

End-to-End Performance Evaluation of Selected TCP Variants across a Hybrid Wireless Network

*A.O. Oluwatope, A. B. Obabire, and G. A. Aderounm
Comnet Laboratory, Obafemi Awolowo University. Ile-Ife, Nigeria*

aoluwato@oauife.edu.ng gaderoun@oauife.edu.ng

*M. O. Adigun
University of Zululand. South-Africa*

madigun@pan.uzulu.ac.za

Abstract

Reliable transport protocols such as TCP are tuned to perform well in traditional networks where packet losses occur mostly because of congestion. TCP is intended for use as a highly reliable end-to-end transport protocol for transporting applications such as World-Wide Web (WWW) between hosts in packet-switched computer communication networks. TCP was originally designed for wired links where the error rate is really low and actually assumed that packet losses are due to congestion in the network. However, the increasing popularity of wireless networks indicates that wireless links will play more important role in future internetworks but TCP performance in such networks suffers from significant throughput degradation and very high interactive delays. TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in wireless and lossy systems. Thus, in a bid to show and determine the possibility of adapting TCP protocol for optimal performance on the wireless link, this paper reviews and models the behaviors of TCP variants with a view to evaluate the end-to-end performance analysis of TCP versions: TCP Reno, TCP SACK and TCP Westwood (TCPW), which are designed to improve the performance of TCP in lossy networks. A wireless network model was developed using NS-2 network simulator which and the model was simulated. The results were analyzed in MATLAB 6.5 using throughput as a metric for comparison. The overall results indicate that TCP Westwood (TCPW) demonstrates better performance indices over other versions in a hybrid wireless network environment.

Keywords: tcp, tcpwestwood, tcpreno, tcpsack, bandwidth, ns-2

Introduction

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

With rapid growth in the number of nomadic users that access the Internet using wireless technology it has been established that wireless links will play more important role in future internetworks. Many popular Internet applications including World-Wide Web (WWW), File Transfer Protocol (FTP) and email require reliable data delivery over the network. The Transmission Control Protocol (TCP) is the most

widely used transport protocol for this purpose and has become the predominant transport protocol in the Internet today. TCP is a large expanded transport layer protocol that provides connection oriented, reliable service to the application layer. It is intended for use as a highly reliable end-to-end protocol between hosts in packet-switched computer communication networks and was originally designed for wired links where the error rate is really low and actually assumes that packet losses are due to congestion in the network. Quality of Service On the fixed wired networks, less than 1% of the packet losses is considered being due to link errors (Lefevre & Vivier, 2003), a missing packet means, in most cases, that the network is congested and in reaction TCP triggers the congestion algorithm aimed at reducing throughput. In a wireless network, however, packet losses occur more frequently due to link error because of the unreliability of the physical link. This implies the unnecessary invocation of the congestion algorithm which reduces the throughput drastically and causes severe performance degradation. The congestion avoidance mechanism is mainly the origin of the poor QoS of TCP over wireless.

However, with the increase in wireless data applications, there is a growing need to adapt TCP to wireless networks. Thus, it is necessary to extend TCP features to be efficient on a wireless interface while taking into account the characteristics of a wireless channel.

The Existing Possible Solutions

There are two different approaches to improving TCP performance in wireless lossy networks. The first approach hides any non-congestion related losses from the TCP sender which requires no change to existing sender implementations. Protocols that adopt this approach attempt to make the lossy link appear as a higher quality link with a reduced effective bandwidth. As a result, most of the losses seen by the TCP sender are caused by congestion.

The second approach attempts to make the sender aware of the existence of wireless hops, and realize that some packet losses are not due to congestion. The sender can then avoid invoking congestion control algorithms when non-congestion related losses occur. We classify the many schemes into three basic groups based on their fundamental philosophy: end-to-end proposals, split-connection proposals, and link-layer proposals. (Lefevre & Vivier, 2003).

End-to-end Proposal

The end-to-end protocols attempt to make the TCP sender handle losses through the use of two techniques. First, the use of selective acknowledgment (SACKs) that allows the sender recovers from multiple packet losses in a window without resorting to a coarse timeout. Second, the attempt to have the sender distinguishes between congestion and other forms of losses using an Explicit Loss Notification (ELN) mechanism.

