications. Improving the performance of Transmission Control Protocol (TCP) in wireless communications has been an active research area. The performance degradation of TCP in wireless and wired-wireless hybrid networks is mainly due to its lack of the ability to differentiate the packet losses caused by network congestion from the losses caused by wireless link errors. In this paper, we proposed new scheme called TCP W_CW. TCP W_CW consists of two components, TCP Westwood and Congestion Warning (CW). CW is router configuration to inform TCP protocol that there is congestion in network, and reduce flow control. TCP Westwood employs an end-to-end bandwidth estimator at the sender side. The simulation results show that TCP W_CW achieves improvements in Goodput over TCP Reno, whit varying random error rate.

Index Terms—Westwood, Explicit Congestion Notification, Loss Differentiation, Wireless TCP.

I. INTRODUCTION

TCP is a large expended transport layer protocol that provides connection-oriented, reliable service to the application layer. It is intended for use as a highly reliable end-to-end protocol between hosts in packet-switched computer communication networks. There are several mechanisms in TCP.

A. *Data Transmission*

TCP endpoints transmit data based on sliding window mechanism. Every byte sent has one sequence number. The receiver gives acknowledgement to the sender by ACK packet when the data is received, then the receiver informs the sender its buffer size (*wnd*) in ACK packet. Then the sender slides its sending window, discards data acknowledged, and sends new suitable data.

B. *Retransmission & Congestion Detection*

When one segment is lost, the sender must retransmit the segment to recover from the loss. The sender can detect segment loss by the following mechanisms.

1). Time-Out: It is calculated from average RTT (Round Trip Time) and its variance.

2). n DUPACKs: When the sender receives *n=3* duplicate ACKs (3 DUPACKs) that contain the same sequence number, it switches directly to congestion avoidance without waiting for the timeout.

C. *Congestion Control*

Congestion Control is the most complex part of TCP protocol. It includes SS (Slow Start), and CA (Congestion Avoidance). The following two parameters are very important in Congestion Control.

1). Congestion Window (cwnd): cwnd and *wnd* constrains the amount of data that can be sent out. The sender probes network's ability by increasing *cwnd* and recovers from congestion by decreasing *cwnd. cwnd* is initially set to one segment.

2). Slow Start Threshold (ssthresh): ssthresh is initially set to 65535 bytes. It is set to half of current *cwnd* when congestion occurs.

3). SS: During slow start, *cwnd* is increased by double when one segment is acknowledged. *cwnd* is increased exponentially.

4). CA: When *cwnd* is larger than *ssthresh*, the sender enters into CA phase. *cwnd* is increased by one segment per RTT.

5). FR (Fast Recovery): Duplicate ACKs tell not only that segment is lost but also that there are still data flowing between the two ends. To avoid one abrupt flow reduction, *ssthresh* is set to half of current *cwnd* and *cwnd* is set to *ssthresh+3*. After that, *cwnd* is increased by one segment for each duplicate ACK. When new ACK packet, which acknowledges all data sent before retransmission is received, *cwnd* is set to *ssthresh* and the sender enters into CA phase. If multi-segments are lost, the sender can not transfer from FR to CA phase. It will wait for one Time-Out and enter into SS phase.

The remainder of this paper is organized as follows: section II explain behavior of TCP over wireless networks. Section III addresses related work. Section V presents our proposed solution. We then present in Section IV various simulation results under different network conditions. Last we conclude the paper in section V.

II. TCP OVER WIRELESS

Nowadays, the TCP protocol is widely used as far as fixed hosts data transport is concerned. Since mobile hosts will expect the same services that are offered to fixed hosts. It is necessary to implement TCP for the mobile domain. Using TCP strictly as it is will lead to severe drop in the throughput. The reasons for this are examined in details in this section [3].

Manuscript received May 1, 2006. E. S. is with Almergib university. This work is a part of the M.S. degree in electrical and computer engineering from Almergib University, Alkhoms, Libya, (e-mail:**Elhusain125124@yahoo.com**).

J.R is with Department of Electrical and Computer Eng. 7-th of October University, Libya ,(e-mail:**jelbergali@yahoo.com**).

M. A. is with Department of Electrical and Computer Higher Institute of Industry, Libya, (e-mail: **mashibani@yahoo.co.uk**)

A. Behavior of TCP Wireless Networks

TCP protocol is designed to work over wired networks, but now, the all-IP wired and wireless hybrid network is becoming a reality. The characteristics of wireless and wired networks are different. The following points explain that.

