Rural Wireless Mesh Network Analysis On Mobile Devices

Ghislaine L. Ngangom Tiemeni, Isabella M. Venter and William D. Tucker
Department of Computer Science
University of the Western Cape, Private Bag X17, Bellville 7535 South Africa
Tel: +27 21 9593010, Fax: +27 21 9591274
Email: {**3261404**, iventer, btucker}@uwc.ac.za

Abstract-Traffic-generator software is a valuable tool for generating synthetic yet realistic workloads that can be analyzed to test communication networks' quality of service. This paper describes the research and design of a packet-level traffic generator (known as MTGawn) on a mobile platform. The main objective is to describe a way to adapt a well-known packet generator—designed for a personal computer (PC)—for use with mobile devices. This will simplify the feasibility testing and monitoring of wireless mesh networks deployed in remote areas where mobile devices are more practical and affordable than PCs or laptops, i.e. ease of battery charging and of usability of a touchscreen given the resource constraints of a mobile device. In order to achieve this objective, a suitable model for emulating realistic workloads randomized in terms of packet size and time between packets was designed using various statistical distributions such as Constant, Uniform, Pareto and Normal. The most common transport protocols, transmission control protocol (TCP) and user datagram protocol (UDP), were used to enable the generation of accurate and representative traffic patterns that were characteristic of user behaviour. The paper covers work done in the laboratory with a mesh network testbed. We employ design science research in a cyclical fashion to move toward demonstrating that a mobile generator can provide acceptable packet generation and analysis functionality on a mobile platform in order to move from the laboratory to rural in-field use.

Index Terms—network performance, end user mobile application, quality of service, traffic emulation, wireless mesh network

I. INTRODUCTION

A wireless mesh network (WMN) is a communication network system in which all nodes communicate together without any centralized infrastructure to control the network. This lack of dependency on any pre-installed infrastructure makes WMNs suitable for addressing connectivity issues in disaster and field scenarios, allowing, for instance, the network to handle geographic challenges in dispersed communities that typically experience large amounts of attenuation or noise due to the proximity of mountains (Gunashekar, Das, Erlebach, & Warrington, 2014; Carrano, Bletsas, & Magalhães, 2007). In addition, WMNs are known to be self-healing and extremely reliable because of their ability to maintain connectivity even if a node fails and the capacity of all nodes to obtain access to a wireless point if one node has access to it (Gunashekar et al., 2014; Hamidian, Palazzi, Chong, Navarro, Korner, & Gerla, 2009).

In comparison to traditional, cable and digital subscriber line broadband networks, which require a substantial investment in order to begin offering services, WMNs are very flexible, hence they can be built progressively and can grow as the number of users grows. Further benefits of WMNs include ease of installation, the absence of cable cost, automatic connection among nodes, the discovery of newly added nodes and redundancy (Carrano et al., 2007; Hamidian et al., 2009; Sichitiu, 2005). However, like many other evolving technologies, WMNs come with advantages and drawbacks. As user data travels through multiple hops, the complexity of the routing protocol impacts on performance and makes it challenging to provide high-level security to end user. Factors that directly affect the performance of WMNs include load balancing, avoiding congested routes and dealing with interference patterns (Carrano et al., 2007).

Nevertheless, the decentralized topology of WMNs, added to their flexibility, low cost and ease of deployment, has made them a useful of providing broadband connectivity to people living in rural areas (Gunashekar et al., 2014; Carrano et al., 2007).

In recent years the deployment of multimedia applications such as video conferencing and voice-over Internet protocol (VoIP)—in addition to traditional data services—has significantly increased in WMNs. As demand for multimedia services evolves, it becomes fairly difficult to maintain quality of service (QoS) in multihop WMNs where dynamic environments cause fragile links and high packet-loss ratios. These factors degrade the QoS of multimedia services and significantly affect user satisfaction (Cheng, Mohapatra, Lee, & Banerjee, 2008). For this reason network managers have to keep track of network evolution in order to identify and resolve possible problems that may occur so as to efficiently deliver network services to the end user. To deal with this issue, many networking experts rely on performance evaluation tools such as traffic generators and packet sniffers.

