Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ARIMA MODEL-BASED RTT ESTIMATION FOR CONGESTION AVOIDANCE MECHANISM IN ASYMMETRIC NETWORKS

[1]JEYASEKAR A., [2]KASMIR RAJA S.V., [3]ANNIE UTHRA R.,
[1]Assistant Professor, Dept.of Computer Science & Engineering, SRM University, Chennai, India.
[2]Dean (Research), SRM University, Chennai, India.
[3]Assistant Professor, Dept.of Computer Science & Engineering, SRM University, Chennai, India.
E-mail: [1]ajeyasekar@yahoo.com, [2]svkr@yahoo.com, [3]annieuthra@yahoo.com

ABSTRACT

The delay-based congestion control mechanism performs well over the bandwidth-symmetric network rather than bandwidth-asymmetric network. The upstream and downstream link of asymmetric network has different bandwidth ie high bandwidth in the downstream link and low bandwidth in the upstream link. Therefore mostly the bottleneck link is the upstream link, not the downstream link. Therefore even if the downstream link of data flow is uncongested, congestion in the upstream link can disrupt the acknowledgement flow. Even if ACKs are smaller in size than data packets, the upstream link is unable to carry the high rate of ACKs. The congestion in the upstream link increases the round-trip-time of a packet and causes loss of ACKs. The increase in round-trip-time triggers the congestion avoidance mechanism and slows the window growth which reduces the throughput performance. Therefore we propose a congestion avoidance mechanism which uses RTT estimation based on Auto Regressive Integrated Moving Average model (ARIMA). It provides better performance than the delay-based TCP in the asymmetric network and also symmetric network. The simulation results show that the proposed congestion algorithm achieves 39% to 400% throughput improvement than the delay-based TCP.

Keywords: *Network Protocol, TCP-Vegas, ARIMA Model, RTT Estimation, Backlog Packets Estimation, Congestion avoidance, Asymmetric Networks.*

1 INTRODUCTION

The growing demand of data communication has lead to the rapid development of network technologies such as Cable TV networks, Digital Video Broadcast, Asymmetric Digital Subscriber Line, Packet Radio networks etc. One of the most evident characteristics of these networks is the difference in transmit and receive capacity of the links which is called asymmetry characteristics of networks. The upstream and downstream link of these networks has different data rates. For example, the bandwidth of downstream is more than the upstream bandwidth in ADSL networks and cable modem. Therefore congestion in the bottleneck link of upstream direction disrupts the acknowledgement flow even if there is no congestion in the downstream link [4][5][6].

The congestion control mechanism of TCP can be classified into two broad categories: Loss-based congestion control and delay-based congestion control. Loss-based congestion control uses reactive strategy which employs congestion avoidance scheme after the occurrence of packet loss in the network. But delay-based congestion control uses proactive strategy which employs congestion avoidance scheme before the occurrence of packet loss in the network. Delay-based scheme outperforms loss-based scheme with respect to network utilization and throughput because it detects the congestion in early stage [11][13]. But it doesn't perform well over the bandwidth-asymmetric network because it assumes that the upstream link has enough bandwidth to carry the ACK packets. The congestion due to bandwidth-asymmetry increases the queuing time at upstream bottleneck link and causes loss of ACKs. The increase in queuing time results in increase in RTT which triggers the congestion avoidance mechanism of TCP sender and slows the window growth. It results in throughput degradation in the asymmetric networks.

The congestion avoidance mechanism of TCP estimates the number of backlog packets in the network with the help of round-trip time (RTT). Sometimes the acknowledgement packets are

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

backlogged at the queue of the upstream link even though there is no backlog packet in the downstream link. It causes under-utilization of the forward link. Therefore in this paper, we propose a congestion avoidance mechanism using ARIMA model-based RTT estimation (CA-ARTT). We present this paper in two-fold, first to estimate the RTT using ARIMA model and second to mitigate the window size based on estimated RTT. The simulation results shows that CA-ARTT perform well in asymmetric network and achieves better throughput performance.

The rest of the paper is organized as follows. Section 2 describes RTT estimation methods and congestion avoidance mechanisms. Section 3 presents the proposed RTT estimation using ARIMA model. In Section 4 the proposed congestion avoidance mechanism is explained. Section 5 and 6 presents the simulation setup and performance analysis of proposed algorithm respectively.

2 RELATED WORKS

2.1 Packet Delay Estimation

Since packet delay of a network exhibits nonlinear and time varying behavior mostly, a times-series model can be applied to estimate the packet delay. A time-series means a temporal sequence of data points typically measured at successive time. The idea behind a time-series model is that some aspects of the past pattern will continue to remain in the future. There are lot of time series models that are used to predict the RTT such as Auto Regressive (AR) [12], Moving Average (MA) [10], and Auto Regressive Integrated Moving Average (ARIMA) [9].

The basic assumption made in the time series analysis is that the time series should be stationary while model fitting [7][16]. But low order AR model proposed in [12] assumes that RTT is a stationary process. As the characteristics of RTT change over time, it has some non-stationary components. Therefore the AR model is inadequate to model the RTT.. The Exponentially Weighted Moving Average (EWMA) model of RTT estimation is given below

$$SRTT = \alpha * SRTT + (1-\alpha) * RTT \qquad (1)$$

where SRTT be the estimated RTT and RTT denotes currently measured RTT. α is called filter gain constant and suggested value is 0.875. The EWMA model is a low pass filter with high filter gain constant and biased towards the long history of measured RTT. But the quick and high variation in the RTT caused by the congestion in the upstream link and loss of ACK packets, affects only the recent RTTs. Therefore the RTT of recently acknowledged packets are very much significant to predict the RTT of next packet in the bandwidth-asymmetric network. Therefore EWMA model is inadequate for detecting the congestion in these networks [1][8][18].

