Performance Evaluation of TCP Vegas versus Different TCP Variants in Homogeneous and Heterogeneous Wired Networks

B. S. Yew[1], B. L. Ong[2], R. B. Ahmad[3]

***Abstract*—**A study on the performance of TCP Vegas versus different TCP variants in homogeneous and heterogeneous wired networks are performed via simulation experiment using network simulator (ns-2). This performance evaluation prepared a comparison medium for the performance evaluation of enhanced-TCP Vegas in wired network and for wireless network. In homogeneous network, the performance of TCP Tahoe, TCP Reno, TCP NewReno, TCP Vegas and TCP SACK are analyzed. In heterogeneous network, the performances of TCP Vegas against TCP variants are analyzed. TCP Vegas outperforms other TCP variants in homogeneous wired network. However, TCP Vegas achieves unfair throughput in heterogeneous wired network.

***Keywords*—**TCP Vegas, Homogeneous, Heterogeneous, Wired Network.

I. Introduction

TCP Vegas is proposed by [1]. TCP Vegas implements modification on the congestion control mechanism which is original proposed in [2]. The modification made in TCP Vegas congestion control mechanism is compatible with the TCP specifications that are stated in [3]. Hence, TCP Vegas is considered as an alternative implementation that interoperates with other TCP variants.

The previous studies [4][5][6] show that TCP Vegas achieves better throughput than other TCP variants. However, this is only true in a homogeneous network that solely involves TCP Vegas. In a heterogeneous network, TCP Vegas performance degrades. TCP Vegas was unable to achieve fair bandwidth allocation in the bottleneck link when competing with other type of TCP sources.

Hence, our aim of conducting this simulation experiment is to evaluate the performance of TCP Vegas in homogeneous and heterogeneous wired networks. In homogeneous wired network, TCP senders implement the same TCP source. In heterogeneous wired network, TCP senders implement different TCP sources and share the available bandwidth.

This paper is organized as follows. In section 2, we present literature review on previous works on TCP Vegas performances. In section 3, we explain the simulation setup that is used to conduct this simulation experiment using network simulator (ns-2). The simulation results that are obtained are discussed in section 4. In section 5, we discuss on the future work of the simulation experiment. Finally, in section 6, we conclude on the performances of TCP Vegas on both wired network.

II. Previous Works

L. S. Brakmo et. al [4] had proposed a new technique of congestion control detection and avoidance called TCP Vegas. From their study, they conclude that TCP Vegas implementation achieves better throughput and minimize the packet loss. However, this is only true when TCP Vegas is implemented in homogeneous network. When TCP Vegas is implements in heterogeneous network, the performance of TCP Vegas degrades.

A. D. Vendictis et. al [5] evaluate the performance of the TCP Vegas behavior in a heterogeneous network. From their study, they conclude that the fairness of TCP Vegas and TCP Reno cannot be achieved. Hence, they had proposed a new TCP congestion control mechanism called TCP NewVegas. TCP NewVegas shows improvement of fairness bandwidth allocation in heterogeneous networks.

III. Simulation Setup

The simulation tool used for this experiment is ns-2. The network topology used in the simulation experiment is shown in Fig. 1. The network topology used is a simple dumbbell topology which consists of TCP senders, TCP receivers and a pair of routers. The link between the TCP senders and router 1 is called as the sender link while the link between the TCP receivers and the router 2 is called the receiver link. The sender and receiver links represent a local area network (LAN). The link between router 1 and router 2 is called the bottleneck link. The bottleneck link represents a wide area network (WAN).

[1]B. S. Yew is a student conducting the research on enhancing the performance of TCP Vegas in computer network, University Malaysia Perlis (e-mail: bseokyew@ gmail.com).

[2]B. L. Ong is a senior lecturer (PhD) of the Computer and Communication Engineering Department, University Malaysia Perlis, Malaysia. Her research interest includes Wireless Communication, Simulation using ns-2 @ omnett, Mobility Management, Session Initiation Protocol (SIP), Cellular Network, IPV6 Network, and Wireless Mesh Network (WMN) (e-mail: drlynn@ unimap.edu.my).

[3]R. B. Ahmad is the dean (PhD, Associate Professor) of the Computer and Communication Engineering Department, University Malaysia Perlis, Malaysia. His research interest includes Computer, Telecommunication and Optical Network Modeling, Embedded System Applications Using Single Board Computer and GNU/Linux (e-mail: badli@unimap.edu.my).

