Improved Model for a Non-Standard TCP Behavior

Mohammed Abdullah Alnuem[†],

King Saud University, Riyadh, Saudi Arabia

Summary
In previous work [1] the author used a simple periodic model to mathematically estimate the performance of TCP when using a non standard congestion window cut policy where TCP cuts window size after packet drops using a function of the error rate p instead of using fixed cut factor (usually 0.5). In this work the author improve the model in [1] to capture more realistic aspects of TCP behavior so that the packet loss does not follow a simple periodic pattern. The proposed model also open the door for more improvements in the future to include other aspects of a TCP behavior like the effect of timeouts which usually occur after sever congestions.

Key words:
TCP, TCP Modeling, Error Discrimination

1. Introduction

In this work the author extend a model developed in [1] to mathematically model TCP behavior with non standard congestion window cut action in the case of packet drops. In the previous model the author used a simple periodic model which assumes a periodic pattern to represent the congestion window dynamics. The aim is to derive a more detailed mathematical model to represent more realistic features of TCP like duplicate acknowledgement, timeouts and window size limits which have not represented in the previous model. This work will cover the first step of this effort and will propose a model for the case where duplicate acknowledgements are used as the main indication for packet drops. Future work will consider the case where timeouts and window size limits can affect the performance. There are incentives for TCP Modeling [4].First, the huge scale of the TCP operating environment (Models in general are required to gain a deeper understanding of TCP dynamics). Second, Incertitude exist in the TCP operating environment (Incertitude can be modeled as stochastic processes to drive TCP responses).Third, to determine the standards that determine the performance of the system. Last one is to effort the design of TCP algorithms for multimedia applications.

Manuscript received October 5, 2011
Manuscript revised October 20, 2011

2. TCP Standard Behavior

Transmission Control Protocol (TCP) [2], [3], [4], [5] is one of the most important and widely used protocols in the Internet. Many Internet based applications and services like email protocols (Exp: SMTP [6]) or file transfer protocols (like FTP [7]) rely entirely or partially on the services provided by TCP [8]. Many authors believe that most Internet traffic is carried through TCP one way or another [9]. Because of that it has become important to understand how TCP behave and mathematically modeling TCP behavior is one of the good ways to explore and understand TCP behavior in detail.

Tenenbaum [10] has good definition for TCP. He said is reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without error on any other machine in the internet. It fragments the incoming byte stream into discrete messages and passes each one on to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream .TCP also handle flow control to make sure a fast sender cannot swamp a slow receiver with more messages than it can handle. From this definition we can extract the main responsibilities for TCP protocol, one of them is dividing messages into manageable chunks of data that will pass efficiently through the transmission medium. The other one is performing error control, flow control, and acknowledgement: For reliable communication, the sending and receiving computers must be able to identify and correct faulty transmissions and control the flow of data.

Standard TCP behavior is dominated by two mechanisms namely Slow Start [4] and Congestions Control [11]. Slow Start mechanism is used to jump-starts the connection while Congestion Control mechanism is used to maintain

the connection life cycle and to keep it running without creating congestion in the network. When packets are corrupted or dropped due to congestions or wireless errors, both Slow Start and Congestion Control mechanisms cut the sending rate and retransmit the dropped packets. The retransmission of dropped packets is important since TCP guarantee the data delivery from the sender to the recover.

3. Previous Work

Typically, Congestion control mechanism is the most dominating phase of the connection since it is used to maintain the connection during its life time. Slow start is just used in the beginning of the connection to explore the link capacity and when the link capacity is reached congestion control then takes the control of TCP connection and start maintaining a steady state routine. TCP does not return to Slow Start unless there are severe disruptions in the connection which force it to restart again (see TCP-Reno for example [5]). For that the author discussion here will focus on modeling the dynamics of congestion control mechanism.