The traditional ARMA is not suitable for RTT estimation because of the non-stationarity and non-linearity nature of end-to-end delay. Hence Box-Jenkins popularized ARIMA model which transforms the non-stationary series into a stationary series by taking successive differences of the data points of time series [7][9][16]. Therefore we propose a new RTT estimation based on ARIMA model which transform the RTT with non-stationary components into stationary. Since it takes RTT of recently acknowledged three packets to estimate the RTT, it will be very useful to predict the congestion in the upstream link of asymmetric network.

2.2 Congestion Avoidance Mechanism

The delay-based approaches take the variation in RTT as the indication of congestion and reacts before the occurrence of packet loss. TCP-Vegas is a delay-based TCP that estimates the backlog packets in the network using the RTT [2]. The backlog packet estimation algorithm of TCP-Vegas is given below

$$\begin{gathered} Expected = cwnd/baseRTT \\ Actual = cwnd/RTT \\ Diff = Expected - Actual \\ Backlog\ Packets\ (N) = Diff * baseRTT \end{gathered}$$

where *Expected* be the expected throughput and *Actual* be the actual throughput. *Diff* be the difference between the expected and actual throughput. cwnd is congestion window size, baseRTT is minimum RTT measured by the TCP source and RTT is the average RTT of the segments acknowledged during the last RTT. The congestion avoidance mechanism uses two thresholds α and β that determine the onset of congestion as given below

if N > β then cwnd -1
else if N < α then cwnd = cwnd+1/cwnd
else no change in cwnd size
endif

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

If $N < \alpha$, Vegas increases the window size linearly during the next RTT with the assumption that there is no congestion in the network. If $N > \beta$, then Vegas decreases the window size linearly during the next RTT with the assumption that there is congestion in the network. Otherwise, it leaves the window size unchanged assuming that the queue starts building-up. Therefore, the goal of TCP-Vegas is to keep a certain number of packets or bytes in the queues of the network. The threshold values, α and β can be specified in terms of number of packets rather than flow rate.

As the number of backlog packet increases, the queuing time also increases which affects the smoothness of RTT. In the asymmetric network, the smoothness of RTT is frequently affected by the backlog of ACK packets, not by the backlog of data packets. Therefore use of baseRTT may be unbefitting for estimating the expected throughput.

Fu et al [3] proposed a TCP variant called TCP-Veno. It estimates the backlog packets in the buffer as follow

$$N = cwnd/SRTT * [SRTT - MinRTT] \quad (2)$$

where SRTT is calculated using (1). When a packet is lost, Veno compares N with β. If $N > \beta$, it assumes that the packet loss is due to congestion. TCP-Veno suggested β=3 as an optimal value. The estimation of minimum RTT is different than Vegas. The minimum RTT (baseRTT) of Vegas is collected from throughout the life time of the TCP connection. But in TCP-Veno, the Minimum RTT is reset whenever a packet loss is detected either due to time-out or duplicate acknowledgements. The MinRTT is then updated as in the Vegas. This is because of the changing traffic from time to time. But the SRTT belongs to long history of RTT which is unbefitting for predicting the congestion. Therefore, in this paper we propose a RTT estimation based on ARIMA model. The estimated RTT is then used to predict the congestion in the asymmetric network by estimating the number of backlog packets.

3 PROPOSED RTT ESTIMATION

The RTT is composed of propagation delay, transmission delay, queuing delay and router processing overhead. The transmission delay, router processing overhead and propagation delay are deterministic components whereas the queuing delay is random noise components. Since RTTs of packet exhibits nonlinear and time varying behavior mostly, a times-series model can be used to estimate the RTT. The data sets for modeling the RTT were collected from DNS root/gTLD RTT dataset of CAIDA/University of Auckland (The Cooperative Association for Internet Data Analysis) [17]. The RTTs collected at equally spaced time intervals are used for modeling the RTT

3.1 Mathematical Background

An Auto Regressive Moving Average model ARMA(p,q) is represented as

$$X(t) = \underbrace{\sum_{i=1}^{p} \emptyset_i X(t-i)}_{\boxed{\text{AR Part}}} + \underbrace{\sum_{i=1}^{q} \theta_i \varepsilon(t-i)}_{\boxed{\text{MA Part}}} + \varepsilon(t) \quad (3)$$

where $t \geq 0$, X(t) is a temporal sequence of data points typically measured at successive times which is called time series. RTT measured at equally spaced time intervals are typically a time series. ε(t) is a white noise process, X(t-i), i=1,2,3…. are a time lagged time series, p and q are order of AR and MA process respectively. Φ_i and θ_i are coefficients of AR and MA process respectively. It can be determined by Least Square Estimation method (LSE). Eqn-(3) can be rewritten as

$$X(t) = \emptyset_1 X(t-1) + \emptyset_2 X(t-2) + \cdots + \emptyset_p X(t-p) + \theta_1 \varepsilon(t-1) + \theta_2 \varepsilon(t-2) + \cdots + \theta_q \varepsilon(t-q) + \varepsilon(t) \quad (4)$$

$$X(t) - \emptyset_1 X(t-1) - \emptyset_2 X(t-2) - \cdots - \emptyset_p X(t-p) = \theta_1 \varepsilon(t-1) + \theta_2 \varepsilon(t-2) + \cdots + \theta_q \varepsilon(t-q) + \varepsilon(t) \quad (5)$$

It can be expressed in terms of backshift operator (B) as

$$X(t)\left(1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p\right) = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q)\varepsilon(t) \quad (6)$$

The characteristic equations of AR and MA part of ARMA model are given below

$$1 - \emptyset_1 z - \emptyset_2 z^2 \ldots - \emptyset_p z^p = 0 \quad (7)$$

$$1 + \theta_1 z + \theta_2 z^2 \ldots + \theta_q z^q = 0 \quad (8)$$

which are the polynomial of AR and MA. For stationarity, the p roots of the polynomials of AR part should lie outside of the unit circle and for invertibility, the q roots of the polynomials of MA part should lie outside of the unit circle. The common assumption made in this model is that the time series is stationary. Since the characteristic of RTT changes over time, the RTT have non-

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

stationary components so that ARMA(p,q) model cannot be applied directly for RTT estimation.

