World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:6, No:2, 2012

Optimizing TCP Vegas' Performance with Packet Spacing and Effect of Variable FTP Packet Size over Wireless IPv6 Network

B. S. Yew , B. L. Ong , R. B. Ahmad

***Abstract*—**This paper describes the performance of TCP Vegas over the wireless IPv6 network. The performance of TCP Vegas is evaluated using network simulator (*ns*-2). The simulation experiment investigates how packet spacing affects the network delay, network throughput and network efficiency of TCP Vegas. Moreover, we investigate how the variable FTP packet sizes affect the network performance. The result of the simulation experiment shows that as the packet spacing is implements, the network delay is reduces, network throughput and network efficiency is optimizes. As the FTP packet sizes increase, the ratio of delay per throughput decreases. From the result of experiment, we propose the appropriate packet size in transmitting file transfer protocol application using TCP Vegas with packet spacing enhancement over wireless IPv6 environment in *ns*-2. Additionally, we suggest the appropriate ratio in determining the appropriate RTT and buffer size in a network.

***Keywords*—**TCP Vegas, Packet Spacing, Packet Size, Wireless IPv6, *ns*-2

I. Introduction

TCP congestion control algorithm provides significant performance gains over wired network but not in wireless network [1][2]. Originally, TCP congestion control algorithm (CCA) is designed to operate in wired network with the assumption that the network congestion is indicated when packet loss occur. CCA's assumption is violates when it is used in wireless network [3]. The reason is due to the wireless losses in the wireless links. Wireless links are characterized by a very high and continuously varying bit error rate (BER). This high BER together with a large round trip times (RTT) and the fact that often small sized packets are exchanges over wireless links are one of the factors that limit the utilization of the wireless network [4][5][6].

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).

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).

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).

Additionally, we listed the parameters that affect the performance of TCP congestion control algorithm in wireless network [2]:

- Limited capacity – The data rates in wireless network is limited.
- Long Round Trip Times – Wireless link exhibit longer transmission delay. This affect TCP throughput and increase the interactive delays perceived by the user.
- Random losses – High bit error rate (BER) that causes packet corruption due to fading channels, shadowing etc.
- Host mobility –Wireless network enable the host to move randomly. When a host is moving from one cell to another, handover needs to be performed. The handover operation requires the exchange of data between the previous network and the latest network where the host is located. Additionally, frequent disconnections occur due to the bad link quality between the previous network and the latest network.
- Short flows – Most services offered in wireless network include the transmission of rather small amounts of data. This means that when the application layer protocol opens a TCP connection for the transfer, there is a very large probability that the whole transfer is completed while the TCP sender is still in the slow start state. Therefore the TCP connection never manages to fully utilize the available bandwidth.

Moreover the queuing theory states that bursty traffic produces higher queuing delays, more packet losses, and lower throughput. Many researchers have observed that TCP's congestion control algorithm can lead to bursty traffic flows on wireless network [7][8].

In our work, we investigate TCP Vegas as the congestion control algorithm with the adaption of packet spacing and effect of the different TCP packet sizes on the wireless IPv6 environment. TCP Vegas congestion control algorithm provides significant performance gains over most TCP variants. TCP Vegas implements congestion control by using round trip time estimation (RTT). TCP Vegas does not force the network to drop packets in order to achieve the available network bandwidth. TCP Vegas is able to achieve more readily identify segments lost due to corruption, retransmitting lost segments sooner that other TCP variants. Despite the advantages of TCP Vegas over other TCP variants, we observed some unusual behavior with TCP Vegas [9]. We observed that TCP Vegas' performance decreased significantly when it is used over wireless IPv6 network. A major disadvantage of TCP Vegas is its path asymmetry. The sender makes decisions on the transmission rate based on the RTT measurements. In wireless links that exhibits wireless losses, TCP Vegas cannot accurately indicate the actual congestion due to packet lost. In general TCP Vegas does not seems to be a good choice for a wireless network as it was shown that it does not achieve high utilization because of its shorter slow-start phase and longer congestion avoidance phase[10].

Hence, our aim of conducting the simulation experiment is to evaluate the performance of TCP Vegas with packet spacing adaption. Packet spacing is a delay that space the data sent into a network over the round trip times. Packet spacing is implemented at the TCP's sender side. At the sender side, instead of transmitting packets immediately upon receipt of an akcnowledgement, the sender insert the packet spacing to delay the transmittion of packets. At the duration where the transmission is spaced, no new packet is sent. There is a minimal queuing of packets at the bottleneck link. Minimal queuing smoothing the bursty traffic behaviour before the bottleneck becomes saturated. Once the bottleneck is saturated, packet is dropped and the performance of TCP degrades.