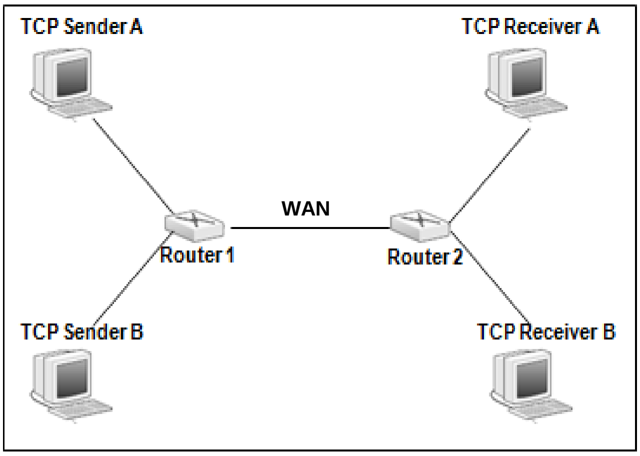

Fig. 1 Dumbbell wired topology

For the sender links and receiver links,we set four simulation environments as shown in Table I.

TABLE I
SIMULATION ENVIRONMENTS

Simulation Environment	TCP Sender (A&B)	TCP Receiver (A&B)
1	Ethernet	Ethernet
2	Ethernet	Fast Ethernet
3	Fast Ethernet	Fast Ethernet
4	Fast Ethernet	Ethernet

The sender links and the receiver links are a full wired duplex link. The bandwidth of the sender links that implement Ethernet LAN is 10Mbps and the link delay in the full duplex Ethernet LAN is set to 10ms. The bandwidth of the sender links that implement Fast Ethernet LAN is 100Mbps and the link delay of in the full duplex wired Fast Ethernet LAN is set to 1ms. The bottleneck link is a full wired duplex link with the capacity of 2Mbps that represents current bandwidth of MyREN network. The link delay of the bottleneck link is set to 50ms.

The link delay of the senders, receivers and bottleneck link is set to respective value so that the resulted bandwidth delay product is the same. The reason is because we want to customize the bandwidth delay product of the bottleneck link equal to the bandwidth delay product of the sender and receiver links. The bandwidth delay product is customized to be equal to minimize the unfairness sharing of available bottleneck capacity [7]. The simulation parameters of the network topology are showed in Table II.

TABLE II
SIMULATION PARAMETERS

Link	Bandwidth	Delay	Queue Limit	Window Size	Packet Size	Traffic Type On TCP Sender
Bottleneck	2 Mbps	50ms	50	17kB	1500B	-
Ethernet	10 Mbps	10ms	-	-	-	FTP
Fast Ethernet	100 Mbps	1ms	-	-	-	FTP

In this simulation experiment, the performances of TCP Vegas in homogeneous and heterogeneous wired network are evaluated.

In homogeneous network, both TCP sender implements the same TCP source. The TCP source used on the sender side in the homogeneous network is varied from Tahoe, Reno, NewReno, Vegas and SACK. For the receiver side, we set TCP Sink as the TCP source. There are five cases considered in simulation experiment of homogeneous wired network as shown in Table III. The performance of TCP variants in homogeneous network is evaluated in order to analyze the average throughput and average delay.

TABLE III
HOMOGENEOUS WIRED NETWORK

Case	TCP Sender A	TCP Sender B
1	Vegas	Vegas
2	Tahoe	Tahoe
3	Reno	Reno
4	NewReno	NewReno
5	SACK	SACK

In heterogeneous network, we set TCP Vegas as the TCP source on the sender side of the flow A. For flow B, the TCP source on the sender side is varied. There are four cases considered in the simulation experiments of heterogeneous wired network as shown in Table IV.

TABLE IV
HETEROGENEOUS WIRED NETWORK

Case	TCP Sender A	TCP Sender B
1	Vegas	Tahoe
2	Vegas	Reno
3	Vegas	NewReno
4	Vegas	SACK

The performance of TCP Vegas versus other TCP variants is evaluated in order to investigate the throughput fairness and average delay.

IV. RESULT AND DISCUSSION

In this section, we present the simulation results and the analysis for both homogeneous and heterogeneous wired network. For both wired network, TCP Vegas performs better in Fast Ethernet/ Fast Ethernet simulation environment.

A. Homogeneous Wired Network

For all the simulation environments in the homogeneous wired network, TCP Vegas performs better than other TCP variants. The average throughput of TCP Vegas outperforms the other four TCP variants. The average delay experience by TCP Vegas is the lowest among the other four TCP variants. The lower the delay, the faster the packet of data can be sent to the destination, this improves the efficiency of the network. Here, we discuss the reasons which contribute to the better performance of TCP Vegas.