The main feature of the congestion control mechanism is the Additive increase Multiplicative decrease dynamics of the window size AIMD [3]. The AIMD means that TCP increases its sending rate (represented by the sender window size) linearly and when drops occur it reduces its sending rate in multiplicative manner. Many TCP implementations reduce the sending rate by halving the window size [3] (exponential multiplicative decrease). Using multiplicative decrease has its reasons. The sender is required to reduce the sending rate quickly after drops (drops are translated as strong singes of drops) so that the congested routers will have enough time to clear the congestion [12]. Moreover, using multiplicative decrease will force users with higher bandwidth share the connection in a fare way with other slower users (bigger window size will cut more data, for example a connection with window size of 2000 packets will cut 1000 packets while a connection with 200 packets window will cut 100 packets only) and hence congestions will be resolved faster.

Previous work like Mathis [13] and Padhye [14] has provided mathematical model of TCP standard AIMD behavior. However, not all TCP implementation uses AIMD. So before going to the Model, the author predefined the Essential of TCP modeling [4]. So before going to the author's previous Model, it is important to predefine the Essential of TCP modeling [4]. Every TCP model must get window dynamics and packet loss process. Mainly NewReno and SACK flavours are modeled: triple duplicate ACK with: window = window / 2 and TCP state = Congestion Avoidance. Packet Loss: with window = 1 and TCP state = slow start. In this work the author presents a mathematical model to represent the behavior of TCP implementation which uses a dynamic decrease system to cope with fluctuating packet drop rates which is caused usually by wireless connections and not by congestion. In these cases there is no need for aggressive cut policy since there is no congestion to resolve. The system is described in detail in [15] but the only feature that needs to know here is that it does not cut the window size by a fixed factor of 0.5 after each drop as standard TCP does, instead it cut the window size dynamically based on the number of dropped packets from the last window as following:

Newwindow= oldwindow – numberof droppedpackets

4. Modeling Standard TCP

4.1 Periodic Model

First standard model for TCP is Periodic model which the goal is to find an expression for the average TCP Throughput as a function of packet loss probability (0.5) with TCP in Steady State (Periodic Window Size and Constant Packet Loss Probability) as shown in Figure 1.

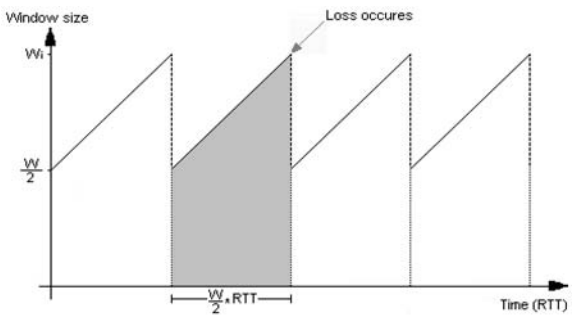

Fig. 1 Periodic model of TCP window dynamics in steady state

Window size (see Eq. 1):

$$w = \sqrt{\frac{8}{3p}} \quad (1)$$

Average sending rate of TCP (see Eq. 2):

$$\bar{x}(p) = \frac{1}{RTT}\sqrt{\frac{3}{2p}} \quad (2)$$

4.2 Detailed Packet Loss Model

Second standard model for TCP is detailed packet loss model which the goal is to find an expression for the average TCP Throughput as a function of packet loss probability (0.5) with random packet loss events and considering triple duplicate ACKs and time outs as shown in Figure 2.

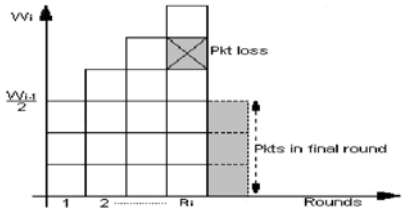

Fig. 2 Detailed analyses of the packets sent during each round of window increase

Expected number of packets loss is:

$$E[\alpha] = \frac{1}{p} \quad (3)$$

Expected number of window size is:

$$E[w] = 1 + \sqrt{\frac{8}{3p} - \frac{5}{3}} \quad (4)$$

Expected number of rounds per period is:

$$E[R] = \frac{1}{2} + \sqrt{\frac{2}{3p} - \frac{5}{12}} \quad (5)$$

Expected duration of the period is:

$$E[A] = RTT\left(\frac{3}{2} + \sqrt{\frac{2}{3p} - \frac{5}{12}}\right) \quad (6)$$