We also investigate the effect of different TCP packet sizes to optimize the transmission rate of the data. The Internet protocol (IP) transmits various forms of packet sizes. The commonly used packet sizes are 64 bytes, 128 bytes, 256 bytes, 512 bytes, and 1024 bytes. Different packet sizes can cause variation in packet delay and throughput. Selecting inappropriate packet sizes can cause higher packet delay and packet loss. Consequently, higher packet delay and packet loss can degrade the wireless IPv6 performance [11].

This paper is organized as follows. In section 2, we present the background and the issues to be covered in this article. 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. Then, we conclude this paper in Section 5.

II. Background

In this section, we discuss the TCP protocol and TCP Vegas congestion control.

(i) Transmission Control Protocol (TCP)

TCP provides a connection oriented and reliable byte stream service. Connection oriented means TCP sender must establish a reliable connection with TCP receiver to exchange data. The established connection is a full duplex link, meaning that supports a pair of byte streams, one flowing in each direction. TCP uses sliding window flow control mechanism for data flow that allow the TCP receiver to limit how much data the TCP sender can transmit. TCP also implements a congestion-control algorithm to minimize the congestion in a network.

(ii) Sliding Window Flow Control Mechanism

TCP is a sliding window based protocol. Sliding window is the method of flow control in a TCP connection. The sliding window algorithm places a buffer between the application and the network data flow. The purpose of sliding window is to prevent the TCP sender to send too many packets to over flow the bottleneck link. The sliding window size is the maximum amount of data TCP sender can send without having to wait for ACK.

Two important parameters are used for TCP to achieve the flow control by using the sliding window algorithm. The first parameter is the congestion window (CWND) which controls the number of packets a TCP flow may have in the network in any given time. The second parameter is the receiver advertised window (ADWN) size which basically tells the TCP sender what is the current buffer of TCP receiver.

(ii) TCP Vegas Congestion Control

TCP congestion control addresses the problem when too many sources sending too much data for the network to handle. TCP congestion control tries to minimize an overflow of the receiver's buffer. In other word, TCP congestion control minimizes packet losses and optimizes the network throughput and efficiency. Many of TCP variants have been proposed [12]. Among these TCP variants, TCP Vegas is claims to have a better throughput [13]. TCP Vegas uses bandwidth estimation scheme to avoid congestion rather than waiting for congestion to happen to invokes its congestion control mechanism [14]. TCP Vegas uses the difference in the expected and actual flow rates to estimate the available bandwidth for the network [15]. Like other TCP variants, TCP Vegas control the amount of data injected into the network by using 2 phases: slow start and congestion avoidance. Fig. 1 shows the mode of congestion window increase in slow start phase and congestion avoidance phase.

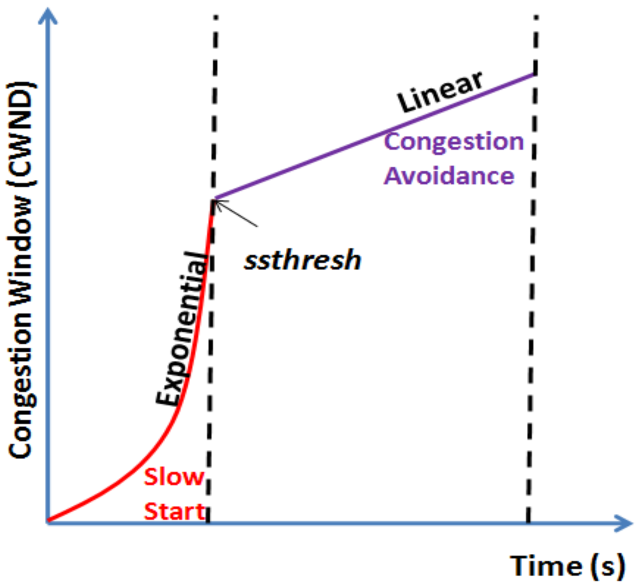

Fig. 1 Congestion window growth in slow start phase and congestion avoidance phase

A. Slow-start

In the slow start phase [16][17], TCP Vegas sender increases the congestion window exponentially in order to reach the available bandwidth as quickly as possible.

$$CWND(n) = \begin{bmatrix} IW, \text{when n} = 0 \\ CWND(n-1) \times 2^{n}, \text{when n} \geq 1 \end{bmatrix} \quad (1)$$

The exponential increase of congestion window (CWND) implies that the TCP sender is bursting data at twice the bottleneck rate, causes the traffic queuing at the bottleneck to increase. Since buffer size at the bottleneck is at the minimum of the receiver advertised window size, the burstiness behavior will cause the router to drop packets when the congestion window size exceeds the size of the router's buffer.