TCP Vegas congestion control algorithm works based on the estimation of round trip time (RTT). TCP Vegas implements fine-grained timer in RTT estimation of the retransmission mechanism. The accuracy of the fine-grained timer is higher than the coarse-grained timer that is implemented in TCP Tahoe, TCP Reno, TCP NewReno and SACK. The fine-grained timer in RTT estimation of the retransmission mechanism minimizes the packet losses in a network [5]. This is because the estimation of RTT by using fine-grained times considers the timestamp and the timeout of packet that is sent from the sender to the receiver. TCP Vegas records the timestamp of the respective packet in a network. There are two situations considered, that is when duplicate and non duplicate acknowledgements (ACK) are received at the sender side. When duplicate ACKs are received at the sender side, TCP Vegas checks the difference of the current time interval and the timestamp recorded. As the difference is larger than the timeout, TCP Vegas retransmits the lost packet without having to wait for the third duplicate ACK that are transmitted by the receiver to the sender. TCP Vegas also detect any other packets that have been lost previously by retransmitting the respective packet [1]. This is when the sender received a non-duplicate ACK. As the time interval of the last packet that was sent to the receiver side is larger than the timeout value, TCP Vegas retransmits the respective packet. The retransmission mechanism of TCP Vegas minimizes the packet losses in a network. As the packet losses in a network are minimized, the resulted average throughput of TCP Vegas is optimized. Hence, this concludes why the average throughput of TCP Vegas outperforms other TCP variants.

As for the average delay, the RTT estimation in TCP Vegas congestion control algorithm enables the detection of congestion in a network at early stage. This means that TCP Vegas detects congestion faster than other TCP variants. Early stage detection of congestion reduces the delay in a network. This is because the network does not have to waste time on waiting for a packet lost to conclude that congestion occurs in a network. This explains why the average delay in a network that is implementing TCP Vegas source is less. The simulation results in the homogeneous wired network is tabulated in Table V and presented in Fig. 2 and Fig. 3.

TABLE V
HOMOGENEOUS WIRED NETWORK

Simulation Environment	Ethernet/ Ethernet (E/E)	Ethernet/ Fast Ethernet (E/F)	Fast Ethernet/ Fast Ethernet (F/F)	Fast Ethernet/ Ethernet (F/E)
	Average Throughput (kbps)			
Tahoe	1900.613	1910.099	1925.499	1910.099
Reno	1964.554	1968.373	1972.069	1968.373
NewReno	1971.822	1975.642	1980.816	1975.378
Vegas	1983.240	1986.000	1988.400	1986.000
SACK	1822.504	1830.388	1831.867	1831.744
	Average Delay (Second)			
Tahoe	0.554	0.543	0.518	0.544
Reno	0.571	0.560	0.542	0.559
NewReno	0.570	0.560	0.540	0.560
Vegas	0.205	0.190	0.136	0.190
SACK	0.552	0.536	0.512	0.539