Literature on some of the most interesting tools used to estimate the performance of networks was reviewed in the initial stages of this study. Despite the powerful features of these tools, the literature survey indicated that they are unable to simplify the feasibility testing and monitoring of WMNs deployed in rural areas. For this reason we propose a new tool that we call MTGawn, i.e. 'mobile traffic generator for analysis of wireless networks'. A prototype of this tool has been designed and deployed on a mobile device, and is being tested using an experimental mesh network as testbed.

The paper is organized as follows. Section II covers a relevant sample of work related to network performance analysis. Section III describes the methods used to follow a design science research (DSR) approach to move toward producing a traffic emulator with analysis capabilities on a mobile form factor. Section IV describes the architecture of a prototype deployed in a laboratory wireless mesh testbed. Section V presents the results of preliminary tests, and finally, Section VI draws conclusions and identifies future work.

II. RELATED WORK

The monitoring and evaluation of WMNs is essential to ensure that the QoS—required for widely used protocols, such as VoIP—is satisfied. To do so, networking experts rely on traffic-generator software to generate synthetic but realistic traffic patterns that can assist in predicting and estimating the performance of networks. These are discussed briefly below.

According to Nicola, Giordano, Procissi, and Secchi (2005), it is essential to evaluate the performance of high-speed networks either because of the lack of reliable tools to generate traffic workloads at high rates or because of the inaccessibility of network equipment. For these reasons they implemented a tool called Brute (browny and robust traffic engine), a Linux application allowing high-speed packet generation on personal computers (PC).

Similar to Brute, Harpoon (Sommers & Barford, 2004; Sommers, Kim, & Barford, 2004) is a flow-level traffic generator for router and network tests that focuses on the generation of transmission control protocol (TCP) and user datagram protocol (UDP) packet flows. These packet flows have the same characteristics as routers for the purpose of showing the empirical behaviour of routers under actual conditions. Another significant feature of Harpoon is its ability to self-configure by automatically extracting parameters from standard net flow logs or packet traces.

Avallone, Pescape, and Ventre (2003) developed D-ITG, a tool for the generation of transport-layer traffic (TCP and UDP) and other types of traffic, including VoIP, Telnet, and domain name service. D-ITG has numerous features such as allowing the measurement of round-trip time and one-way delay, while it has the capacity to keep information about received and transmitted packets. This feature allows the evaluation of important network QoS metrics such as throughput, jitter, packet loss and average bit rate. D-ITG has additional functionality such as using different network loads or different network configurations to study scalability problems. It allows the generation of complex and varied traffic sources, and offers the option to repeat exactly the same traffic pattern (Avallone et al., 2003).

III. DESIGN AND METHODOLOGY

DSR methodology was used to build a mobile traffic generator. Each cycle of the DSR iterative process consists of six phases—identify, build, document, select, evaluate and communicate (Brocke & Buddendick, 2006) (see Figure 1).

A. Identify

During this phase methods such as a literature survey and document analysis were used to identify and clarify our main concern, which is the lack in the literature of a tool capable of running efficiently on devices with limited screen interface and computing power such as mobile devices. Our aim is to make available a cost-effective tool that generates typical traffic and that will run on affordable and easily accessible devices.