1). Packet Loss: The most outstanding characteristic of wireless links is its bit-error rate due to fading, interference, etc. It brings high segment loss rate to TCP over wireless links. This fact violates the basic assumption of TCP protocol that regards segment loss as the signal of congestion. In this situation, conventional TCP will reduce its *cwnd* frequently. Hence *cwnd* will always be very small and network bandwidth can not be fully utilized [1].

2). Mobility (Frequent Disconnection): In wireless links, temporary disconnections will occur frequently. Handover (Lossy or Non-Lossy) will bring temporary disconnections. Bad link condition, such as in tunnel, may also cause disconnections. When disconnections occur, conventional TCP will suffer Time-Out and enter into slow start and RTO is doubled every time one retransmission is scheduled. When disconnection is recovered, *cwnd* is very small and RTO is very large. This trouble will become even more crucial in the future with the development of pico-cells. In fact this cell size reduction will lead to small cell latencies i.e. the user will not stay for long in a given cell but rather roam from one cell to another quite often [1][3].

3). Limited Capacity: It is commonly agreed that spectrum is a rare resource for wireless applications. Bandwidth and capacity for a given application are limited and results in a low maximum rate at which packets can be transmitted over the air interface. Using FEC to solve the error problem in wireless, but it will waste the bandwidth when correction is not necessary [3].

B. Proposed Solutions To Enhance TCP Performance Over Heterogeneous Networks

TCP is designed to work over wired networks, and interpret all lost of packets as indication to congestion, and can not distinguish between lost of packet caused by congestion from those caused by wireless errors. A lot of work have been done to solve the above problem. The following paragraphs will explain the categories of solutions [1].

1). End-to-End Proposals: In end-to-end proposals, TCP sender and receiver are responsible for the flow control to improve the performance of TCP over wired-wireless networks. The changes are restricted to the endpoints. TCP Westwood [4] is proposed in this field.

2). Split-Connection: TCP connection is comprised by two segments. One is between end host and base station, the other is between mobile terminal and base station. The later can be optimized for wireless link, such as local retransmission. One of these is Snoop [5], which is very common.

3). Link-Layer: The link layer approach rectifies wireless link errors at the second layer. FEC with ARQ is a scheme has been proposed [3].

4). Cross-Layer: A co-operation of multiple layers would lead to a promising solution, or the network assists the transport layer through the intermediate nodes such as routers, and transport layer take this information from third layer same as used in TCP New Jersey [2].

III. RELATED WORK

Related work falls into two components: one is router notification and the other is TCP Westwood.

A. Red and Explicit Congestion Notification (ECN)

Active Queue Management (AQM) methods detect congestion before router queues overflow and try to prevent the state of heavy congestion. One of AQM mechanisms is Random Early Detection (RED). When the average queue size stays below minimum threshold (*minth*), RED router does not do anything. When the average queue size is above *minth* and below maximum threshold (*maxth*), packets are dropped probabilistically. If average queue exceeds *maxth*, all packets are dropped [6], ECN refines this mechanism and uses packets marking instead of dropping when queue remains between *minth* and *maxth*. Therefore, the marking probability at the router increases linearly as the average queue length builds up from the minimum threshold to the maximum threshold. The marking function is depicted in Fig.1 [2]. ECN works by configuring the intermediate router to mark packets with Congestion Experienced (CE) bit in the IP header when the router's average queue occupancy exceeds a threshold, so that the TCP receiver can echo this information back to the sender via ACK by setting the Explicit Congestion Echo (ECE) bit in the TCP header as in Fig.2 [2].

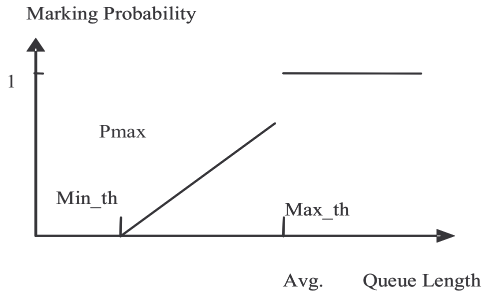

Fig.1. Marking function of ECN [2]