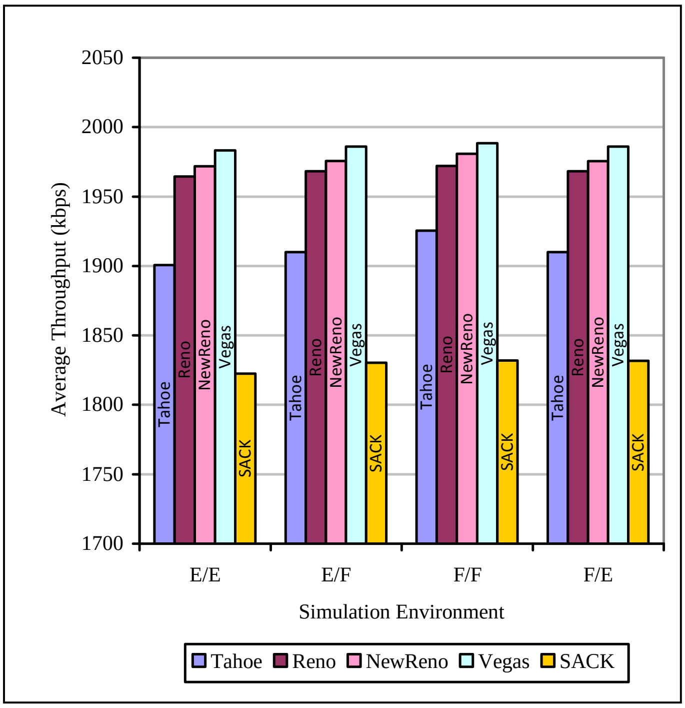

Fig. 2 Average throughput of TCP variants in homogeneous network

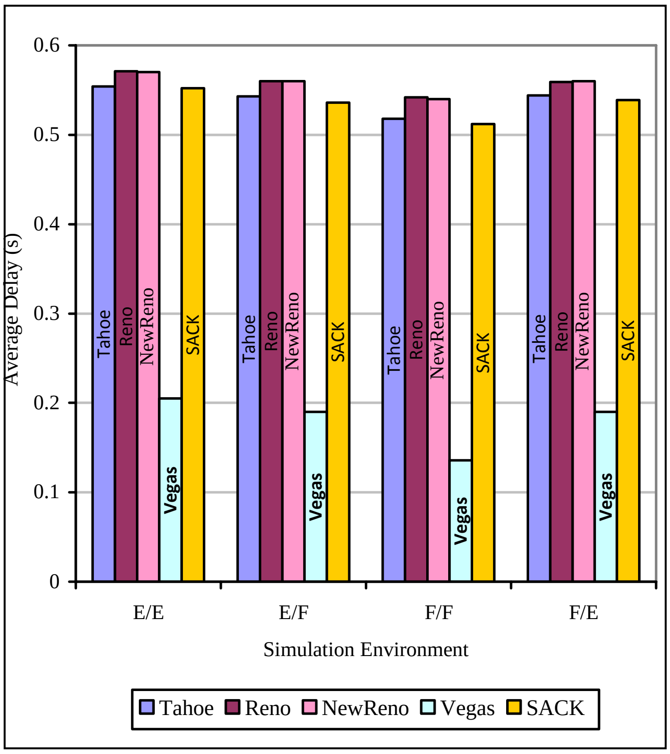

Fig. 3 Average delay of TCP variants in homogeneous network

B. Heterogeneous Wired Network

In heterogeneous wired network, the performance of TCP Vegas against TCP variants in a network where the bottleneck link is shared is evaluated. The average throughput fairness of TCP Vegas when sharing the bottleneck link with other TCP Vegas is analyzed.

For all the simulation experiments in the heterogeneous wired network, TCP Vegas is unable to achieve fair bandwidth allocation when sharing the bottleneck link with different TCP Variants.

TCP Vegas received unfair average throughput compared with TCP Tahoe, TCP Reno, TCP NewReno and TCP SACK. TCP Vegas received the most unfair throughput when it is sharing the bottleneck with TCP Reno and TCP NewReno. (TCP NewReno is a slight modification over TCP Reno. The congestion mechanisms implemented in TCP NewReno are slightly the same as TCP Reno except that TCP NewReno is able to detect multiple packet losses. Hence the results of TCP Reno and TCP NewReno do not vary much.) TCP Reno and TCP NewReno dominate most of the available bandwidth of the bottleneck link and being the most unfair to TCP Vegas. This is because the congestion avoidance mechanism of TCP Reno and NewReno is aggressive compared with TCP Vegas. In order to fully utilize the available bandwidth of the bottleneck link, TCP Reno and TCP NewReno continue to increase the window size until packet losses occur. As window size increases, the buffer size of TCP Reno and TCP NewReno in the bottleneck link also increases. The larger buffer size keeps more packets and hence dominates most of the available bandwidth in the bottleneck link. Thus, TCP Reno and TCP NewReno connections received higher throughput as the buffer size is larger.

On the other hand, TCP Vegas congestion avoidance mechanisms detect congestion at early stage. TCP Vegas detects congestion faster than TCP Reno and TCP NewReno. TCP Vegas congestion avoidance mechanism reduces the window size in order to maintain a smaller buffer queue size. The smaller buffer size minimizes the packet losses due to buffer overflows [8]. Thus, the TCP Vegas uses smaller bandwidth capacity in the bottleneck link. This concludes why TCP Vegas is unable to receive fair bandwidth allocation and the resulted throughput is significant low when sharing the bottleneck link with TCP Reno, TCP NewReno and other TCP variants.

Also, from all the simulation results, the value of delay experiences in the connection that is implementing TCP Vegas is smaller. This is due to the smaller buffer size that is maintained by TCP Vegas in the bottleneck link. The smaller buffer size resulted in smaller value of delay. On the other hand, in the connection that is implementing other TCP variants, the buffer size is larger compared to the buffer size in a connection that is implementing TCP Vegas. The larger buffer size experience unnecessarily long delays due to large queues size in the buffer. Hence, the value of delay is larger.

The simulation results in the heterogeneous wired network is tabulated in Table VI and presented in Fig. 4 and Fig.11.