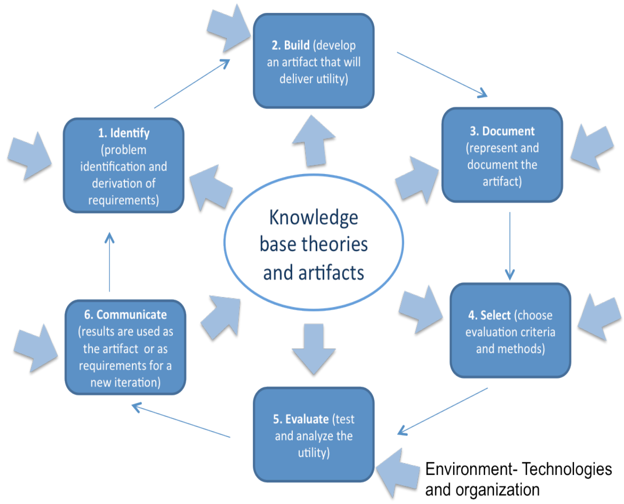

Figure 1: The basic form of DSR cycles

Source: adapted from Brocke and Buddendick (2006)

B. Build and document

These phases guided us through the development and representation of a model capable of representing the relevant features of real-life traffic flows. To create a successful model it is essential to classify a particular network's activities, because different user activities produce different traffic patterns and each traffic pattern can be characterized by various parameters. Besides, different traffic can flow in the network at the same time, because users often browse the web, read emails, send text messages, play games, make voice calls and stream videos simultaneously. In this case data traffic is a result of parallel user activities (Varga & Olaszi, 2013).

The conceptual model for this research categorizes traffic in five different types of flows—custom (UDP and TCP flows using various statistical distributions), voice, data, game, and text, and incorporates a mixture of these flows in order to represent real-life traffic. In a communication network traffic flows circulating between a sender and a receiver are characterized by two significant parameters: the size of each transmitted packet and the elapsed time between packet transmissions (as shown in Figure 2). In this research a stochastic model was developed to represent both packet size and time between two consecutive packets in the network. Other significant aspects to consider while defining a model are the facts that packets circulating in the network have different sizes, while the time between two consecutive packets is not always constant. Our model was implemented in order to emulate these two parameters using various statistical distributions (Constant, Uniform, Pareto, Normal, etc.) to randomize them. The distributions chosen to emulate each type of traffic should be able to capture the relevant and representative characteristic of the emulated traffic.

For the sender, each time period represents the elapsed time between the transmission of the current packet and the transmission of the next packet, while for the receiver each time period represents the elapsed time between the reception of the previous packet and the reception of the current packet.

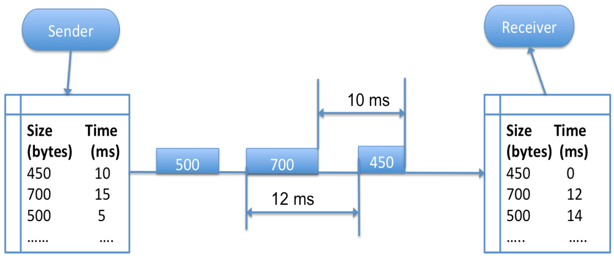

Figure 2: Example of traffic flow circulating in the network between a sender and receiver

C. Select and evaluate

After designing and implementing the traffic-generator tool based on the conceptual model defined earlier, the next step is to select the appropriate performance metrics (packet loss, delay and jitter) to represent the service under evaluation. Then the system is tested. The sender and receiver generate a trace file containing detailed information about each packet sent and received during the generation process. Both trace files are then analyzed to determine the functioning of the network or protocol under evaluation.

D. Communicate

This step includes a qualitative analysis in order to verify the efficacy of the tool. In the present research several experiments will be completed both in a laboratory testbed environment and in an actual WMN deployed in a rural area. The results obtained using the MTGawn tool will be compared to results obtained using well-recognized computer-based traffic generators such as D-ITG in the same experiments. In this step, if the results are not satisfactory the cycle is repeated and more experiments are done until the usefulness of the tool has been proved.

IV. ARCHITECTURE OF THE SYSTEM

The proposed MTGawn system was divided into four main elements: modelling, sender, receiver and analyzer. Each component plays a specific role, but all the components communicate simultaneously to provide the underlying features. The system structure follows a linear form, i.e. the flow is first created, then sent over the network and received at the other endpoint of the network, and then analyzed (see Figure 3).