B. Congestion avoidance phase

In congestion avoidance phase, TCP Vegas tries to maintain high throughput without causing congestion. The congestion window size is reduces to one-eighth of its current size. The congestion avoidance phase increase congestion window in linear mode. TCP Vegas increase the congestion size by $\frac{1}{CWND}$, decreased by one segment or left unchanged, depending on two threshold α and β.

$$CWND(n) = \begin{bmatrix} CWND(n-1) + \dfrac{1}{CWND}, & \text{if } \Delta < \alpha \\ CWND(n-1) - 1, & \text{if } \Delta > \beta \\ CWND(n-1), & \alpha < \Delta < \beta \end{bmatrix} \quad (2)$$

III. Problem In TCP Vegas

In this section, we present the reasons that contribute burstiness in the wireless network. Fig.2 illustrates the problem exhibits in TCP Vegas.

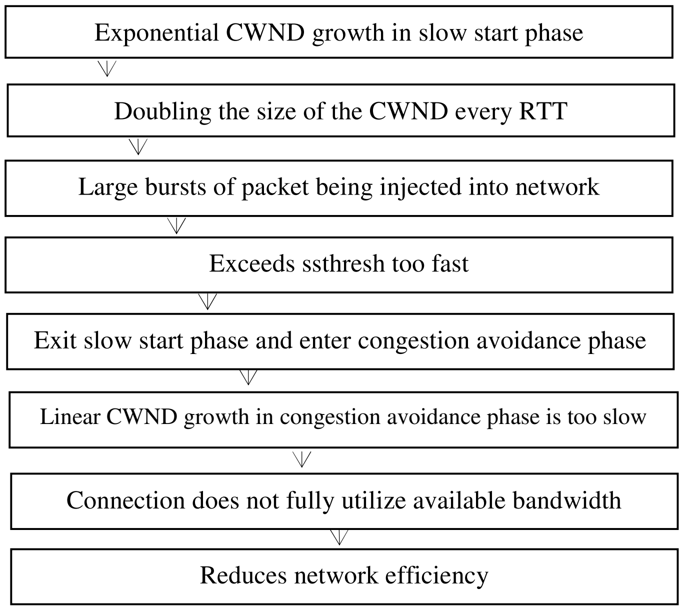

Fig. 2 Problems in TCP Vegas

In the slow start phase, doubling the sending rate too fast also causes the TCP connection to reach slow start threshold (ssthresh) in a short time. Reaching slow start threshold (ssthresh) too fast may lead to early transition of slow start phase into congestion avoidance phase. Early transition into the congestion avoidance phase degrades TCP performance. The reason is due to the reduction of congestion window to one-eighth in the congestion avoidance phase. Additionally, it is also difficult to estimate the optimum available bandwidth in the wireless network that involves mobile movement.

In the congestion avoidance phase, since the linear increase of CWND in (2) is smaller than the linear decrease of CWND, the connection requires longer duration to ramps up to the available bandwidth. Moreover, many TCP sessions are short lived. For the case of smaller data size that is to be exchanged between networks, the data transfer is completed before the connection reaches its available bandwidth. This means that the congestion window size may never reach the optimal level before the session is terminated [15]. This causes the waste of bandwidth use and limits the utilization of the wireless network.

IV. Enhancement In TCP Vegas

The exponential growth of congestion window in slow start phase doubles the sending rate of data. This results in large bursts of packets being injected into the network. The bursty traffic at the bottleneck link causes large fluctuations in the network delay due to the increasing queuing delay. The increasing queuing delay due to the bursty traffic increases the round trip time (RTT) estimation used by the TCP Vegas. As RTT increases with respect to the bursty traffic, network delay is increases accordingly and causes unusual long delay in a network.

In order to minimize bursts of packets from being injected into the network, we proposed to insert a spacing delay between the transmissions of each segment. The spacing delay is called packet spacing. Packet spacing is inserted based on the current RTT measurement and the current congestion window size. The packet spacing equation is given in (3):

$$\text{Packet Spacing} = \frac{\text{Current RTT measurement}}{\text{Current CWND size}} \tag{3}$$

We proposed to space the transmission in a TCP connection using the current RTT measurement instead of base RTT. This is due to the network condition in wireless network that vary with time. This is due to the mobility of hosts in wireless network. The hosts move randomly in the wireless network. Hence, using the current RTT measurement in the packet spacing calculation in (3) increase the accuracy if RTT estimation in TCP Vegas. In the other hand, the host in wired network is located at stationary address; it does not involve change of location.

The packet spacing will evenly spread the transmission of a window of packets across the entire duration of round trip times. Packet spacing is implements by the TCP sender. Instead of transmitting packets immediately after receiving the acknowledgment from the TCP receiver, TCP sender insert the packet spacing to delay the transmission at the rate defined by (3) accorded to network condition.