TABLE VI
HETEROGENEOUS WIRED NETWORK

Simulation Environment	Ethernet/ Ethernet (E/E)	Ethernet/ Fast Ethernet (E/F)	Fast Ethernet/ Fast Ethernet (F/F)	Fast Ethernet/ Ethernet (F/E)
	Average Throughput (kbps)			
Vegas/ Tahoe	218.640 / 1705.707	80.640/ 1835.190	218.16 1722.34	80.640/ 1835.190
Vegas/ Reno	191.640 / 1778.765	77.280/ 1888.906	152.88 1819.42	77.280/ 1888.906
Vegas/ NewReno	190.320 / 1783.077	77.280/ 1893.834	88.20 1896.17	77.280/ 1893.834
Vegas/ SACK	246.360/1 1646.694	84.000/ 1790.099	134.280/ 1756.589	84.000/ 1790.099
	Average Throughput Fairness (%)			
Vegas/ Tahoe	10.932/ 85.256	4.032/ 91.760	10.908/ 86.115	4.032/ 91.760
Vegas/ Reno	9.582/ 88.938	3.864/ 94.445	7.644/ 90.971	3.864/ 94.445
Vegas/ NewReno	9.516/ 89.154	3.684/ 94.692	4.410/ 94.809	3.684/ 94.692
Vegas/ SACK	12.318/ 82.335	4.200/ 89.510	6.714/ 87.830	4.200/ 89.510
	Average Delay (Second)			
Vegas/ Tahoe	0.249/ 0.284	0.242/ 0.273	0.220/ 0.265	0.242/ 0.273
Vegas/ Reno	0.258/ 0.285	0.256/ 0.276	0.241/ 0.266	0.256/ 0.276
Vegas/ NewReno	0.260/ 0.285	0.256/ 0.276	0.255/ 0.269	0.256/ 0.276
Vegas/ SACK	0.231/ 0.284	0.231/ 0.275	0.218/ 0.265	0.231/ 0.275