Expected number of packets transmitted is:

$$E[y] = \frac{1}{p} + \sqrt{\frac{8}{3p} - \frac{5}{3}} \quad (7)$$

Sending rate of the TCP source is:

$$\bar{x} = \frac{\frac{1}{p} + \sqrt{\frac{8}{3p} - \frac{5}{3}}}{RTT\left(\frac{3}{2} + \sqrt{\frac{2}{3p} - \frac{5}{12}}\right)} \quad (8)$$

Duplicate ACKs with timeout and the limitation on window size of the receiver:

$$B(p) = \min\left(\frac{W_m}{RTT}, \frac{1}{\frac{1}{x} + T_0 * \min\left(1, \frac{3}{E[W]}\right) * p(1 + 32p^2)}\right) \quad (9)$$

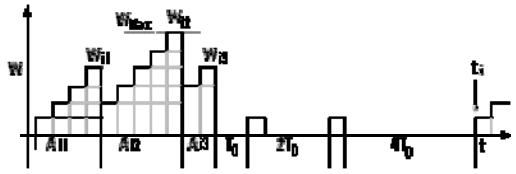

Fig. 3 Intervals that constitute the dynamic of the window in one cycle including a timeout period

5. Proposed Model for Non-Standard TCP Behavior

In previous work [1] the author used a simple periodic model to model the performance of a non standard TCP performance. Meaning of non-standard is that the TCP cuts its window size after drops using other functions other than the well known cut policy which is to cut the window size to half after packet drops. In [1] TCP cuts window size after packet drops using a function of the error rate p instead of using fixed cut factor (0.5).

However, the model the author presented in [1] is a simple

periodic model which does not consider important features of TCP performance like the effect of timeouts and the limits imposed by the receiver on the sender window size. So in order to capture these extra features of TCP there is a need to use a more detailed model which uses duplicate acknowledgements as indication of packet drop.

In the following the author will follow same steps presented in [14] to derive the new model. The difference in this case will be in the window cut factor, where in the model presented in [14] the cut factor in 0.5 so that after each drop the window size will be W=2. However, in this case the author will use the cut factor as a function of the error rate f(p) = Wp (i.e. after each drop will cut from the window size W an amount equal to Wp) so that after each drop the window will be W(1-p). For more information about the reasons of choosing W*p as a cut factor please refer to [1].

After each sending round the window size increases by one packet. So that after Xi rounds the window size will be Wi-1 (The concept of rounds is originally proposed by [13]). However after a drop the window size Wi-1 will be reduced by factor Wi-1 p so that the new window size will be Wi-1 (1 - p).

5.1 Duplicate ACKs

The aim is to calculate the total number of packets T sent during period D = (Xi + 1)RTT where Xi is the number of rounds during period i and RTT is the round trip time between sending a packet and receiving the acknowledgement. If the author has these two values (i.e. T and D) then the performance can be calculated as the rate of number of sent packets to the time needed to transmit them. From that the sending rate will be T=D.

From above the final window size is Wi = (Wi-1 (1-p)) + Xi. Now if the author takes the expected value of the window size (see Eq. 10-14)

$$E[W] = E[W](1-p) + E[X] \tag{10}$$

$$E[W] - E[W](1-p) = E[X] \tag{11}$$

$$E[W](1-1+p) = E[X] \tag{12}$$

$${}_pE[W] = E[X] \tag{13}$$

$$E[W] = \frac{1}{p}E[X] \tag{14}$$

To find the total number of packets Ti sent during period i we do summation of number packets sent from the time the window size was Wi-1 (1 - p) to the time the window size became Wi-1 as following:

$$T_i = \sum_{m=0}^{X_i-2}(W_{i-1}*(1-p)+m) + L_i \tag{15}$$

$$= X_i * W_i * (1-p) + \frac{X_i}{2} * (X_i - 1) + L_i \tag{16}$$

$$= X_i\left(W_{i-i} * (1-p) + \frac{1}{2}(X_i - 1)\right) + L_i \tag{17}$$

where Li is number of sent packets in the last round. Observe that when the author took the expected value of Ti:

$$E[T] = E[X]\left(E[w]*(1-p) + \frac{1}{2}(E[X]-1)\right) + E[L] \tag{18}$$

It is observed that from Eq. 14, I get:

$$E[T] = {}_pE[W](E[W]*(1-p) + \frac{1}{2}({}_pE[W]-1)) + E[L] \tag{19}$$

The author will follow the assumption in [14] that the number of packets sent in the last round is a uniformly distributed number between 1 and W then it can assume that Li = W/2 then Eq. 19 will become:

$$E[T] = {}_pE[W](E[W]*(1-p) + \frac{1}{2}({}_pE[W]-1)) + \frac{E[W]}{2} \tag{20}$$

Also [14] showed that for error rate p and window size W following equation is true:

$$E[T] = \frac{1-p}{p} + E[W] \quad (21)$$

From Eq. 20 and Eq. 21:

$$\frac{1-p}{2} + E[W] = {}_pE[W](E[W]*(1-p) + {}_pE[W]\frac{E[W]-1}{2}) + \frac{E[W]}{2} \quad (22)$$

$$E[W]^2(2p-p^3) + E[W](-p-1) + \frac{-2+2p}{p} = 0 \quad (23)$$

And by solving the resulted quadratic equation, I get Eq. 24:

$$E[W] = \frac{P+1+\sqrt{9p^2-22p+17}}{4p-2p^2} \quad (24)$$

Note that the author takes only the positive root since window size is always positive.

In order to calculate the performance the author need also the duration D. For one round D is considered as the round trip time RTT. For X rounds D = (X + 1) RTT where RTT is the average round trip time. From that the expected value for D is given in Eq. 25:

$$E[D] = (E[X] + 1)RTT \quad (25)$$

From Eq. 14 we can see that E[X] = p E[W] and by substituting E[W] from Eq. 24 I get Eq. 27:

$$E[X] = p * \frac{p+1+\sqrt{9p^2-22p+17}}{4p-2p^2} \quad (26)$$

$$= \frac{p+1+\sqrt{9p^2-22p+17}}{4-2p} \quad (27)$$

From Eq. 25 and Eq. 27, I get:

$$E[D] = \frac{RTT}{4-2p} * (-p+5+\sqrt{9*p^2-22p+17} \quad (28)$$

Using equations Eq. 21, Eq. 24 and Eq. 28, the author has obtained both E[T], E[W] and E[D] which makes the necessary elements to calculate the expected sending rate S(p) = E[T]/E[D] as following:

$$S(p) = \frac{E[T]}{E[D]} \quad (29)$$

$$= \frac{\frac{1-p}{p} + E[W]}{E[D]} \quad (30)$$

$$= \frac{\frac{1-p}{p} + \frac{p+1+\sqrt{9p^2-22p+17}}{4p-2p^2}}{\frac{RTT}{4-2p} * (-p+5+\sqrt{9*p^2-22p+17}} \quad (31)$$

5.2 Duplicate ACKs with Timeouts

From the measurements done, the majority of window decreases is due to time-outs rather than fast retransmits. To capture time-out loss indications, the model has to be extended to include the case where the TCP sender times-out. This occurs when: packets (or ACKs) are lost, and less than three duplicate ACKs are received.

By substituting E [W] and S (p) from Eq. 24 and Eq. 31 put in Eq. 32, the author obtain the TCP throughput, B(p), when the window is limited. By following same steps in [16] to get X, I obtain following equation:

$$\bar{X} = \frac{1}{\frac{1}{S(p)} + T_0 * \min\left(1, \frac{3}{E[W]}\right) * p(1+32p^2} \quad (32)$$

where To is the time of sending waits This equation same in [16] but the difference in S(p) and E[W].