During the slow start phase, doubling the size of congestion window causes bursty packets to arrive all at once. As a result, queuing delay grow proportionally with the load. By implementing the packet spacing, the bursty traffic at the bottleneck is minimizes by evenly space out the traffic. The bottleneck make use the given packet spacing duration to forward the large burst of data to the TCP receiver. There will be a minimal of queuing until all the data matches the bottleneck capacity. By doing so, packet lost is minimizes.
Once packet lost is minimizes, early termination of slow start phase can be avoided. The utilization of the bottleneck link in the connection is then optimizes to reach its available bandwidth.

V. Simulation Setup

The network topology used in our simulations in shown in Fig.3. The network topology is designed for IPv6 environment. The network topology is simulated by using ns-2. The simulation parameter is tabulated in Table 1.

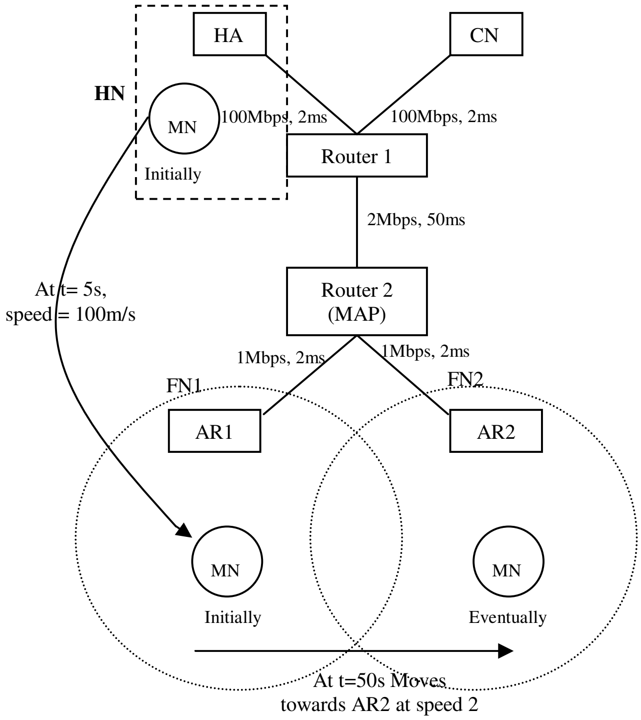

Fig. 3 Network topology

The links in the network topology are set to full duplex link. CN and HA are connected to node N1 using a wired link with 100Mbps, which represents high speed Ethernet LAN. The link delay is set to 2ms. The bottleneck link that connected N1 and MAP is set to 2Mbps, which represent the MyREN network. The link delay of bottleneck link is set to 50ms.The bottleneck link represent the wide area network (WAN). AR1 and AR2 are the base stations that act as gateway between wired network and wireless network. AR1 and AR2 are connected to the MAP with the bandwidth of 1Mbps and link delay of 2ms, which represent an 802.11b technology. The MN moves from AR2 to AR2 at normal human speed, 1m/s. For our simulations experiments, we implement FTP traffic application and TCP variants as the transport protocol. CN is attached with TCP source while MN is attached with TCP sink agent. The total simulation duration is 100s.Initially, MN is located in HN. At t=5s, MN moves towards FN1 at a really fast speed of 100m/s. MN reached FN1 and attached to AR1 and configure its new IP address. The new IP address of MN is the address of MAP (RCoA). MN updates HA of its RcoA by sending BU to HA. Packet forwarded from CN will be directed to RcoA and transmitted to MN.

In the MAP domain, MN is firstly attached to AR1 (PAR). The on-link address of MN is based on the IP address of AR1, which is LcoA1. At t=30s, MN performs handover and start to moves towards FN2. MN will enter the overlapping region during its movement towards FN2. The overlapping region of AR1 and AR2 are set to be 80 meters apart with free space environment in between. In the overlapping region, MN will configure a new on-link IP address from AR2, which is LcoA2. At this moment, MN will exhibit two on-link IP address (LcoA1 and LcoA2). Once MN completely moves out from the coverage of FN1, the LcoA1 is deleted and only LcoA2 is available. The changes of on-link address are only updated at MAP. As MAP is updated with the latest on-link IP address, MAP forwards the packets from the CN towards MN.

TABLE I
SIMULATION PARAMETER

Node/Link	Bandwidth (Mbps)	Delay (ms)
Home Agent (HA)	100	2
Correspondence Node (CN)	100	2
Bottleneck Link (WAN)	2	50
AR1	1	2
AR2	1	2

VI. RESULT AND DISCUSSION

In this section, we present the simulation results and discussion for the TCP Vegas with packet spacing adaption in the wireless IPv6 network. The simulation is divided into two parts. The first part compares the performance of standard TCP Vegas and TCP Vegas with packet spacing by varying the packet size. For the second part, we investigate the effect of variable RTT on the performance of TCP Vegas with packet spacing with a fixed packet size and buffer size.