3.2 RTT Estimation using ARIMA(p,d,q)

Since the ARMA model is not suitable for non-stationary time series, Box-Jenkins popularized ARIMA model which transforms the non-stationary series into a stationary series by taking successive differences of the data points of time series [7][9][16]. Therefore we propose Auto Regressive Integrated Moving Average model ARIMA(p,d,q) for RTT estimation because it removes the non-stationary components from the RTT before the model fitting. The arguments p and q are order of AR part and MA part similar to ARMA(p,q) model and 'd' denotes the order of successive differences of the data points which makes the non-stationary time-series into stationary time-series. In case of RTT data set obtained from CAIDA, the first order difference is enough for transforming the non-stationary RTT series into stationary series. Therefore the first order difference of X(t) be

$$(\Delta^1 X(t)) = Y(t) = X(t) - X(t-1) \tag{9}$$

Then the Auto Regressive Integrated Moving Average model of X(t) is

$$\Delta^1 X(t)\left(1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p\right) = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q)\varepsilon(t) \tag{10}$$

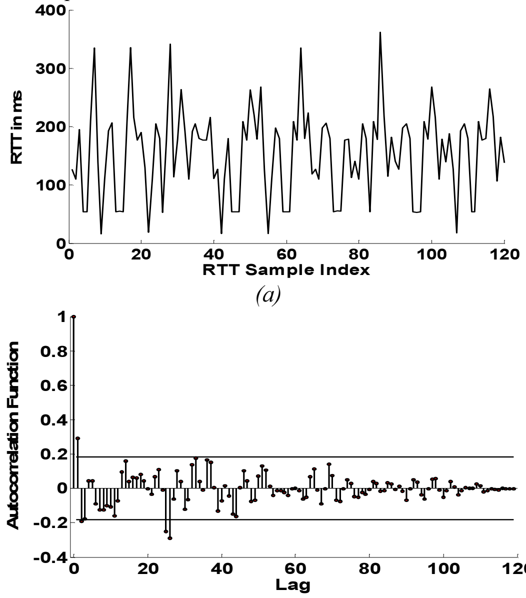
$$Y(t)\left(1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p\right) = (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q)\varepsilon(t) \tag{11}$$

The order p and q are determined by investigating the autocorrelation and partial correlation plot of Y(t). The autocorrelation (ρ_k) and partial correlation function (φ_{kk}) of Y(t) at lag k are calculated as given below

$$\rho_k = \frac{\sum_{t=1}^{n-k}(Y(t)-\mu)(Y(t+k)-\mu)}{\sum_{t=1}^{n}[(Y(t)-\mu)]^2} \tag{12}$$

$$\emptyset_{kk} = \frac{\rho_k - \sum_{t=1}^{k-1}[\emptyset_{k-1,t} * \rho_{k-1}]}{1 - \sum_{t=1}^{k-1}[\emptyset_{k-1,t} * \rho_t]} \tag{13}$$

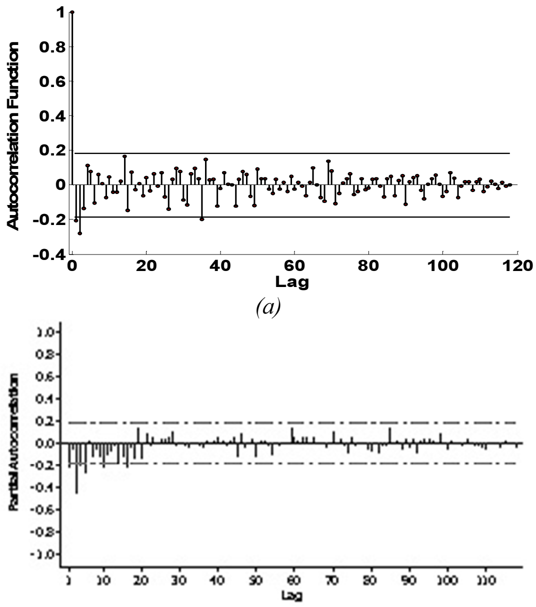
where μ is mean of Y(t), n is number of observation. The run sequence and autocorrelation plot of a sample RTT dataset are shown in Fig. 1. Since the run sequence of RTT has some non-stationary components, the first order difference of RTT is used to transform non-stationary time series into stationary time series. The autocorrelation and partial autocorrelation of the first-order differenced RTT (Y(t)) are shown in Fig. 2. The partial autocorrelation of Y(t) at a lag of 3 is significant. Therefore the ARIMA model with order less than 3 is adequate to model the RTT



(a)

(b)

Figure 1 (a) Run sequence of RTT (b) Autocorrelation of RTT



(a)

(b)

Figure 2 (a) Autocorrelation of first order differenced RTT (b) Partial autocorrelation of first order differenced RTT

In order to select the best order for p and q, we set the order of p and q varying from 0 to 2 because ARIMA model with order less than 3 is adequate. Therefore the possible models are ARIMA(0,1,1), ARIMA(0,1,2), ARIMA(1,1,0), ARIMA(1,1,1)