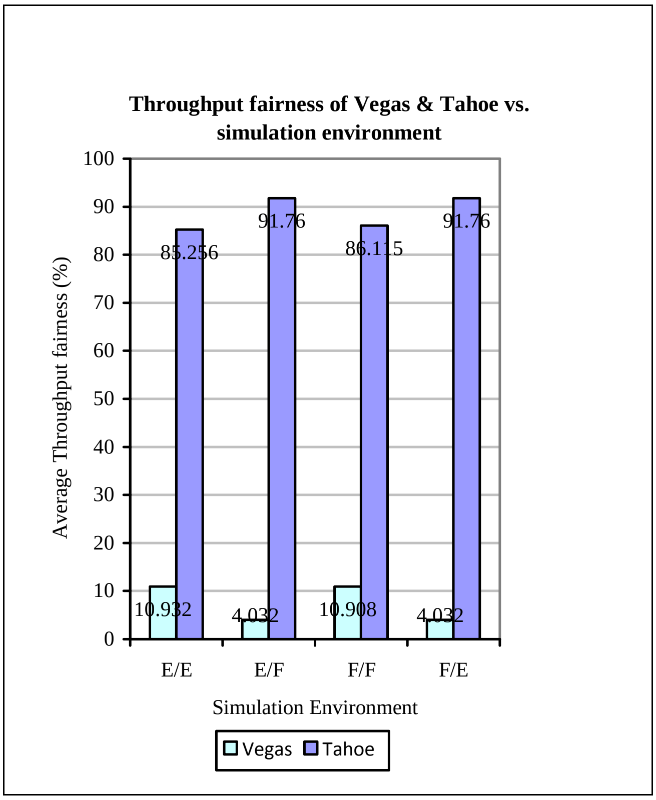

Fig. 4 Throughput Fairness of Vegas & Tahoe vs. Simulation Environment

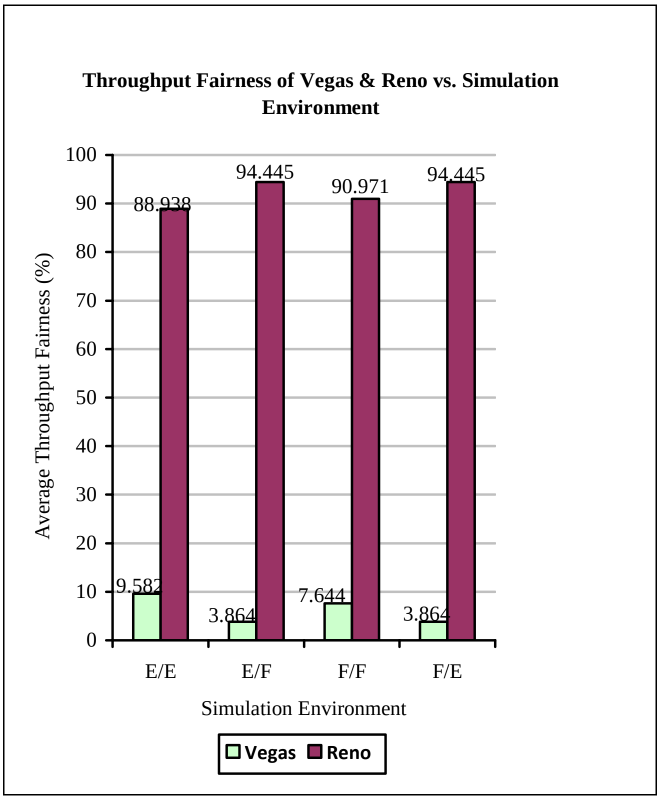

Fig.6 Throughput Fairness of Vegas & Reno vs. Simulation Environment

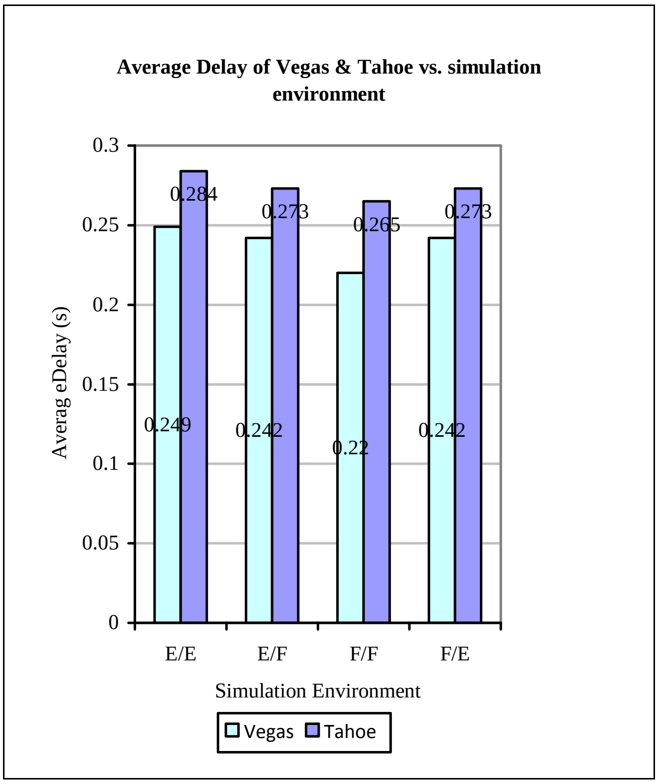

Fig. 5 Average Delay of Vegas & Tahoe vs. Simulation Environment

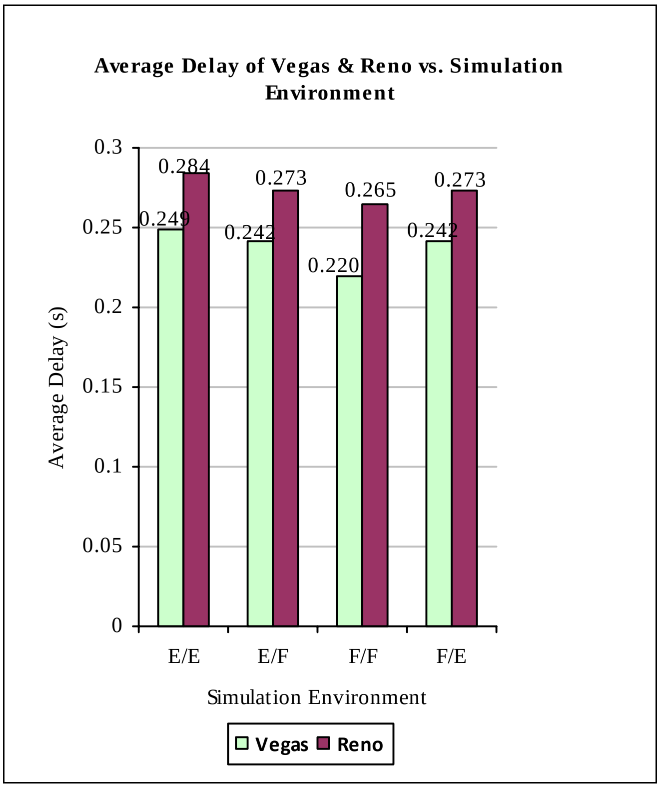

Fig. 7 Average Delay of Vegas & Reno vs. Simulation Environment

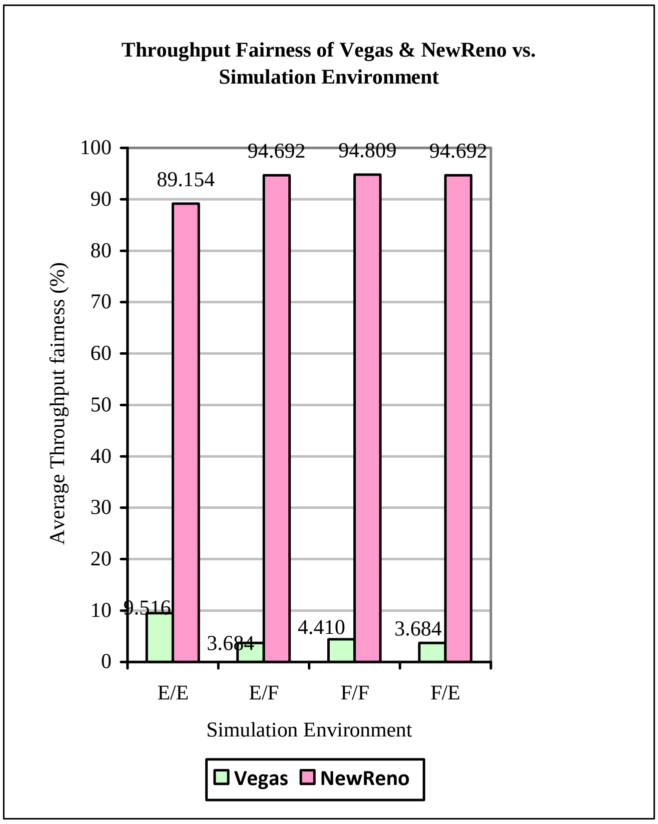

Fig. 8 Throughput Fairness of Vegas & NewReno vs. Simulation Environment

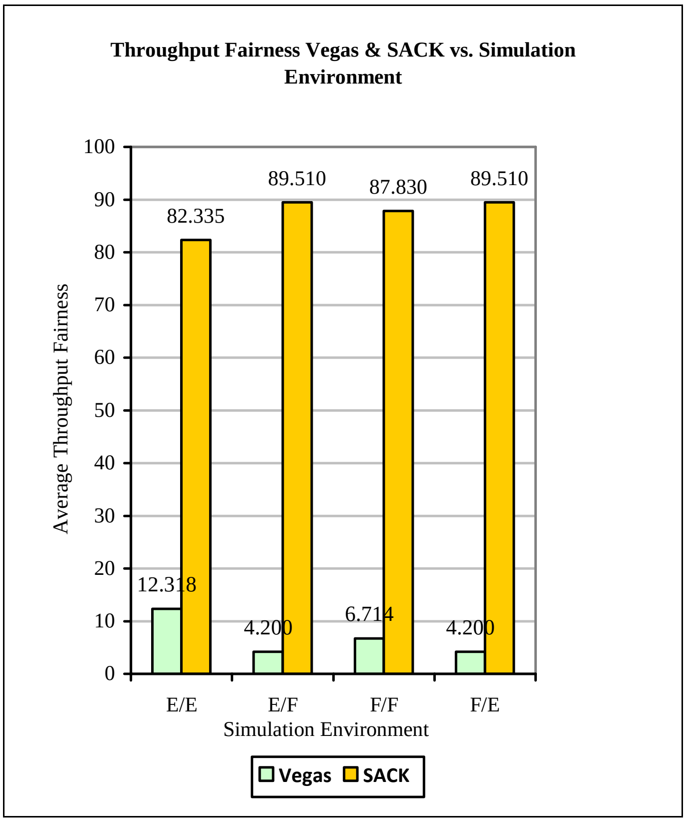

Fig. 10 Throughput Fairness of Vegas & SACK vs. Simulation Environment

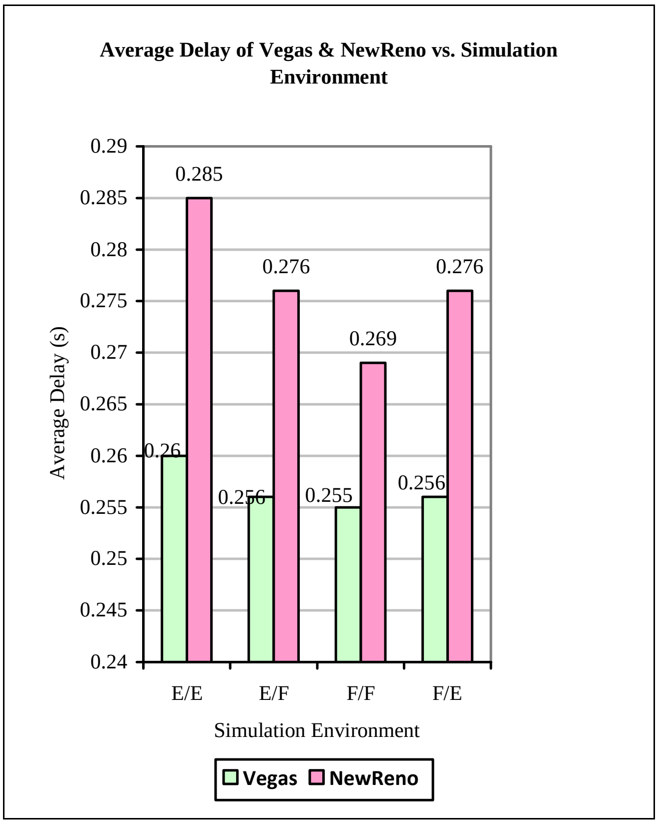

Fig. 9 Average Delay of Vegas & NewReno vs. Simulation Environment

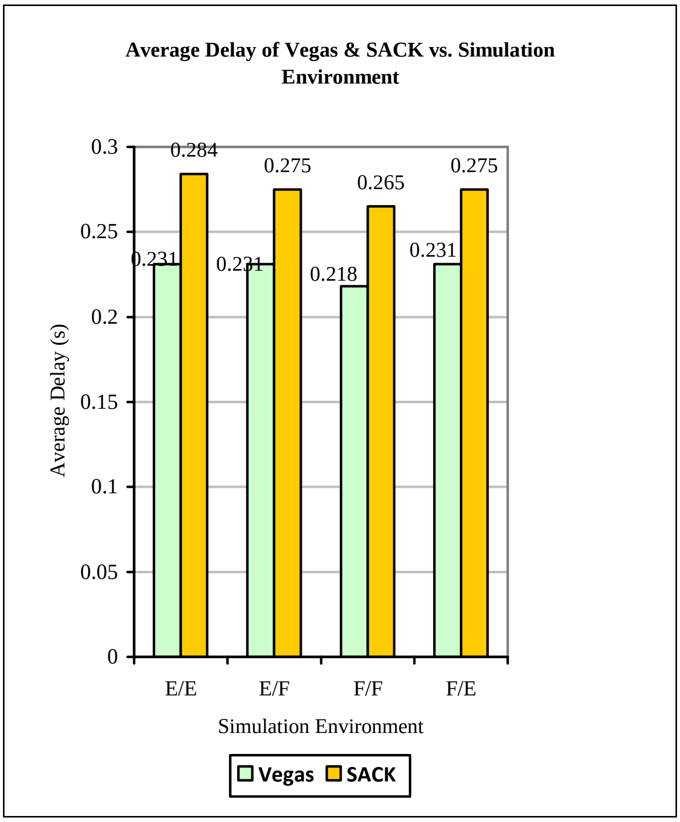

Fig. 11 Average Delay of Vegas & SACK vs. Simulation Environment

V. FUTURE WORK

This simulation experiments analyze the performance of TCP Vegas versus other TCP variants in term of average throughput, throughput fairness and average delay in wired network connection. The results