(i) Varying FTP Packet Size with Fixed RTT

In the first part of our simulation experiment, we investigate how different FTP packets affect the handover delay by using the packet spacing implementation in TCP Vegas. The different packet sizes used in the simulation are 128 bytes, 256 bytes, 512 bytes, and 1024 bytes. Since 1500 bytes is the maximum transfer unit (MTU) of high speed Ethernet LAN [18][19], the FTP packet size is vary within this boundary. The buffer size is set to the default value, 50 packets. The numerical result of standard TCP Vegas is presented in Table II.

TABLE II
NUMERICAL RESULT OF STANDARD TCP VEGAS USING FIXED BUFFER SIZE

Packet Size (byte)	Standard TCP Vegas			
	Network Delay (ms)	Network Throughput (kbps)	Network Efficiency (%)	Delay w.r.t Throughput
64	52.266	18.642	99.121	2.804
128	54.104	35.072	99.128	1.543
256	65.078	67.16	99.145	0.969
512	64.890	228.065	99.491	0.285
1024	76.587	123.699	99.011	0.619

TABLE III
NUMERICAL RESULT OF TCP VEGAS WITH PACKET SPACING USING FIXED BUFFER SIZE

Packet Size (byte)	TCP Vegas with Packet Spacing			
	Network Delay (ms)	Network Throughput (kbps)	Network Efficiency (%)	Delay w.r.t Throughput
64	54.104	23.875	99.397	2.266
128	53.886	34.857	99.065	1.546
256	57.676	67.420	99.276	0.855
512	72.2201	316.006	99.402	0.229
1024	76.7154	227.901	99.143	0.337

The numerical result is plot into graphical output. From the graph, we can observe that as the packet size increases the network delay increases. The reason is because as the size of packet increases, the network needs more time to transmit the packet [11].

The delay with respect to throughput is calculated by the total of network delay divided by the throughput gained. The equation is shows in (4)

$$\text{Delay w.r.t throughput} = \frac{\text{Network Delay}}{\text{Network Throughput}} \tag{4}$$

From graph in Fig.4, we observe that as the packet size increases, the delay w.r.t throughput decreases accordingly. However, this inversely proportional relationship of packet size and delay w.r.t throughput is only true until the packet size of 512 bytes. The lower the delay w.r.t throughput represents that the TCP connection exhibit lesser network delay but produce higher throughput. Hence, the network which exhibits the lowest value of delay w.r.t throughput represents the network with optimum performance.

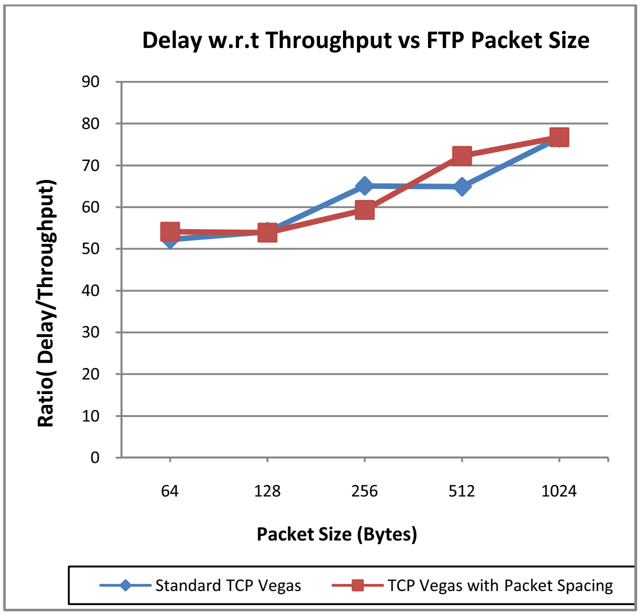

Fig. 4 Ratio of delay with respective to throughput with variable FTP Packet Size

From the result in Table III, the packet size of 512 bytes exhibits the lowest value of delay w.r.t throughput. This implies that packet sizes of 512 bytes is the suitable packet size to send FTP packet size over the wireless IPv6 network by using TCP Vegas with Packet Spacing implementation. We can observe that for packet size that is equal to 1024 bytes the delay w.r.t throughput increases as the packet size increase. This implies that the network performance decreases starting for the packet size equal and larger 1024 bytes. In ns2, the maximum value of FTP packet size that is set in the ns library is equal to 1000 bytes. For the packet size that exceeds 1000 bytes, ns2 will performs fragmentation in order to forward the packets. Fragmentation is the operation where the packet is broken up into smaller pieces. The fragmentation operation of packet is illustrated in Fig.5.