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ARIMA(1,1,2), ARIMA(2,1,0), ARIMA(2,1,1) and ARIMA(2,1,2). For each model, we estimate the φ and θ using Least Square Estimation (LSE) method. Among these models, an ARIMA model with smallest Akaike Information Criterion Indicator (AIC) and Final Prediction Error (FPE) is selected as the most suitable ARIMA model. The AIC and FPE indicator is calculated as follows

$$AIC = log\hat{\sigma}^2 + 2 * \left[\frac{p+q}{n}\right] \quad (14)$$

$$FPE = V * \left[\frac{1+\frac{p+q}{n}}{1-\frac{p+d}{n}}\right] \quad (15)$$

where σ is the estimated variance of ε(t) and n is the number of observations. The Table 1 shows the ARIMA models and their AIC and FPE. Among these models, ARIMA(2,1,1) is suitable for forecasting the one-step-ahead of RTT because it has small AIC and FPE. The residual is calculated by differencing the actual RTT from the estimated RTT. For the ARIMA(2,1,1) model, the histogram of residual has fixed distribution and normal probability of residual is linear as shown in Fig 3. It means that the ARIMA(2,1,1) model is suitable for forecasting the RTT.

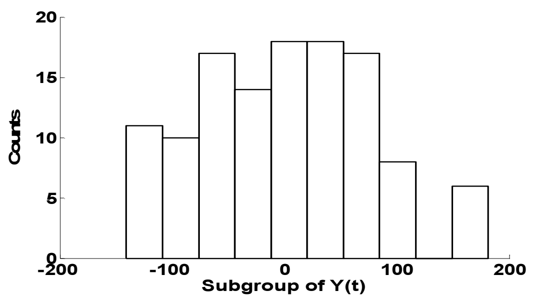
Table 1. The ARIMA model and their AIC and FPE

Sl.No.	Model	AIC	FPE
1	ARIMA(0,1,1)	8.8367	6.8828
2	ARIMA(0,1,2)	8.7475	6.2950
3	ARIMA(1,1,0)	9.0381	8.417
4	ARIMA(1,1,1)	8.7425	6.2636
5	ARIMA(1,1,2)	8.7531	6.3306
6	ARIMA(2,1,0)	9.05122	8.5290
7	ARIMA(2,1,1)	8.73964	6.2457
8	ARIMA(2,1,2)	8.7455	6.2829

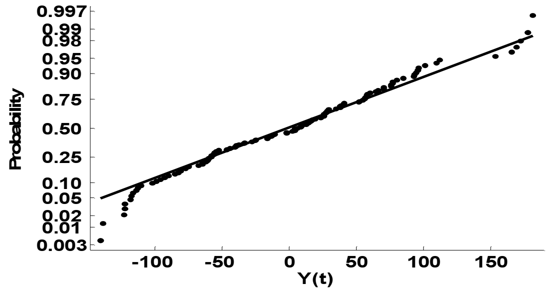
Therefore Eqn-(11) with p = 2 and q = 1 is expressed as

$$Y(t)(1 - \emptyset_1 B - \emptyset_2 B^2) = (1 + \theta_1 B)\varepsilon(t) \quad (16)$$

$$Y(t) = \emptyset_1 Y(t-1) + \emptyset_2 Y(t-2) + \theta_1 \varepsilon(t-1) + \varepsilon(t), t \in \mathbb{Z} \quad (17)$$



(a)



(B)

Figure 3 (A) Histogram Of Residuals (B) Normal Probability Of Residuals

Hence $(X(t), t \in \mathbb{Z})$ is said to be ARIMA(2,1,1) if Eqn-(17) has a stationary solution i.e., the mean, variance and autocorrelation structure do not change over time.

$$E(Y(t)) = \mu = \text{Constant} \quad \forall t \in \mathbb{Z} \quad (18)$$

$$E(Y^2(t)) \leq \infty \; \forall t \in \mathbb{Z} \quad (19)$$

$$\gamma_{YY}(t,s) = \gamma_{YY}(t+h, s+h) \quad \forall s,t,h \in \mathbb{Z} \quad (20)$$

$$Cov(t, t+h) = \gamma_{YY}(h) - \mu_x^2 \quad t,h \in \mathbb{Z} \quad (21)$$

and all p roots of Eqn-(11) with p = 2, $1 - \emptyset_1 z - \emptyset_2 z^2 = 0$ are outside of the unit circle and q roots of Eqn-(11) with q=1, $1 - \theta_1 z$. The coefficient $\emptyset_1$, $\emptyset_2$ and θ_1 are estimated using Least Square Estimation Method. By substituting Eqn-(9) in Eqn-(17), then

$$X(t) - X(t-1) = \emptyset_1[X(t-1) - X(t-2)] + \emptyset_2[X(t-2) - X(t-3)] + \theta_1\varepsilon(t-1) + \varepsilon(t) \quad (22)$$

$$X(t) = (1 + \emptyset_1)X(t-1) + (\emptyset_2 - \emptyset_1)X(t-2) - \emptyset_2 X(t-3) + \theta_1\varepsilon(t-1) + \varepsilon(t) \quad (23)$$

Therefore the one-step-ahead forecast of RTT is

$$ARTT = RTT(t+1) = (1 + \emptyset_1)RTT(t) + (\emptyset_2 - \emptyset_1)RTT(t-1) - \emptyset_2 RTT(t-2) + \theta_1\varepsilon(t-1) + \varepsilon(t) \quad (24)$$

where φ_1 = -0.36251, φ_2 = 0.17573, θ_1 = -0.95108 and residue standard deviation of 0.075

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4 PROPOSED CONGESTION AVOIDANCE MECHANISM USING RTT ESTIMATION BASED ON ARIMA(2,1,1) MODEL