End-to-end protocols

TCP Sack (Selective Acknowledgments): Since standard TCP uses a cumulative acknowledgment scheme, it often does not provide the sender with sufficient information to recover quickly from multiple packet losses within a single transmission window. Study (e.g., Fall & Floyd, 1996; Hoe, 1996) has shown that TCP enhanced with selective acknowledgments performs better than standard TCP in such situations. SACK's were added as an option to TCP by RFC 1072 (Jacobson & Braden, 1988). The SACK RFC proposes that each acknowledgment contain information about up to three noncontiguous blocks of data that have been received successfully by the receiver. Each block of data is described by its starting and ending sequence number. Due to the limited number of blocks, it is best to inform the sender about the most recent blocks received. The RFC does not specify the sender behavior, except to require that standard TCP congestion control actions be performed when losses occur.

TCP Reno: The Reno TCP assumes a packet has been lost (due to congestion) when it observes 3 duplicate ACKs (DUPACKS), or when a retransmission timer expires. In either case, it reacts by setting *ssthresh* to half of the current window size (the minimum of *cwnd* and *window*). It then initializes *cwnd* back to the value of 1. This will typically cause the TCP to enter slow-start (Fall & Floyd, 1996; Naito, Okada, Saito, Yamazaato, & Katayama, 2002; Padhye, Firoiu, & Towsley, 1999; Padhye, Firoiu, Towsley, & Kurose, 2000).

TCP Westwood: The key innovative idea in TCP Westwood is that the TCP sender continuously computes the connection BandWidth Estimate (BWE) which is defined as the share of bottleneck bandwidth available to the connection i.e. the packet rate of the connection by properly averaging the rate of returning ACKs. BWE is equal to the rate at which data is delivered to the connection destination. The delivery rate estimate is based on information in the ACKs, and the rate at which the ACKs are received (Claudio et al., 2002). After a packet loss indication, which is mostly due to congestion but could also be due to link errors, through packets loss symptoms which include the reception of 3 duplicates acknowledgment (DUPACKS) or coarse timeout expiration, the sender uses the estimated bandwidth to properly set the congestion window and the slow start threshold. In contrast to TCP Reno, which “blindly” halves the congestion window after three duplicate ACKs, TCP Westwood (TCPW) attempts to select a slow start threshold and a congestion window which is consistent with the effective connection rate at the time congestion is experienced.

Bandwidth Estimation Mechanism

When an ACK is received at the source at time t_k such that d_k bytes have been received at the TCP receiver, the sample bandwidth used by this connection can be measured as (Claudio et al, 2002):

$$b_k = d_k / (t_k - t_{k-1}) \dots\dots\dots (1)$$

where t_{k-1} is the time the previous ACK was received.

Letting $\Delta t_k = t_k - t_{k-1}$,

$$\text{Then } b_k = d_k / \Delta t_k \dots\dots\dots (2)$$

Since congestion occurs whenever the low-frequency input traffic rate exceed the link capacity, a low-pass filter is employed to average sampled measurements and to obtain the low-frequency components of the available bandwidth.

The low-pass filter is proposed as follows: b_k is the bandwidth sample, and \hat{b}_k the filter estimate of the bandwidth at time t_k . Also, α_k is the time varying exponential filter coefficient at t_k . The TCPW filter is then given by:

$$\hat{b}_k = \alpha_k \hat{b}_{k-1} + (1 - \alpha_k) \left(\frac{b_k + b_{k-1}}{2} \right) \dots\dots (3)$$

where

$$\alpha_k = \frac{2\tau - \Delta t_k}{2\tau + \Delta t_k} \dots\dots\dots (4)$$

and $1/\tau$ is the filter cut-off frequency.

The coefficients α_k are made to depend on Δt_k to counteract the effect of non-deterministic inter-arrival times. When the inter-arrival time Δt_k increases, the last value \hat{b}_{k-1} should have less significance. On the other hand, a recent \hat{b}_{k-1} is given higher significance. The coefficient α_k decreases when the inter-arrival time increases, and thus the previous value b_{k-1} is accorded less significance with respect to the last two recent samples which are weighted by:

$$(1 - \alpha_k) / 2 \dots\dots\dots (5)$$

Since the TCPW filter has a cut-off frequency equal to $1/\tau$, all frequency components above $1/\tau$ are filtered out.

Work Done

Architecture of the Hybrid Wireless Network

We considered a sample campus area network setup in Obafemi Awolowo University (OAU) Ile-Ife Nigeria that connects to the UNIVERSITY of ZULULAND campus network via the Internet. At the OAU campus network end, several local area networks are linked to the central control via wireless links. To simplify the architecture, we built and simulated only one wireless link connecting two sub-nets. This wireless link is a full-duplex data communication link. The sub-nets traffic are transported over wire-line LAN. OAU campus wireless network is the largest university network in Nigeria. It is known as OAUNet. It records over 2,000 users log-on per day from 13 sub-networks connected by wireless (packet radio frequency) links. Figure 1 depicts our simple model.