Case (1): Packet Size = 1024 bytes

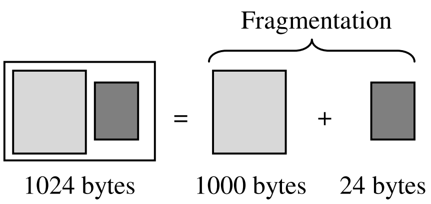

Fig. 5 Data fragmentation

Fragmentation causes in increase of network delay in the network but decreases the network throughput of a connection. The reason is because by the sending the same amount of data with smaller packet size, more overhead are sent over the IP.

(ii) Varying RTT with fixed FTP Packet Size

In order to study the effect of variable RTT the performance of TCP Vegas with packet spacing implementation, we vary the size of the RTT from 40ms until 150ms. RTT is varies in order to analyses how far packet spacing affect the packet transmission rate and the resulted throughput. We set the simulation time to 500s in order to investigate the impact of burstiness in long run environment. We set the packet size to 512 bytes to achieve optimum performance. Table IV presents the average result of the packet spacing in TCP Vegas with variable RTT. The graphical outputs for the network throughput and cumulative throughput are generated directly by using GNUPLOT as a function of time. Refer to Fig.6 to Fig.8

TABLE IV
NUMERICAL RESULT OF TCP VEGAS WITH PACKET SPACING (FIXED PACKET SIZE)

RTT	TCP Vegas with Packet Spacing			
	Average Network Delay (ms)	Average Network Throughput (kbps)	Average Network Efficiency (%)	Delay w.r.t Throughput
40	41.946	248.201	99.894	0.169
50	46.856	215.925	99.878	0.217
60	58.545	848.044	99.955	0.069
70	56.639	170.918	99.847	0.331
80	67.896	835.836	99.904	0.081
90	66.358	141.648	99.835	0.468
100	77.805	815.907	99.943	0.095
110	83.046	817.684	99.888	0.102
120	87.906	804.446	99.897	0.109
130	92.774	793.543	99.917	0.117
140	97.628	799.709	99.923	0.122
150	102.627	777.150	99.872	0.132