A. Modelling

The 'modelling' module is in charge of emulating real-life traffic, i.e. it is responsible for defining appropriate stochastic models to generate realistic traffic in terms of packet size and the time between packets.

B. Sender

The 'sender' module is the core function responsible for sending the flows over the network. To do so, it uses the type of transport protocol appropriate for the specific type of flow (UDP or TCP). For example, in case of voice traffic, either RTP (real-time protocol) or cRTP (compressed real-time protocol) packets are created and encapsulated in the UDP packet to carry the voice packet. Both the sender and the receiver module integrate another component ('request manager') whose role is to manage parallel incoming and outgoing traffic involving single or multiple senders or receivers. The request manager's purpose is to allow multiple flows to be sent and received simultaneously.

C. Receiver

The 'receiver' module's role is to receive the flow, manage it and control the parallel incoming flows. The relevant information about each packet received is saved in a file. Each packet includes information such as the flow identifier, the packet identifier, the time sent, the time received and the payload size of the packet.

D. Analyzer

The 'analyzer' module is responsible for analyzing the sent and received flows in order to compute the QoS of the network under consideration.

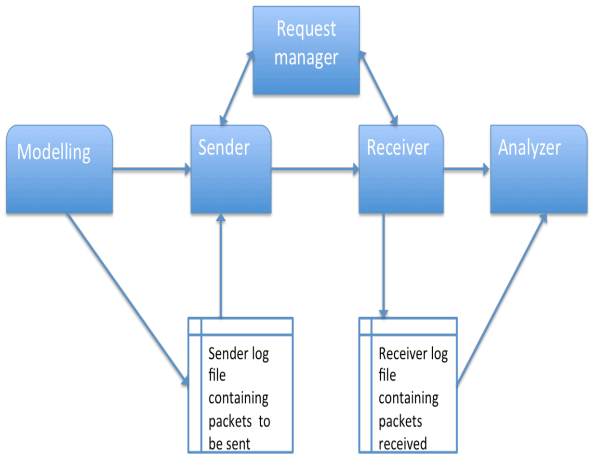

Figure 3: Basic architecture of the MTGawn system

V. RESULTS

The experiments conducted so far in this research focused on estimating the QoS of TCP and UDP in a testbed consisting of a WMN deployed in a laboratory. The testbed to carry out the measurement consisted of a WMN with three mesh potato nodes and two android phones connected to the network wirelessly. Figure **4** depicts the testbed environment.

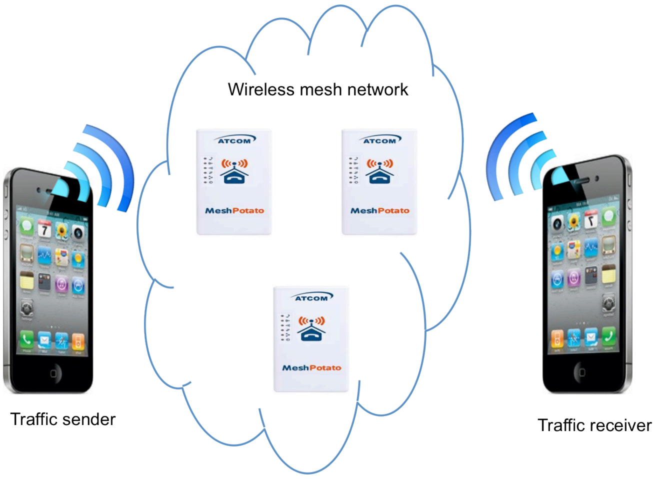

Figure 4: Testbed environment

The purpose of these experiments was to generate packets in the network without impacting on the services offered to end users. In addition, this testbed will serve as a platform for assessing the performance of UDP and TCP traffic flows in terms of various pre-defined scenarios without affecting the QoS of other applications. In order to do this the application was deployed on two android devices connected to the testbed WMN. One of the devices was configured as a traffic sender and the other as a traffic receiver.