A sub-net gateway runs a dynamic host configuration protocol (DHCP) service that allocates internet protocol (IP) address numbers to the workstations-clients. Sub-net gateways also acts as routers between a particular local area network (LAN) and the central university network (CUN). The mode of connectivity between a sub-net gateway and the central university network is a wireless radio link whose bandwidth is 2Mbps. At the CUN is an omni-directional antenna concentrating 13 wireless links from different directions of the campus.

Limitations

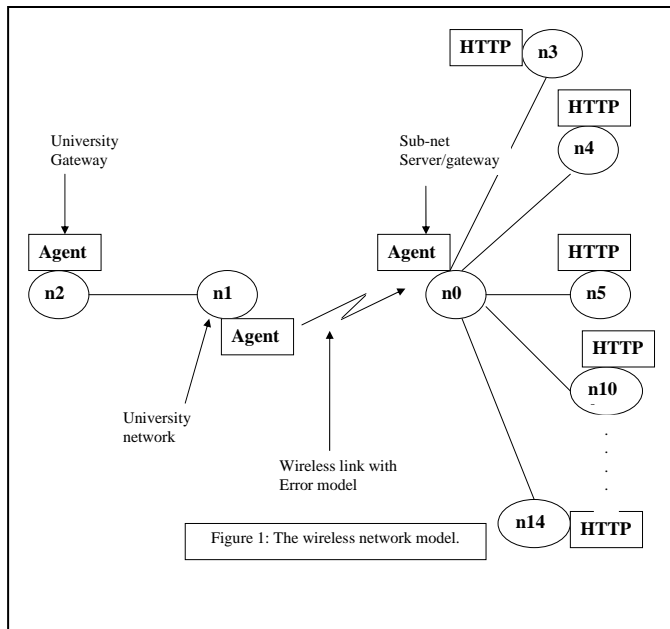
The modifications on TCP in this work will be preferably processed without implying an entire alteration to the widespread TCP protocol. The simulation study was carried out on the OAU wireless network using only one sub-net gateway out of a total of 13 sub-nets. Also, this work is limited to the accuracy of the model developed using the NS simulator (NS2) (Altman & Jimenez, 2002; Kevin & Kannan, 2003). The application assumed to be running on the OAU network model is HTTP, although, in future, VOIP is a likely application being considered.

Development Tools

The development tool used in this work includes:

- **NS simulator (NS2):** NS2 is a discrete event simulator that operates in virtual time and supports various flavors of TCP and other transport protocols NS uses NAM, an anima-

tion tool, developed in Tcl/Tk, to visualize the simulation packet traces which are created when running NS scripts (ns-2) (Allman & Paxson, 2000).



- **MATLAB 6.5:** The data collected, based on performance measures, were saved in an output trace file and afterwards plotted and analyzed using MATLAB 6.5

Simulation scenario

NS-2 network simulator ran on a Linux operating system installed on a Pentium III 1.33GHz, 256MB RAM system. The simulation was run several times within the range of 15ms - 20ms. Our simulation environment parameters setting of 5% wireless packet error rate, 2Mbps bottleneck bandwidth and two-way propagation time of 5ms were set as standard. The simulated wireless network model is as shown in Figure 1.

Results of the Simulation

Simulation of the Model at Constant Packet Error Rate 5%

The result obtained is shown in Figure 3. At 5% PER, there was a transient behavior in which all the TCP protocol versions were in the slow start phase (first 10s). Here, TCPs window size initially increased exponentially with TCPWestwood exhibits a very high throughput, in excess of .45Mbps as against TCPReno and TCPSACK that recorded throughput of about .35 and .38Mbps respectively. Also during the slow start phase, at time 3.5s, there were losses traceable to time-out making TCPReno and TCPSACK to suddenly half their window size, therefore reducing