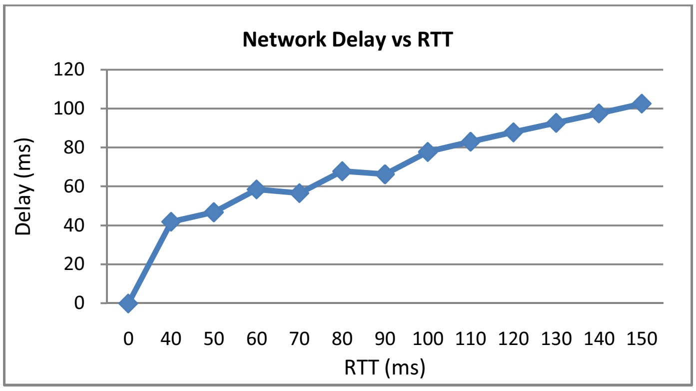

Fig. 6 Network delay versus RTT

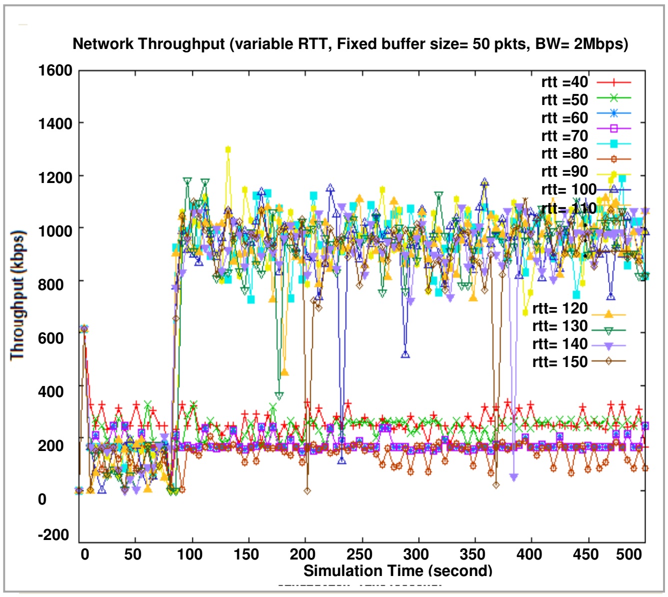

Fig. 7 Network throughput for the variable RTT by implementing packet spacing

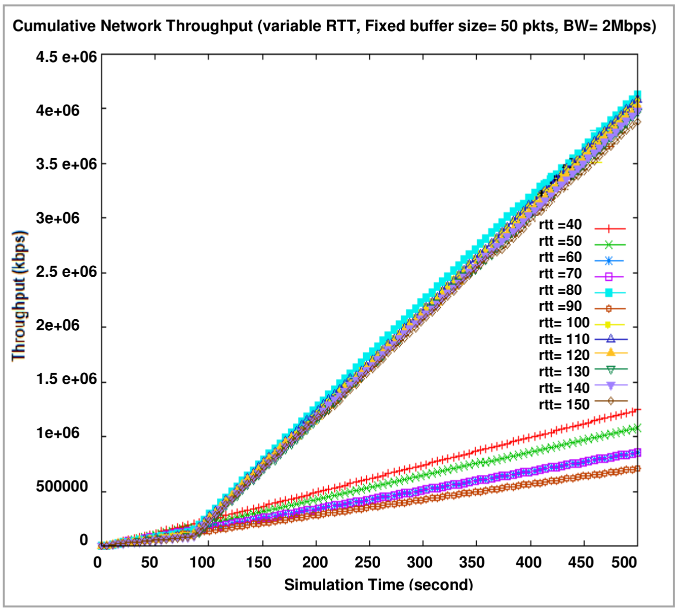

Fig. 8 Cumulative network throughput for the variable RTT by implementing packet spacing

From Fig.6, we can observe that as RTT increases, network delay increase accordingly. As expected, with variable RTTs, burstiness in a connection increases as the connection exhibit longer RTT. The burstiness increases the queuing delay. Additionally, connection with longer RTT exhibit longer transmission delay.

Fig.7 and Fig.8 represents the network throughput and cumulative throughput of TCP Vegas with packet spacing when variable RTTs are used. We can observe that the network throughput fluctuates. There are two boundary of results exists. We called these two boundaries as lower boundary and upper boundary. The RTT that exists in lower boundary are RTT that is equal to 40ms, 50ms, 60ms, 70ms, 80ms and 90ms. The lower boundary represents that the packet spacing perform worse when the stated RTT is used in the connection. For the upper boundary, it represents that the packet spacing outperforms when the RTTs within the boundary is used. The RTT that exists in the upper boundary are RTT that is equal to 100ms, 110ms, and 120ms, 130ms, 140 ms and 150ms.

In a window-based protocol such as TCP, the performance of TCP is dependent on the RTT, bottleneck bandwidth and buffer size. The amount of data that fill a TCP window is represented by the product of bandwidth and the RTT [20]. The amount of data is called Bandwidth-delay product (BDP). Bandwidth-delay product (BDP) and the TCP Receive Window limit our connection to the product of the RTT and the bandwidth. The bandwidth-delay product (BDP) of the network is given in equation (5):

$$\text{BandwidthDelayProduct(BDP)} = \frac{\text{RTT} \times \text{BottleneckBandwidth}}{\text{MaximumTransferunit}} \qquad (5)$$

In our work, we set the buffer size to a fixed value. By varying the RTT, we obtain the boundary for RTT value that allows packet spacing to outperform.

Let us define the ratio between buffer size and bandwidth-delay product of the network in equation (6):

$$X = \frac{BufferSize}{BDP} \qquad (6)$$

The ratio X is used as a parameter to determine the appropriate buffer size for the given topology with the variable RTTs. The ratio X for upper boundary of RTT is used as a parameter to determine the buffer size. The buffer size is directly proportional to the RTT in a connection.

TABLE V
THE RATIO BETWEEN BUFFER SIZE AND BANDWIDTH-DELAY PRODUCT OF THE NETWORK IN THE UPPER BOUNDARY

RTT(ms)	BDP	Buffer Size (pkts)	Ratio,X	Mean X
100	48.828	50	1.024	0.835
110	53.711	50	0.931	
120	58.594	50	0.853	
130	63.477	50	0.788	
140	68.359	50	0.731	
150	73.242	50	0.683	

TABLE VI
THE RATIO BETWEEN BUFFER SIZE AND BANDWIDTH-DELAY PRODUCT OF THE NETWORK IN THE UPPER BOUNDARY

RTT(ms)	BDP	Buffer Size	
		Calculated	Rounded
40	19.53125	16.309	16
50	24.414063	20.386	20
60	29.296875	24.463	24
70	34.179688	28.540	29
80	39.0625	32.617	33
90	43.945313	36.694	37
100	48.828125	40.771	41
110	53.710938	44.849	45
120	58.59375	48.926	49
130	63.476563	53.003	53
140	68.359375	57.080	57
150	73.242188	61.157	62
Given ratio, X= 0.835			

VII. CONCLUSION

In this paper, we present the result of simulation experiment on wireless IPv6 network by implementing TCP Vegas with the packet spacing adaption. Packet spacing improves the performance of TCP Vegas. The burstiness traffic at the bottleneck link is minimizes by evenly space out the traffic.