5.3 The Impact of Receiver Limitation on Window Size

From the measurements done, it do not consider any limitation of window size which receiver can only process packets up to maximum receiving rate, in which the TCP source should not transmit a explosion of packets at a rate that overhead of the receiver. By following same steps in

[16] to get B(p), I obtain following equation:

$$B(p) = \min\left(\frac{W_m}{RTT}, X\right) \quad (33)$$

where Wm is maximum buffer size which determine a maxi-mum congestion window size for receiver.

The Eq. 33 present a more detailed model with duplicate ACKs, timeouts and the limitation on window size of the receiver to express TCP performance when the cut rate is a function of the error rate. This model in more detailed compared with my previous model presented in [1] and more detailed from Eq. 31 and Eq. 32.

In the following section the author will compare the three models with standard one.

6. Model Validation

In this section, the author will compare between three new non standard models with three standard models. This comparison mainly on mathematically estimate performance of TCP when using a non standard and standard congestion window cut policy for where TCP cuts window size after packet drops using a function of the error rate p and the usual 0.5. First compare on Periodic model I, second compare on detailed packet loss model only with duplicate ACKs II, and the third compare with duplicate ACKs, timeouts and the limitation on window size of receiver III.

Table 1 shows a comparison between simple periodic model with error rate p (called A1) and 0.5 (called A2).Table II shows a comparison between detailed packet loss model only with duplicate ACKs p (called A1) and 0.5 (called A2).Table III shows a comparison with duplicate ACKs, timeouts and the limitation on window size of receiver p (called A1) and 0.5 (called A2).

RTT is the round trip time which starts from 10ms (0.01 second) to 100ms (0.1 second) which covers both inter-city and cross country RTTs [17]. The error rate used to generate values in tables which are 1%, 10%, 30%, 50% and 90%.

Table 1: Comparison in simple periodic model

RTT (s)	*P=.01* *A1-----A2*	*P=0.1* *A1-----A2*	*P=0.3* *A1-----A2*	*P=0.5* *A1-----A2*	*P=0.9* *A1-----A2*
0.01	*9974.97-1224.74*	*974.68— 387.30*	*307.32— 223.61*	*173.21— 73.21*	*82.40—129.10*
0.02	*4987.48— 612.37*	*487.34— 193.65*	*153.66— 111.80*	*86.60 — 86.60*	*41.20— 64.55*
0.03	*3324.99— 408.25*	*324.89— 129.10*	*102.44— 74.54*	*57.74 — 57.74*	*27.47— 43.03*
0.04	*2493.74— 306.19*	*243.67— 96.82*	*76.83— 55.90*	*43.30 — 43.30*	*20.60— 32.27*
0.05	*1994.99— 244.95*	*194.94— 77.46*	*61.46— 44.72*	*34.64 — 34.64*	*16.48— 25.82*
0.06	*1662.49— 204.12*	*162.45— 64.55*	*51.22— 37.27*	*28.87 — 28.87*	*13.73— 21.52*
0.07	*1425— 174.96*	*139.24—55.33*	*43.90— 31.94*	*24.74 — 24.74*	*11.77— 18.44*
0.08	*1246.87— 153.09*	*121.83— 48.41*	*38.41— 27.95*	*21.65 — 21.65*	*10.30— 16.14*
0.09	*1108.33— 136.08*	*108.30— 43.03*	*34.15— 24.85*	*19.25 — 19.25*	*9.16— 14.34*
0.1	*997.50— 122.47*	*97.47— 38.73*	*30.73— 22.36*	*17.32 — 17.32*	*8.24— 12.91*