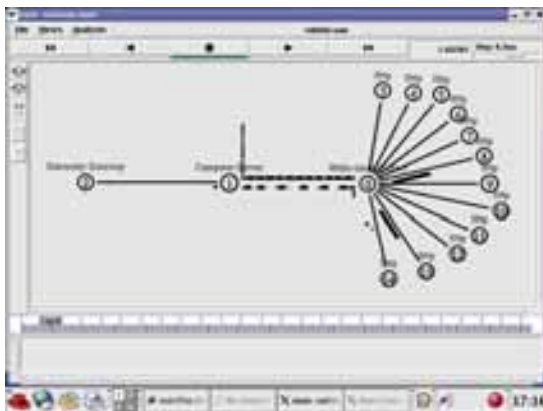


Figure 2: OAU Wireless Network Model showing the link between computer

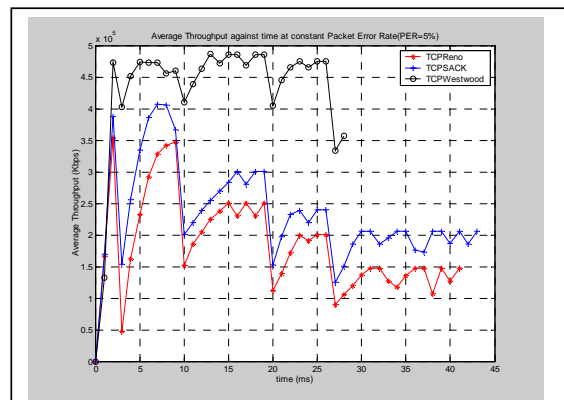
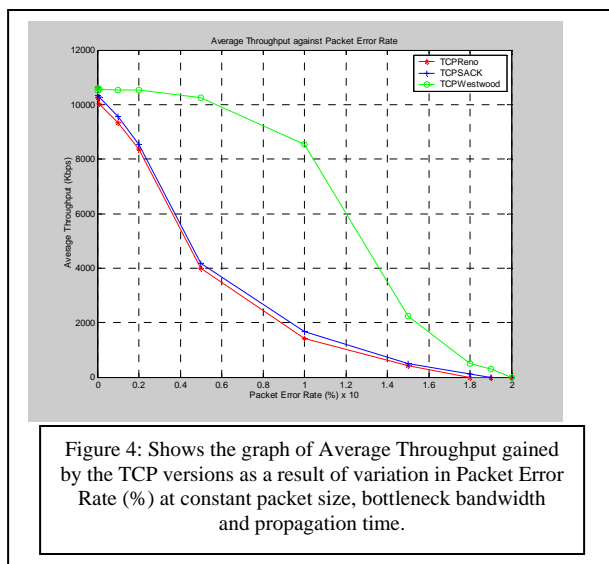


Figure 3: Shows the graph of Average Throughput gained by the TCP versions against time at constant

throughput to about 0.4 and 1.5 respectively. While TCPWestwood reduced its window size by setting it to the measured rate currently experienced by the connection using bandwidth estimation mechanism, contrary to the conventional multiplicative decrease scheme (i.e., divide the current window by half) which caused the throughput to only reduced by 0.05Mbps. After time 10s all TCP versions switched into congestion avoidance phase and their window size increases almost linearly until losses occurred again at time 20 and all TCP versions entered into slow start phase again. The other control mechanisms (i.e., slow start after timeout, exponential/additive increase in slow start/congestion avoidance, etc.) are identical to all the TCP versions. The transmission continued in this order with TCPWestwood exhibiting the highest and most regular throughput of approx. 0.5Mbps all through the transmission phases.

Simulation of the Model at Random Packet Error Rate



A bottleneck bandwidth was set to 2Mbps, and a two-way propagation time set at 5ms. At 0% PER, the performances of TCP protocols were virtually identical at a throughput of about 10000 Kbps (1Mbps) approximately 50% link utilisation. At 85 PER, TCPWestwood remained within the neighbourhood of 1Mbps while TCP Reno and TCPSACK had dropped below 0.4Mbps. At 10% error rates, appropriate for wireless links, the throughput TCPWestwood just dropped slightly below 0.9Mbps as against 0.2Mbps recorded for TCP Reno and TCPSACK. This shows an improvement is about 400%. However, as the error rate increases beyond 10%, the improvement obtained via TCPWestwood