In the simulation experiment, different FTP packet sizes are sent over the wireless IPv6 environment. The simulation result shows that among the different packet sizes, packet size 512 bytes is the most suitable size to send the FTP packet in terms of small loss rate, low delay, and high bottleneck link utilization). Thus, we propose that packet size is packetized into 512 bytes when it is sent to the wireless IPv6 network. In order to generate throughput in upper boundary, we suggest that the ratio, X= 0.835 is uses in determining the appropriate RTT and buffer size in a network

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] J. Sing and B. Soh, TCP New Vegas: Improving the performance of TCP Vegas over High Latency Links, In the Proceedings of the 4th IEEE International Symposium on Network Computing and Applications (NCA'05), 2005.
[2] M. Loetscher, Simulative Performance Optimization for TCP over UMTS, Diploma Thesis 2002/2003, Swiss Federal Institute of Technology Zurich.
[3] B. S. Yew, B. L. Ong, R. B. Ahmad, Performance Evaluation of TCP Vegas versus Different TCP Variants in Homogeneous and Heterogeneous Networks by Using Network Simulator 2, In the Proceeding of International Journal of Electrical & Computer Sciences IJECS-IJENS, Vol: 11, No. 03, Pp.95-103, June 2011.
[4] G. Xylomenos, G. C. Polyzos, TCP Performance Issues over Wireless Links, In the Proceedings of IEEE Communications Magazine, Volume 39, Number 4, pp. 52–58,2001,
[5] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, Improving TCP/IP Performance over Wireless Networks, In the Proceeding of the 1st ACM International Conference on Mobile Computing and Networking (Mobicom), November 1995.
[6] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, In the Proceeding of the Journal of IEEE/ACM Transactions on Networking (TON), Volume 5, issue , December 1997.
[7] L. Kleinrock, Queuing Theory, Wiley, New York 1975.
[8] A. Aggarwal, S. Savage and T. Anderson, Understanding the Performance of Packet Pacing, In the Proceedings of the ACM SIGCOMM'91 Conference on Communications Architecture and Protocols, pages 133-147, September 1991.
[9] D.G. Le, D. H. Guo and B. X. Wu, TCP Performance Improvement through Inter-layer Enhancement with Mobile IPv6.
[10] S.Lee, B. Kim, Y. Choi: TCP Vegas Slow Start Performance in Large Bandwidth Delay Networks, Seoul National University Korea, ICOIN, Sep 2002
[11] B. L. Ong and S. Hassan, " Effects of Different Packet Sizes in Mobile IPv6 Real-time Communications", In the Proceedings of the Electrical/ Electronic Engineering, Computer, Telecommunication and Information Technology Conference 2005 (ECTI-CON 2005), Pattaya, Thailand. pp 315-319. May 2005.
[12] J. Postel, Transmission Control Protocol Darpa Internet Program Protocol Specification, *Request of Comment 793, Internet Engineering Task Force*, September 1981.
[13] 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,October 1999.
[14] L. L. Peterson, S. H. Low and L. Wang, "Understanding Vegas: A duality model", Journal of the ACM Transactions on Networking, vol.49, no.2, pp 207-235, March 2002.
[15] C. Y. Ho, Y. C. Chan, Y. C. Chen, "An Enhanced Slow-Start Mechanism for TCP Vegas", In the Proceedings of the 11th IEEE International Conference on Parallel and Distributed Systems (ICPADS'05), vol. 1, pp. 405-411, July 2005.
[16] M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", Request for Comment 2581, Internet Engineering Task Force, April 1999.
[17] Van Jacobson. Congestion Avoidance and Control. In *Proceedings of the ACM SIGCOMM '88 Conference on Communications Architectures and Protocols*, pages 314–329, August 1988.
[18] B. L. Ong and S. Hassan, " Effects of Different Packet Sizes in Mobile IPv6 Real-time Communications", In the Proceedings of the Electrical/ Electronic Engineering, Computer, Telecommunication and Information Technology Conference 2005 (ECTI-CON 2005), Pattaya, Thailand. pp 315-319. May 2005.
[19] M. Laubach and J. Halpern, *Classical IP and ARP over ATM*. RFC 2225, Network Research Group, Internet Society, IETF, April 1998. http://www.ietf.org/rfc/rfc2225.txt?number=2225.
[20] The WNDW Production Team (Rob Flickenger, Corinna "Elektra" Aichele, Carlo Fonda, Jim Forster, Ian Howard, Tomas Krag and Marco Zennaro) Wireless Networking in the Developing World, 1st Edition, 2006.