TCP-Vegas takes baseRTT which is minimum of all measured RTT, to calculate the expected throughput. The baseRTT is the sum of transmission delay and propagation delay with the assumption that queuing delay or delay due to backlogged packets equal to zero. If there is no backlogged packet, then round-trip time of the packet is equal to baseRTT. If there are some backlogged packets, then the round-trip time of the packet is sum of baseRTT and delay due to backlogged packets. For the available bandwidth of bottleneck link of downstream and upstream link (C_{fa} and C_{ra}), the round-trip time of the packets is estimated as

$$RTT = baseRTT + N_f/C_{fa} + N_r/C_{ra} \quad (25)$$

Therefore the delay due to backlogged packets in both link can be approximated as

$$N_f/C_{fa} + N_r/C_{ra} = RTT - baseRTT \quad (26)$$

In case of asymmetric network, the bottleneck link is the upstream link, not the downstream link. Therefore the acknowledgement packets are backlogged in the upstream link even though there is no data packets backlogged in the downstream link. Hence Eqn-(26) becomes

$$N_r/C_{ra} = RTT - baseRTT \quad (27)$$

Therefore the difference of RTT and baseRTT can also be used to estimate the delay of backlogged ACK packets. Next the expected throughput is defined as the best possible throughput. TCP-Vegas takes baseRTT to calculate the expected throughput. Because of the onset of congestion occurring in the upstream link of asymmetry- network frequently, the recent RTTs are suitable for estimating the best possible throughput. Therefore in the proposed algorithm, we try to use the ARIMA(2,1,1) model based RTT estimation for predicting the expected-throughput because it takes RTT of recently acknowledged three packets, Hence the new backlog packet estimator using ARTT is given below

$$Expected\ Throughput = cwnd/ARTT \quad (28)$$

$$Actual\ Throughput = cwnd/RTT \quad (29)$$

$$Diff = Expected - Actual \quad (30)$$

$$delay\ due\ to\ backlog = \mathrm{ARTT} - \mathrm{baseRTT} \quad (31)$$

$$N = Diff * (\mathrm{ARTT} - \mathrm{baseRTT}) \quad (32)$$

$$\mathrm{N} = \left[\frac{\mathrm{cwnd}}{\mathrm{ARTT}} - \frac{\mathrm{cwnd}}{\mathrm{RTT}}\right] * [\mathrm{ARTT} - \mathrm{baseRTT}] \quad (33)$$

Therefore the proposed congestion avoidance algorithm is given below

```
On receipt of non-repeated acknowledgement
// Estimation of the RTT using ARIMA(2,1,1)
    If (first segment's acknowledgement)
        //Initialize RTT variables
        RTT1 = RTT2 = RTT3 = RTT
    Else
        RTT3 = RTT2; RTT2=RTT1; RTT1=RTT
    Endif
    Estimate ARTT using Eqn-(24)
//Estimation of backlog packets
    Calculate the estimated throughput
        (cwnd/ARTT)
    Calculate the actual throughput (cwnd/RTT)
    Estimate N using Eqn-(33)
//Congestion Avoidance Mechanism
    If N > β  // Indication of congestion
        cwnd=cwnd-1//the flow rate is decreased
    Elseif N < α  // Indication of no congestion
        cwnd=cwnd+(1/cwnd)// the flow rate is
    Else                          increased
        // current flow rate is unchanged
        no change in current cwnd size
```

5 SIMULATION SETUP

We evaluate the proposed algorithm using ns-2 simulator [14]. The network topology for the simulation is shown in Fig.4. The source node (S) and destination node (D) are sharing the bottleneck link which is bandwidth-asymmetric nature. It means that the upstream link and downstream link of S has different bandwidth. The source node (S_i) and destination node (D_i) are sharing the same bottleneck link and generating backward traffic. We assume a traffic pair in the downstream link and two traffic pair in the upstream link. The propagation delay between the source node and destination node is set to 100ms. The size of data packet and acknowledgement packet is set to 1000 bytes and 40 bytes respectively. The simulation time for all experiments is set to 200sec.

C_r is set based on the normalized asymmetric factor (K) [5]. It is defined as the ratio of the raw bandwidth of the downstream link to the

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

upstream link, divided by the ratio of the packet sizes used in the both links.

$$K = (C_f/C_r) * (S_a/S_d) \quad (34)$$

where C_f and C_r are bandwidth of downstream and upstream link respectively and S_d and S_a are the size of a data packet and ACK packet respectively. The ACK packet size is set to 40 bytes and data packet size is set to 1000 bytes. For example, for a 16 Mbps downstream link and 80 Kbps upstream link, the raw capacity ratio (C_f/C_r) is 200. With 1000 bytes data packet and 40 bytes ACK packet, the ratio of the packet size (S_d/S_a) is 25. This implies that K is 200/25 = 8. Thus, if the receiver acknowledges more frequently than one ACK packet for every 8 data packets, the upstream link will become saturated before the downstream link, limiting the throughput in the downstream link. The downstream and upstream bandwidth for different K is calculated and listed in the Table 2.