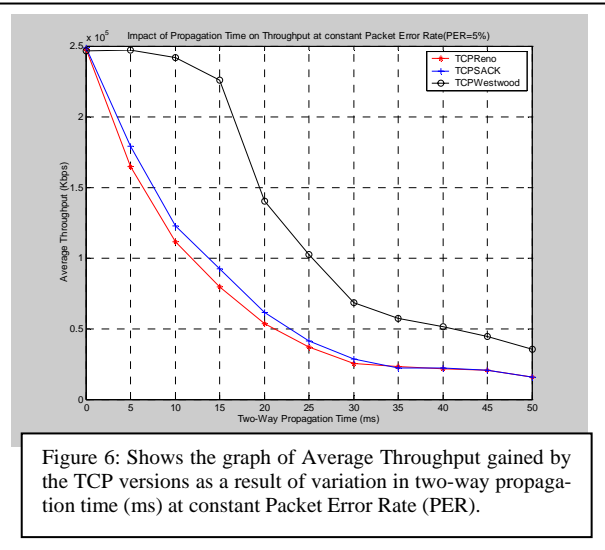
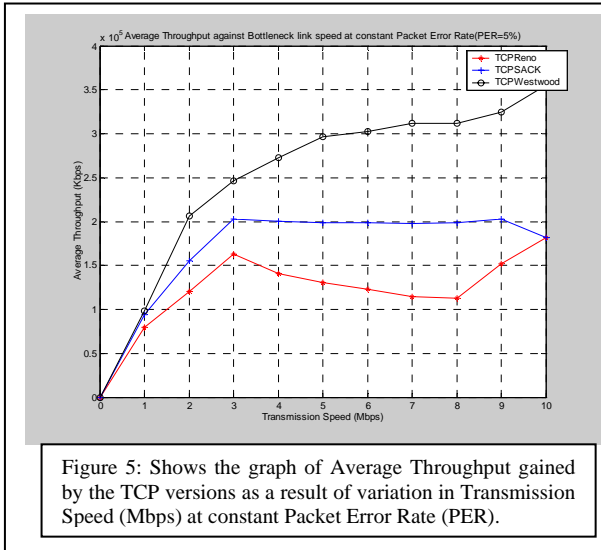
decreases. At very high error rates of about 20% and above, TCPWestwood as well as TCP Reno and TCPSACK suffer low bandwidth utilization which results to zero throughput due to outrageous packet drop.

Simulation of the Model at Random Transmission Speed (Mbps) (Bottleneck Link Speed)

Here, the channel errors rate are independent and correspond to a 5% packet loss rate, equivalent to a BER in the order of 10^{-50} and round trip propagation time is set at 5ms. At transmission speed of 2Mbps, TCPWestwood was operating at over 0.2Mbps while TCP Reno and TCPSACK were operating at 0.15 and 0.12Mbps respectively. The simulation result as shown in Figure 5 indicates significant gain in throughput in TCPWestwood relative to the other protocols as the bottleneck link transmission speed increases. Thus, TCPWestwood provides better utilization at higher link speeds. At 10 Mbps, TCPWestwood achieved a throughput equal to about 300% of TCP Reno and TCPSACK. This graph also establishes that retransmission rate is a function of link speed. This shows that TCPW, while less conservative in reducing congestion window (*cwin*) and slow start threshold (*ssthresh*), did not have a significantly higher retransmission rate than TCP Reno.

Simulation of the Model at Random Propagation Time (ms)

A random propagation time with the channel error fixed at 0.05% packet error rate and bottleneck link transmission speed at 2Mbps was used in order to compare impact of the propagation time on the performance of TCP versions. The simulation results are presented in Figure 6. TCPWestwood throughput remained uniform from 0ms to 5ms propagation time and had dropped by 0.1% at 15ms. In the case of TCPReno and TCPSACK average throughput had by 60% of the starting throughput. There was a steep reduction immediately propagation time increased to 20 and 25ms. The reason may not be unconnected delay in bandwidth estimation processes of TCPwestwood at such propagation times.



Conclusion

When packets are lost in the wireless network for reason other than congestion, it results to unnecessary reduction in end-to-end throughput and hence, in suboptimal performance. This project studies the behavior and performance of TCP when applied over wireless network by surveying the TCP versions based on End-to-End scheme which have been designed for best-effort traffic in wireless network. Various result obtained from the simulation analysis using performance parameters when TCP versions were applied over the wireless network shows the expected performance level.

Thus, TCPWestwood showed superior performance and improved throughput over the wireless network. In particular, the throughput performance of TCPWestwood over a lossy wireless network exhibited a 100 % improvement over TCPReno and TCPSACK implementations. In future we hope to scale upward the applications running the network with particular interest in real-time applications (VOIP and Streaming multimedia).

References

Allman, M. & Paxson, V. (2000). TCP congestion control. *IEEE Communications Surveys & Tutorials*, 4(5). Retrieved from <http://www.comsoc.org/pubs/surveys>

Altman, E. & Jimenez, T. (2002). NS Simulator for beginners. Retrieved from <http://www-mash.cs.berkeley.edu/ns>

Balakrishnan, H. (1998). Challenges to reliable data transport over heterogeneous wireless networks. PhD thesis, Computer Science Division, University of California at Berkeley, Berkeley, CA. pp 25-30.