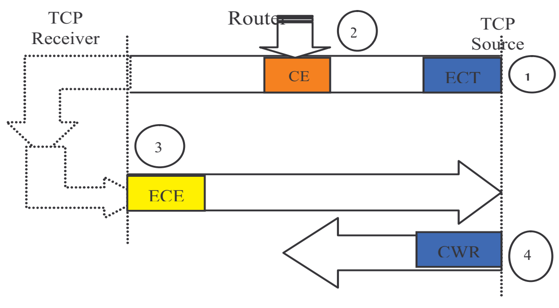

Fig.2. ECN in action [2]

B. Congestion Warning (CW)

The current ECN scheme marks packets probabilistically. The router thereby does not only inform the sender of the congestion, but also influences the congestion window of TCP protocol. The congestion warning (CW), with fewer parameter settings provides essential and accurate congestion information to the sender. The router shall mark all the packets when the average queue length exceeds a threshold (thresh) and leave the TCP sender who receives marks to decide its window adjustment strategy [2]. The router's marking scheme is depicted in Fig. 3.

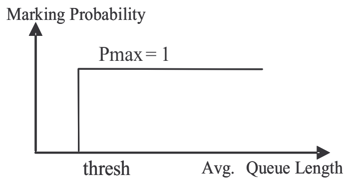

Fig.3. Marking function of ECN [2]

The non-probabilistic packet marking of CW routers helps the TCP sender to differentiate the cause of packet losses. When the TCP sender receives DUPACK with the CW mark, it knows for sure that network is in the congested state and assumes that the packet loss indicated by DUPACK is more likely to be caused by congestion. On the other hand, if DUPACK without the CW mark is received, the sender could assume with higher confidence that the loss has occurred due to transmission of errors.

C. *TCP Westwood*

The key idea of TCP Westwood is to use the "bandwidth estimate" to directly control the congestion window and the slow start threshold. The bandwidth is estimated by monitoring the TCP ACKs. Namely, the source performs an end-to-end estimate of the bandwidth available along a TCP connection by measuring and averaging the rate of returning ACKs. When an ACK is received by the source, it conveys the information that an amount of data corresponding to a specific transmitted packet was delivered to the destination. If the transmission process is not affected by losses, simply averaging the delivered data count over time yields a fair estimation of the bandwidth currently used by the source. When DUPACKs reach to the source, they should also count toward the bandwidth estimate, and a new estimate should be computed right after their reception [4].

V. THE PROPOSED SCHEME

The proposed TCP scheme, TCP W_CW (TCP Westwood with Congestion Warning). TCP W_CW incorporates Westwood and CW introduced in Section III.

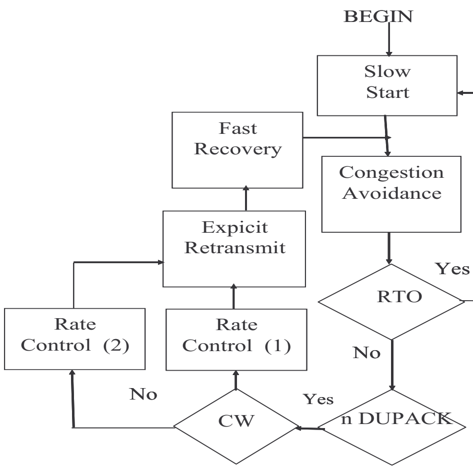

Fig. 4. Flowchart of TCP-W_CW sender's to *n* DUPACKs

TCP W_CW operates as follows: If the received ACK or *n* DUPACKs are marked with the CW bit, it calls the rate control (1) procedure to reduce the window size to half with SS or CA if it is an ACK; or enters the explicit retransmit if it is *n* DUPACKs. When the *n* DUPACKs are received without the CW mark, TCP W_CW renders the packet drop is caused by a random error, and therefore it calls the rate control (2) to adjusting the window size by Westwood mechanism. The flowchart of TCP W_CW sender's response to DUPACK is illustrated in Fig. 4.

IV. SIMULATION RESULTS

We have simulated TCP W_CW in order to show its efficiency, fairness, and friendliness in mixed wire and wireless networks. The simulation tool is the NS-2 network simulator [7].