Table 2. Normalized Asymmetric Factor (K) And Its Bandwidth Of Downstream And Upstream Link

K	C_r in Kbps			
	C_f =2Mbps	C_f =4Mbps	C_f =8Mbps	C_f =16Mbps
2	40	80	160	320
4	20	40	80	160
8	10	20	40	80
16	5	10	20	40
32	2.5	5	10	20

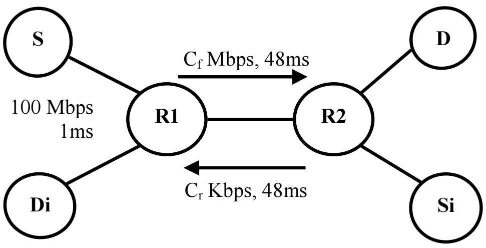
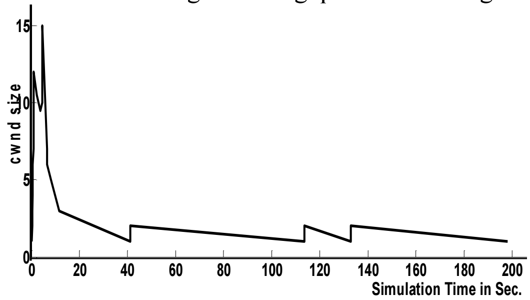


Figure 4. Asymmetric Network Topology

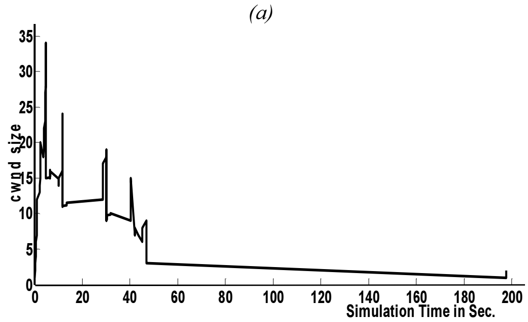
6 PERFORMANCE EVALUATION

When proposed congestion avoidance mechanism is running over the network topology as shown in Fig. 4 without the backward traffic, its performance is similar to Vegas with no retransmitted packets. In order to observe the performance of proposed algorithm under the backward traffic environment, two source-destination pair is used for generating the traffic in the reverse path with the packet size of 1000 bytes. The backward traffic makes the upstream link more congested.

The evolution of congestion window size is shown in Fig. 5 with 1% packet error rate and normalized asymmetric factor K=2. TCP-Vegas reaches the congestion avoidance phase quickly whereas the proposed algorithm stays in the slow start phase long time and increases the window size. It results in higher throughput than the Vegas.



(a)



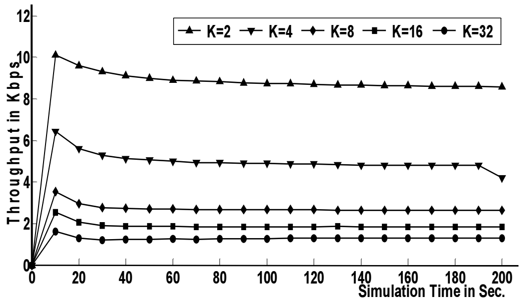
(b)

Figure 5. Evolution Of Congestion Window Size (A) TCP-Vegas (B) Proposed Algorithm

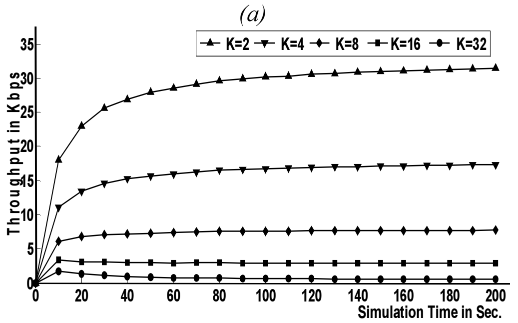
With 0% packet error rate and the increasing value of K from 2 to 32, the average throughput of proposed algorithm is greater than the TCP-Vegas as shown in the Fig. 6. The proposed algorithm achieves 39% to 400% throughput improvement than TCP-Vegas. The packet loss can be analyzed by measuring the number of retransmitted packets during the data transfer. Reduced retransmitted packets indicate that the bandwidth is more effectively utilized. Table 3 shows the comparison of throughput and number of packets retransmitted for the packet error rate of 0% and Table 4 for packet error rate of 1%. Referring Table 3, the proposed algorithm retransmits same or fewer numbers of packet comparing to Vegas. But in the most of cases, the throughput of proposed algorithm is better than Vegas.

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a)



(b)

Figure 6. Comparison of Throughput in Asymmetric Network in the presence of backward traffic (a) TCP-Vegas (b) Proposed Algorithm.

Fig. 7 shows the performance of proposed algorithm under different packet error rate. When the packet error rate is set to 10^{-2}, then Bit-Error-Rate is close to 10^{-4} for the packet size of 1000 bytes. The BER of wireless network can be as high as 10^{-4}. Therefore we set the packet error rate ranging from 10^{-5} to 10^{-2}. Under low lossy environment, the performance of proposed algorithm is improved by 45% to 165% than TCP-Vegas for different value of K.

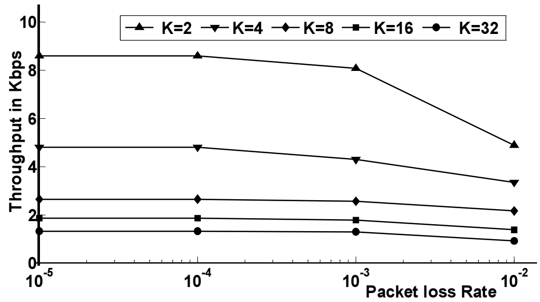
Table 3. Comparison of TCP-Vegas and Modified TCP-Vegas with respect to throughput and retransmitted packets under 0% packet error rate

K	Throughput of Vegas/Proposed Algorithm in Kbps, Total no of retransmitted packets of Vegas/Proposed Algorithm			
	C_f =2Mb/s C_r=20Kb/s	C_f =4Mb/s C_r=40Kb/s	C_f =8Mb/s C_r=80Kb/s	C_f =16Mb/s C_r=160Kb/s
2	2.3/9.7 0 / 0	3.5/19.6 0 / 0	9.2/39.0 0 / 0	19.4/57.4 0 / 0
4	4.2/1.04 12 / 12	2.33 / 9.7 0 / 0	3.5/19.6 0 / 0	9.2/39.04 0 / 0
8	0.52/0.53 9 / 9	4.2 / 1.04 12 / 12	2.33/9.66 0 / 0	3.5/19.62 0 / 0
16	0.31/0.3 11 / 10	0.52/0.52 9 / 9	4.24 / 1.04 12 / 12	2.33/9.7 0 / 0
32	0.16/0.17 11 / 12	0.35/0.3 11 / 10	0.52/0.56 9 / 13	4.24/1.04 12 / 12