Table 2: Comparison between detailed packet loss models with only duplicate ACKs

RTT (s)	*P=.01* *A1-----A2*	*P=0.1* *A1-----A2*	*P=0.3* *A1-----A2*	*P=0.5* *A1-----A2*	*P=0.9* *A1-----A2*
0.01	*9956.20 — 1206.29*	*956.61 — 375*	*291.09 — 211.72*	*159.31 — 159.31*	*75.74 — 08.72*
0.02	*4978.10— 603.14*	*478.31 — 187.50*	*145.54 — 105.86*	*79.65 — 79.65*	*37.87— 54.36*
0.03	*3318.73— 402.10*	*318.87 — 125*	*97.03— 70.57*	*53.10 — 53.10*	*25.25— 36.24*
0.04	*2489.05— 301.57*	*239.15— 93.75*	*72.77— 52.93*	*39.83 — 39.83*	*18.93— 27.18*
0.05	*1991.24— 241.26*	*191.32 — 75*	*58.22— 42.34*	*31.86 — 31.86*	*15.15— 21.74*
0.06	*1659.37— 201.05*	*159.44— 62.50*	*48.51— 35.29*	*26.55 — 26.55*	*12.62— 18.12*
0.07	*1422.31— 172.33*	*136.66—53.57*	*41.58— 30.25*	*22.76 — 22.76*	*10.82— 15.53*
0.08	*1244.52— 150.79*	*119.58— 46.88*	*36.39— 26.47*	*19.91 — 19.91*	*9.47— 13.59*
0.09	*1106.24— 134.03*	*106.29— 41.67*	*32.34— 23.52*	*17.70 — 17.70*	*8.42— 12.08*
0.1	*995.62— 120.63*	*95.66— 37.50*	*29.11— 21.17*	*15.93 — 15.93*	*7.57— 10.87*

Table 3: Comparison between detailed packet loss model with duplicate ACKs, timeouts and the limitation on window size of received where Wm=20 PKTs and To=500 MS

RTT (s)	*P=.01* *A1-----A2*	*P=0.1* *A1-----A2*	*P=0.3* *A1-----A2*	*P=0.5* *A1-----A2*	*P=0.9* *A1-----A2*
0.01	*2000 — 1056.77*	*61.66 — 56.05*	*2.59 — 2.58*	*0.44 — 0.44*	*0.08 — 0.08*
0.02	*1000— 563.30*	*57.92 — 48.77*	*2.56 — 2.55*	*0.44 — 0.44*	*0.08 — 0.08*
0.03	*666.67— 383.99*	*54.62 — 43.15*	*2.54 — 2.52*	*0.44 — 0.44*	*0.08 — 0.08*
0.04	*500.00— 291.27*	*51.67 — 38.70*	*2.52 — 2.49*	*0.44 — 0.44*	*0.08 — 0.08*
0.05	*400.00— 234.62*	*49.02 — 35.08*	*2.50 — 2.46*	*0.44 — 0.44*	*0.08 — 0.08*
0.06	*333.33— 196.42*	*46.63 — 32.08*	*2.48 — 2.43*	*0.44 — 0.44*	*0.08 — 0.08*
0.07	*285.71— 168.91*	*44.46 — 29.55*	*2.46 — 2.40*	*0.44 — 0.44*	*0.08 — 0.08*
0.08	*250.00— 148.17*	*42.49 — 27.39*	*2.44 — 2.38*	*0.43 — 0.43*	*0.08 — 0.08*
0.09	*222.22— 131.96*	*40.68 — 25.53*	*2.42 — 2.35*	*0.43 — 0.43*	*0.08 — 0.08*
0.1	*200— 118.95*	*39.02 — 23.90*	*2.40 — 2.32*	*0.43 — 0.43*	*0.08 — 0.08*

The author based his model on the work presented in Padhye [14] where the window cut factor that is presented is Wp instead of 0.5, that means in case of p is equal to 0.5, both TCP standard and non-standard models must be equals. As we can see from Table 1, Table 2 and Table 3 the estimated sending rates using A1 and A2 are equals when the probability of packet loss is 0.5.

7. Conclusion and Future Work

In this paper the author provided a mathematical model to predict TCP performance when using non-standard congestion window cut policy after errors. The policy TCP use in this case is to cut the congestion window after drops by a factor Wp where W is the congestion window size and p is the error. This work is based on the model presented in Padhye [14] where the author followed similar steps but in this case the author used W. instead of 0.5 as window cut factor. The author compared the result produced by the proposed model with standard TCP models. The results showed that the models produced very huge and good result when the p is less than 0, and it is equal to the standard TCP models when p is equal to 0.5, and a small difference with the standard TCP models when p greater than 0.5.