End-to-End Performance Evaluation of Selected TCP Variants

- Claudio, C., Mario, G., Scott, L., Saverio, M., Sanadidi, M. Y. & Ren, W. (2002). TCP Wwood: end-to-end bandwidth estimation for enhanced transport over wireless links. *ACM/Baltzer Wireless Networks Journal*, 8, 467-479.
- Fall, K. & Floyd, S. (1996). Simulation-based comparisons of Tahoe, Reno, and Sack TCP. *Computer Communications Review*, 26(3), 5-21.
- Hoe, J. C. (1996). Improving the start-up behavior of a congestion control scheme for TCP. Proceedings of the *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 26(4), 270-280. ISBN 0-89791-790-1.
- Jacobson, V. & Braden, R. T. (1988), TCP extensions for long delay paths. RFC, RFC-1072.
- Kevin, F. & Kannan, V. (2003). The ns manual. Retrieved from <http://www.isi.edu/nsnam/ns/>
- Lefevre, F. & Vivier, G. (2003). Understanding TCP's behavior over wireless links. Motorola Labs-Centre de Recherche de Motorola-Paris, pp 1-3.
- Naito, K., Okada, H., Saito, M., Yamazaato, T., & Katayama, M., (2002). New analytical model for TCP in wireless environment. *IEICE Trans. Journal*, E85-A(7).
- ns-2 network simulator, version 9, LBL. Retrieved from <http://www-mash.cs.berkeley.edu/ns>
- Padhye, J., Firoiu, V., & Towsley, D. (1999). A stochastic model TCP Reno congestion avoidance and control. CMPSCI Technical Report 99-02. Dept. of Computer Science, University of Massachusetts Amherst, MA.
- Padhye, J., Firoiu, V., Towsley, D. F. & Kurose, J. F. (2000). Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE ACM Transactions on Networking*, 8(2), 133-145.
- TCP Westwood modules for ns-2. Retrieved from <http://www.telematics.polito.it/casetti/tcp-westwood>

Biographies



A. O. Oluwatope obtained his Bachelors and Masters degree in Computer Engineering and Computer Science from Obafemi Awolowo University, Ile-Ife, in 1995 and 2003, respectively. He is a registered computer engineer with Council for the Regulation of Engineering Practice in Nigeria (COREN), a member of Nigerian Society of Engineers (NSE) and an Associate member of Nigeria Computer Society (NSC). He has co-authored several articles published in Nigeria and abroad. His current research interests include internet congestion control, internet protocol development, algorithm modeling and simulation and grid computing. He has about seven years of teaching and research experience. He is currently a doctoral student and a lecturer in the Department of Computer Science and Engineering of the same university.

A. B. Obabire obtained his Bachelor of Science degree in Computer Engineering from Obafemi Awolowo University, Ile-Ife, in 2004. He is a student member of Nigerian Society of Engineers (NSE). His research interest is in internet protocol simulation. Mr. Obabire has since completed a one year National Youth Service.



G.A. Aderounmu holds a research degree M.Sc./PhD in Computer Science from Obafemi Awolowo University, Ile-Ife, Nigeria (1997 and 2001, respectively). He is a member of the Nigerian Society of Engineers (NSE) and is also a registered computer engineer with Council for the Regulation of Engineering Practice in Nigeria (COREN). He is also a Member of Nigerian Computer Society (NCS) and Computer Professional Registration Council of Nigeria (CPN). He has over 13 years of experience in teaching and research. He is an author of many journal articles in Nigeria and abroad. His special interests include engineering education in Nigeria, curriculum development, and computer communication and network. He is a Visiting Research Fellow to the University of Zululand, Republic of South Africa. He is currently a Senior Lecturer, Deputy Director of the Information Technology and Communication Unit (INTECU) and the Acting Head of the Department of Computer Science and Engineering of the same university.



M. O. Adigun holds a research degree PhD in Computer Science from Obafemi Awolowo University, Ile-Ife, Nigeria which he obtained in 1989. Currently, he is a Professor and Head, Department of Computer Science, University of Zululand, Republic of South Africa. He is an author of many journal articles in Nigeria, Republic of South Africa and abroad. His research interests include Software Engineering, Mobile Computing, Modeling and Simulation, and Performance Analysis of Computer System.