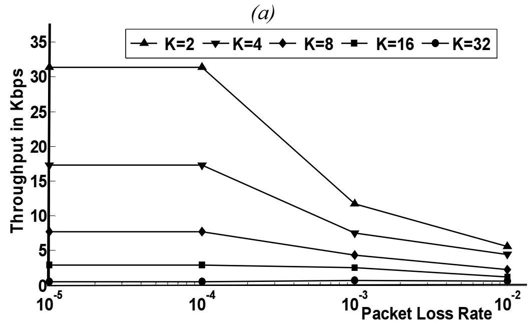
Further we evaluate the proposed algorithm with the round-trip propagation delay set to 100ms, 150ms, 200ms, 250ms, 300ms, 350ms and 400ms. Considering the satellite of geostationary orbit with one way delay of 200ms, we choose 400ms as the maximum round-trip time. The proposed algorithm shows the significant throughput improvement for different propagation delay under 0% picket error rate and 1% packet error rate as shown in Fig. 8 and Fig.9 respectively.

Table 4. Comparison Of TCP-Vegas And Proposed Algorithm With Respect To Throughput And Retransmitted Packets Under 1% Packet Error Rate

K	Throughput of Vegas/Proposed Algorithm in Kbps, Total no of retransmitted packets of Vegas/Proposed Algorithm			
	C_f =2Mb/s C_r=20Kb/s	C_f =4Mb/s C_r=40Kb/s	C_f =8Mb/s C_r=80Kb/s	C_f =16Mb/s C_r=160Kb/s
2	1.57/2.94 7 / 9	2.71/3.13 6 / 20	5.75/6.89 26 / 20	9.47/9.51 28 / 39 /
4	3.19/1.13 18 / 19	1.52/2.94 7 / 9	2.89/6.87 10 / 14	5.75/6.91 26 / 18
8	0.47/0.59 9 / 13	3.72/1.4 22 / 14	1.52/2.94 7 / 9	2.90/4.06 10 / 11
16	0.77/0.3 12 / 10	0.47/0.47 9 / 9	2.69/1.13 22 / 15	1.52/2.94 7 / 9
32	0.16/0.17 11 / 12	0.57/0.3 13 / 10	0.45/0.49 11 / 15	2.45/1.5 20 / 17



(a)

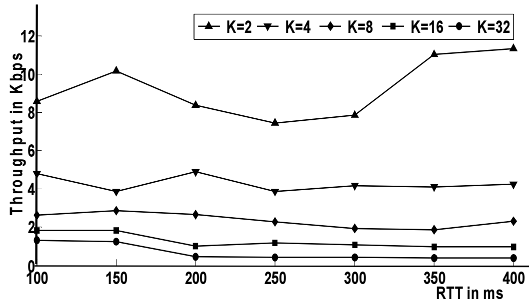


(b)

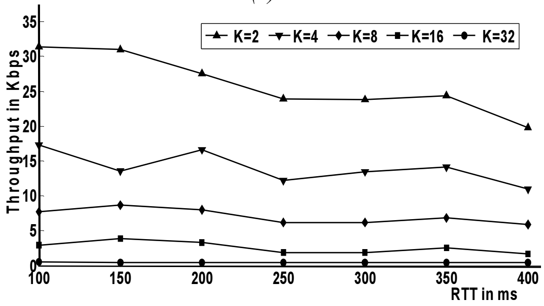
Figure. 7 Comparison Of Throughput In Asymmetric Network With The Packet Error Rate Varying From 10^{-5} To 10^{-2} (A) TCP-Vegas (B) Proposed Algorithm

Journal of Theoretical and Applied Information Technology
10th August 2014. Vol. 66 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

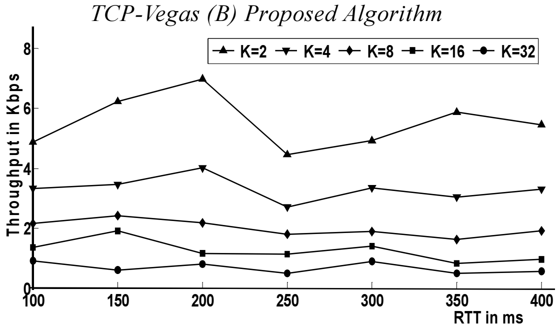


(a)

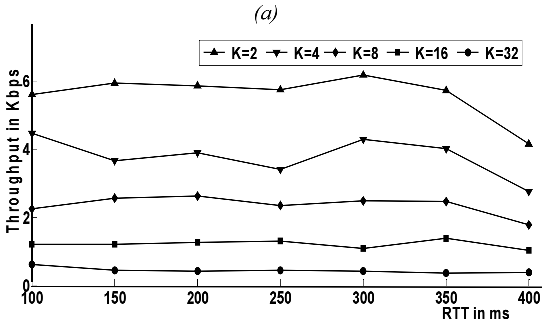


(b)

Figure. 8. Throughput Comparison With RTT Varying From 100ms To 400ms Under Non- Lossy Link (A) TCP-Vegas (B) Proposed Algorithm



(a)



(b)

Figure 9. Throughput Comparison With RTT Varying From 100ms To 400ms Under Lossy Backward Link Ie 1% Packet Error Rate. (A) TCP-Vegas (B) Proposed Algorithm