A. *Goodput*

Goodput is the effective amount of data delivered

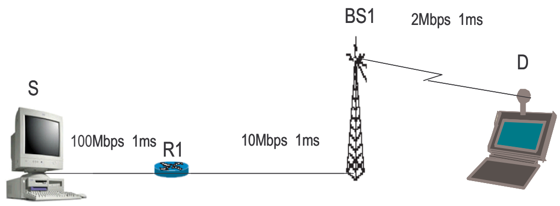

Fig. 5. Goodput simulation environment

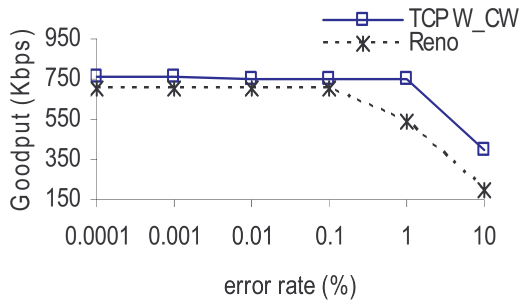

Fig. 6. Comparison of Goodput with TCP Reno and W_CW

through the network. It is a direct indicator of network performance. We expect that a good TCP scheme transmits as many data as possible, while behaving friendly to other TCP flows in term of bandwidth. Fig. 5. shows, the source connects to a router via a 100Mb/s, error free link with 1ms delay, and the router connected to a wireless base station via a 10Mb/s, error free link with 1ms delay. The base station is linked to a wireless mobile node, via a 2Mb/s lossy link with 1 ms delay. A single TCP connection running a long-live FTP application delivers data from the source to the destination. We run the simulation for TCP Reno, and TCP W_CW. The random link error rate (the packet loss rate instead of the BER) at the wireless link varies from 0.00001% to10%. The simulation time has been chosen to be 150s. Fig. 6. shows, the goodput of TCP W_CW achieve higher than TCP Reno.

B. *Fairness of TCP W_CW*

Another important issue of TCP is the fairness. Multiple connections of the same TCP scheme must interoperate nicely and converge to their fair shares. We use the fairness index function in equation (1) to justify the fairness of TCP schemes [2].The Fairness Index Function is

$$F(x)=\frac{(\sum x_i)^2}{n(\sum x_i^2)} \quad (1)$$

Where x is the throughput of the *i-th* connection, and n is the number of connections. *F(x)* ranges from *1/n* to 1. A perfectly fair bandwidth allocation would result in a fairness index of 1. On the contrary, if all bandwidth are consumed by one connection, equation (1) would yield *1/n*. We setup the simulation scenario as shown in Fig. 7, where a total of 6 same TCP flows share 10Mbps link. We run the simulation for different TCP schemes and comparison between their fairness index carried out; the results are summarized in Table I. TCP W_CW , achieves a fairly satisfactory fairness index.

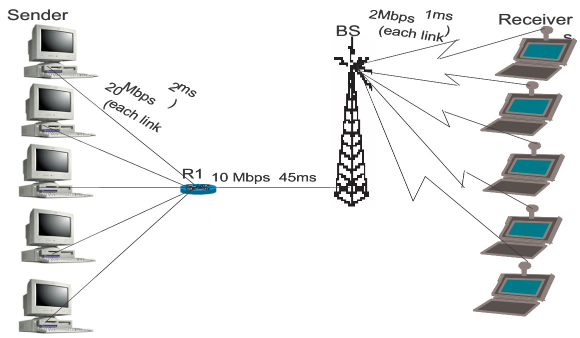

Fig.7. scenario network for obtaining fairness index

TABLE I
Fairness Comparison

Error Rate	TCP Reno	TCP W_CW
0	0.97	0.97
0.1	0.98	0.98
0.5	0.99	0.99
1	0.98	0.99
5	0.96	0.98
10	0.97	0.97

The fairness index is calculated based on a total of 5 TCP connections running the FTP application. Error rate is in units of percentage of packet drops. Link error rate varies from 0.0% to 10%.

C. Friendliness of TCP W_CW

A friendly TCP scheme should be able to coexist with other TCP variants and do not cause bandwidth starvation for the existing TCP connections.

To evaluate the friendliness of TCP W_CW, a mixed wired and wireless simulation scenario have been create, where TCP W_CW coexists with Reno. The simulation scenario is as shown in Fig. 8. The wired link has 70Mbps bandwidth and 45ms delay, and the wireless link has 2Mb/s and 1 ms delay. There are 6 pairs of TCP connections, of which *m* are W_CW connections and *n* are Reno connections. We varied the proportion of these two TCP schemes by adjusting the variables *m* and *n*. All 6 connections are expected to share the same link bandwidth equally , i.e, roughly 11.5Mb/s per connection.