of this simulation experiments prepared a comparison medium for the performance evaluation of enhanced-TCP Vegas in wired network and for wireless network that will be implemented for future work.

For future work, we propose to enhance the performance of TCP Vegas in wireless network. We aim to optimize the performance of the wireless IPv6 network by implementing the enhanced-TCP Vegas algorithm. The enhanced-TCP Vegas algorithm can be implemented not only in wireless IPv6 network but can also be implemented in the existing IPv4 network. This enhanced-TCP Vegas algorithm is to be developed and is believed that it can improve the TCP performance in wireless IPv6 environment.

ACKNOWLEDGMENT

We would like to thank UniMAP for provided useful literature materials and grant FRGS 1/2010 (9003-00216) for the financial support of this research.

REFERENCES

[1] L. S. Brakmo, S. W. O'Malley, L. L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," *In the Proceedings of the ACM SIGCOMM,* vol. 24, pp. 24-35, Oct 1994.

[2] V. Jacobson, "Congestion Avoidance and Control", *In the Proceddings of the SIGCOMM'88 Symposium*, pp. 314-329, Aug. 1988.

[3] M.Allman, V.Paxson and W.Stevens, "TCP Congestion Control", *Request for Comment 2581, Internet Engineering Task Force*, April 1999.

[4] L. S. Brakmo, L. L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", *In the Proceedings of IEEE Journal On Selected Areas In Communications,* vol. 13, no. 8, pp. 1465- 1480, Oct. 1999.

[5] A. D. Vendictis, A. Baiocchi, M. Bonacci, "Analysis and Enhancement of TCP Congestion Control in a mixed TCP Vegas and TCP Reno Network Scenario", *Performance Evaluation 53,* pp. 225-253, 2003.

[6] C. M. Tsang, K. C. Chang, "A Simulation Study on the Throughput Fairness of TCP Vegas", *In the Proceedings of the 9th IEEE International Conference on Networks,* vol. 21, pp.469-474 , Oct. 2001.

[7] D. Katabi, M. Handley, "Congestion Control for High Bandwidth-Delay Product Networks", *In the Proceedings of SIGCOMM'02,* Feb. 2007.

[8] J. Mo, R. J. La, V. Anantharam, J. Walrand, "Analysis and Comparison of TCP Reno and Vegas", *In the Proceedings of IEEE INFOCOM 1999,* vol.3, pp.1556-1563, March 1999.