The proposed algorithm is also evaluated in the symmetric network with C_f and C_r equal to 2Mbps. Without the backward traffic, the throughput of proposed algorithms is similar to Vegas in the symmetric network also. But in the presence of backward traffic, its performance is better than Vegas in the symmetric network also as shown in Table 4. The proposed algorithm achieves 19.6% throughput improvement under non-lossy bottleneck link. But under lossy bottleneck link, the throughput of modified-Vegas is less than Vegas but the number of retransmitted packet is reduced by 74.19% which indicates that the bandwidth of bottleneck link is effectively utilized. The overall performance of proposed algorithm is better than TCP-Vegas

Table 5. Comparison Of TCP-Vegas And Modified TCP-Vegas With Respect To Throughput And Retransmitted Packets Under Symmetric Network

Cf =Cr=2Mbps, Packet error rate is set to 0%		
	Vegas	CA-ARTT
Throughput in Kbps	123.8	148.04
No. of Retransmitted packets	0	0
Cf =Cr=2Mbps , Packet error rate is set to 1%		
Throughput in Kbps	40.07	25.71
No. of Retransmitted packets	108	62

7 CONCLUSION

Asymmetric capacity causes TCP acknowledgements to be lost or become inordinately delayed when a bottleneck link is shared between many flows. It degrades the performance of TCP. In this paper, performance of TCP-Vegas in asymmetric network is improved by using the RTT estimation based on ARIMA(2,1,1) model in the proposed congestion avoidance mechanism. The prosed RTT estimation doesn't require any clock synchronization with receiver. When the proposed algorithm is running over a single TCP connection ie without the backward traffic, the throughput performance is similar to Vegas. But in the presence of backward traffic, the proposed algorithm achieves 40% to 300% for the normalized asymmetric factor ranging from 2 to 32. The proposed algorithm performs well in the symmetric network also. The simulation results show that the overall performance of modified-Vegas is better than TCP-Vegas. We used SRTT for calculating the retransmission timeout and it may be updated using ARTT in future which may improve the performance further

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] Balakrishnan H, V. Padmanabhan, S. Seshan, M. Stemm, M. Katz. (1998) "TCP Behavior of a busy Internet Server: Analysis and Improvements". in proc INFOCOM'98. 7th Annual joint conf. of the IEEE computer and communications societies, Berkeley, CA, pp. 252-262.

[2] Brakmo L.S. and L.L. Peterson.(1995) "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet". IEEE Journal on Selected Areas in Communications vol. 13(8): pp. 1465–1480.

[3] C.P.Fu, S.C.Liew, (2003) "TCP-Veno: TCP Enhancement for Transmission over Wireless Access Networks", IEEE Selected Areas in Communication, Vol. 21(2), pp. 216-228.

[4] Cheng P. Fu, Soung C. Liew, (2003) "A Remedy for Performance Degradation of TCP-Vegas in Asymmetric Networks", IEEE Communication Letters, Vol. 7(1), pp. 42-45.

[5] Cheng-Yuan Ho, Cheng-Yun Ho, Jui-Tang Wang (2011) "Performance Improvement of Delay-Based TCPs in Asymmetric Network" IEEE Communications Letters Vol. 15(3): pp. 355-357.

[6] D. Papadimitriou, Alcatel-Lucent, (2011) M. Welzl "Open Research issues in congestion control" IRTF-RFC 6077, Feb-2011.

[7] Guorui Li, Ying Wang, (2013) "Automatic ARIMA Modeling-based Data Aggregation Scheme in Wireless Sensor Networks", EURASIP Journal on Wireless Communications and Networking, 2013:85, pp. 1-13.

[8] Haeri, A. H. Mohsenian Rad.(2004) "TCP Retransmission Timer Adjustment Mechanism Using System Identification". in proc. of the 2004 American Control Conference, Boston, Massachusetts, pp. 2328-2332

[9] Huda M. A. El Hag, Sami M. Sharifa A. (2007) "Adjusted ARIMA Model for Internet Traffic". AFRICON'07; pp. 1-6.

[10] Jacobson V. (1988) "Congestion Avoidance and Control". in proc. ACM SIGCOMM'88, Stanford, USA, pp. 314-329.

[11] Jim Martin, Arne Nilsson, Injong Rhee, (2003) "Delay based Congestion avoidance for TCP", IEEE/ACM Transactions on Networking. Vol. 11(3), pp. 356-368.

[12] L.L.Lam, Kai Su, C.W.Chan, X.J.Liu.(2009) "Modeling of Round Trip Time over the Internet" in proc 7th Asian Control Conference, Hong Kong, China, pp. 292-297.

[13] M. Allman, V.Paxson and W. Stevens, "TCP Congestion Control", IETF RFC 2581, Apr 1999.

[14] NS-2 Simulator available at www.isi.edu/nsnam/ns April 2010

[15] Purvang D. Dalal, Ridhdhi I. Satoniya. (2012) "Impact of False RTT estimation on TCP performance over wireless networks". International conference on advances in mobile network, communication and its application pp. 115-118.

[16] SEMATECH. Engineering Statistics Handbook. available at http://www.itl.nist.gov/div898/ handbook, downloaded on March 2011.

[17] The CAIDA DNS root/gTLD RTT Dataset, Support for the DNS root/gTLD RTT Dataset is provided by CAIDA, WIDE, the Univ. of Colorado and the Univ. of Auckland. http://www.caida.org/data/passive/dns_root_gtld_rtt_dataset.xml

[18] Vern Paxson (1999) "End-to-End Internet Packet dynamics", IEEE/ACM Transaction on Networking, Vol. 7(3), pp. 277-292