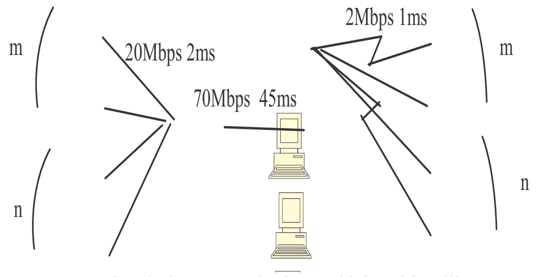

FIG. 8. simulation scenario for verifying friendliness

We first set the link error rate to 0% at the wireless link and record the throughput of each connection. The results are listed in Table II. It could, observed that the bandwidth allocation for each TCP connection is close to its fair share at the bottleneck link.

Table.II
Throughput Comparison over Good Link

Reno Source	TCP W_CW Source	Reno Mean Throughput	TCP W_CW Mean Throughput
2	4	139.1	234.8
3	3	151.2	254.2
4	2	190.6	229.8

The number of Reno and W_CW sources varies. The total number of all sources is 6. Mean throughput is in units of Kbps. The wireless link has no random error.

Table.III
Throughput Comparison over Good Link

Reno Source	TCP W_CW Source	Reno Mean Throughput	TCP W_CW Mean Throughput
2	4	126.0	241.9
3	3	158.0	248.3
4	2	171.6	265.7

The number of Reno and W_CW sources varies. The total number of all sources is 6. Mean throughput is in units of Kbps. The wireless link has 0.1% random error.

In the next set of experiment, the link-error rate is set to 0.1%. The equivalent throughput results are listed in Table III. In this experiment the TCP W_CW achieves higher throughput than Reno when a lossy link exists.

VI. CONCLUSION

TCP over heterogeneous networks consisting of wired and wireless links suffer from degradation in throughput, because TCP can not distinguish between congestion and wireless errors, and many solutions have been proposed. In this paper, we have proposed a new TCP scheme, called TCP-W_CW, to improve the TCP performance in the heterogeneous networks.

TCP-W_CW detects the congestion by CW mechanism implemented at the router that marks packets before the congestion happen, and the TCP sender will reduce the flow control to the half. TCP-W_CW detects that the losses from the wireless link when CW is not set; then, the sender will reduce the flow control using an end-to-end bandwidth estimator as in TCP Westwood. Future work in this direction would be to improve TCP W_CW so that the sender could further differentiate various types of wireless loss.

REFERENCE

[1] *Wu xiuchao*," investigation of reaction TCP and link characterists estimation for wireless links these , 2004.

[2] Kai Xu, Ye Tian, and Nirwan Ansari,"*TCP –Jersey for Wireless IP Communications*", IEEE Journal on Selected Areas In Communications, May 2004.

[3] Fabienne Lefevre, Guillaume Vivier, "Understanding TCP's behavior over wireless Links", Paris.

[4] Claudio, Mario, Saverio, Yahya, Ren, "TCP Westwood: end-to-end bandwidth estimation for enhanced transport over wireless links", Kluwer Academic publishers, 2002.

[5] Furat, Mohamed, Borhanuddin, "Improving wireless snoop performance using fack ACK technique", the Libyan arab international conference, 2006.

[6] Marek Malowidzki, "*Simulation-based study of ECN performance in RED networks"*, military communication institute, Poland.

[7] UCB/LBNL/VINT Network Simulator[Online]. Available:http://www.isi.edu/nsnam/ns/

First A. Elhusain Saad: Received his B.Sc. from the department of Electrical Engineering, at Higher Institute of Industry, Misurata-Libya. Currently, he is in final year of study towards the degree of M.Sc. in Electrical Engineering at Almargib University, Libya.

Second B. Jamal Elbergali: Received B.Sc. 1990 from Alfateh University, Tripoli, Libya, M.Sc. in Electronics Engineering, 1993 from Aligarh Muslim University, India, and Ph.D in Electrical Engineering in 2003 from University of Cape Town. His research focuses on wireless and mobile networks.

Third C. Majdi Ashibani: Received B.Sc. 1989 from Alfateh University, Tripoli, Libya, M.Sc. in Electronics Engineering, 1993 from Aligarh Muslim University, India, and Ph.D in Electrical Engineering in 2003 from University of Cape Town. Since 1993 he has been at Higher Institute of Industry. His research focuses on QoS provisioning for NGN and IP/MPLS networks