In the future work my aim is to improve the model to capture another aspect of TCP performance like the effect of TCP timeout after serious congestions or long packet drop episodes.

Acknowledgments

This research work is supported by the chair of Mobile and Pervasive Computing, King Saud University, Saudi Arabia.

References

[1] Alnuem. M. A., 2009. Improving TCP Performance over heterogeneous networks: The Investigation and design of end to end techniques for improving TCP performance for transmission error over heterogeneous data networks. PhD thesis, University of Bradford.
[2] Postel, J., 1981. Transmission Control Protocol. RFC 793: 1-85, University of Southern California, USA, Last visited: Sept 19, 2011.
[3] Jacobson. V., 1988. Congestion avoidance and control. In: Proceedings of SIGCOMM. August 1989, Stanford, California, USA, ACM Press, pp: 314-329.
[4] Braden. R., 1989. Requirements for internet hosts – communication layers. FRC 1122: 5-112, Internet Engineering Task Force, Last Visited: Sept 19, 2011.
[5] Stevens, W., 1997. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. RFC 2001, Network Working Group, Last Visited: Sept 19, 2011.
[6] Postel. J. B., 1982. Simple mail transfer protocol. RFC 821, Information Sciences Institute, University of California, USA, Last Visited: Sept 18, 2011.
[7] Bhushan. A., 1971. A file transfer protocol. RFC 114, Network Working Group, Last Visited: Sept 18, 2011.
[8] Hassan, M. and R. Jain, 2005. High Performance TCP/IP Networking: Concepts, Issues and Solutions., Prentice Hall.
[9] Elaarag, H., 2002. Improving TCP performance over mobile networks. ACM Computer Survey., 34(3): 357-374, BOI: 10.1145/568522.568524.
[10] Tanenbaum. A. S., 2003. Computer Networks, 4th Edition., Prentice Hall.
[11] Allman, M., V. Paxson, and W. Stevens., 1999. TCP congestion control. RFC 2581, Network Working Group, Last Visited: Sept 16, 2011.
[12] Comer. D. E., 1995. Internetworking with TCP/IP. Vol. 1., 3rd Edition., Prentice Hall.
[13] Mathis. M., Semke. J., Mahdavi. J. and Ott. T., 1997. The macroscopic behavior of the TCP congestion avoidance algorithm. ACM SIGCOMM, Computer Communication Review., 27(3): 1-16, DOI: 10.1145/263932.264023.
[14] Padhye. J., Firoiu. V., Towsley. D. F. and Kurose. J. F., 2000. Modeling TCP Reno performance: a simple model and its empirical validation. IEEE/ACM Transactions on Networking, 8(2): 133-145, DOI:10.1109/90.842137.
[15] Alnuem. M., Mellor. J. and Fretwel. R., 2009. New algorithm to control TCP behavior over lossy links. The Internation Conference on Advanced Computer Control (ICACC), IEEE Computer Society., pp: 941-944.
[16] Padhye. J., Firoiu. V., Towsley. D. and Kurose. J., 1998. Modeling TCP Throughput: a simple model and its empirical validation, University of Massachusette.
[17] Biaz. S. and Vaidya. N. H., 1999. Heterogeneous data networks: congestion or corruption?, PhD Thesis, Taxes A&M University, ISBN: 0-599-45120-3.

Mohammed Abdullah Alnuem is an Assistant Professor in the field of Computer Science, in the college of Computer and Information Sciences, King Saud University, Saud Arabia. Further, he is also serving as Vice Dean for Development and Quality in E-Transactions and Communications. He received his PhD in Mobile Computing and Networks from the school of Informatics, University of Bradford, UK and M.S. in Distributed Systems and Networks from the same university. His fields of research interests include Computer Networks, Distributed Systems, Wired and Wireless Networks and Software Engineering.