Different applications (data/voice/multimedia) have different requirements for network service in order to be useful to users—some applications are impaired by message loss and others are impaired by delay or jitter (Marsic, 2013). For this reason, we focused on three performance metrics: delay, jitter and packet loss.

These performance metrics were computed as follow:

i. **Delay:** If S_i is time transmitted for packet *i* and R_i is the time of arrival for packet *i*, then the delay D_i is determined as the difference between the received and transmitted time.

$$D_i = R_i - S_i$$

ii. **Jitter:** The inter-arrival jitter is defined as the difference (D) in packet spacing at the receiver (R) compared to the sender (S) for a pair of packets. For two packets (*i* and *j*), D is expressed as:

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

$$\textbf{Average Jitter} = \frac{\sum_i^n |D(i-1,i)|}{n}$$

where *n* is the total number of packet sent.

iii. **Packet loss:** The amount of packet loss is the difference between the numbers of transmitted and received packets. If *n* represent the number of packets transmitted and *m* represent the number of packets received, then the percentage of packet loss is defined as:

$$\textbf{Packet loss (\%)} = \left(\frac{n-m}{n}\right) \times 100 = 100 - \frac{m}{n} \times 100$$

For simplicity, time synchronization between the mobile sender and the mobile receiver was not considered during these experiments.

The first DSR cycle focused on the design of an MTGawn prototype for a single TCP and UDP flow using two different distributions—Constant and Uniform—for the representation of both time between packets and packets size. During this phase, a series of experiments was completed:

1. The first experiment was carried out to test the performance of a single UDP flow over the WMN. A total number of 1000 UDP packets were sent at a constant rate of 100 packets per second (a constant time of 10 millisecond (ms) between two consecutive packets) and the size of each packet was equal to 500 bytes.
2. The same experiment was repeated for a single TCP flow. A total number of 1000 TCP packets were sent at a constant rate of 100 packets per second (a constant time of 10 ms between two consecutive packets) and the size of each packet was equal to 500 bytes. Table 1 shows the results obtained in the testbed environment.

TABLE 1: SINGLE TCP AND UDP FLOW WITH CONSTANT DISTRIBUTION FOR BOTH PACKET SIZE AND TIME BETWEEN PACKETS

	TCP	UDP
Minimum delay (ms)	1045	956
Maximum delay (ms)	1310	1002
Average jitter (ms)	1.028	0.928
Packet loss	0%	0%

3. Another experiment was carried out to evaluate the performance of a single UDP traffic flow using the Uniform distribution. A total number of 1000 UDP packets were sent with uniformly distributed time between packets (between 10 and 20 ms) and uniformly distributed packet sizes (between 500 and 1000 bytes).

4. The previous experiment was repeated for a single TCP flow. A total number of 1000 TCP packets were sent with uniformly distributed time between packets (between 10 and 20 ms) and uniformly distributed packet sizes (between 500 and 1000 bytes). Table 2 shows how the testbed WMN performs with the Uniform distribution.

TABLE 2: SINGLE TCP AND UDP FLOW WITH UNIFORM DISTRIBUTION FOR BOTH PACKET SIZE AND TIME BETWEEN PACKETS

	TCP	UDP
Minimum delay (ms)	1151	1008
Maximum delay (ms)	1276	1115
Average jitter (ms)	1.235	1.017
Packet loss	0%	0%

In order to emulate multiple user activities simultaneously, a new DSR cycle was iterated in order to estimate the performance of testbed WMN networks under more realistic conditions. The purpose of this phase was to generate complex and varied TCP and UDP traffic sources representative of a wide range of traffic conditions. Different traffic patterns were produced to simulate different user activities. In this case, four distinctive traffic patterns were generated and sent simultaneously between the mobile sender and mobile receiver; the results are given in Table 3.

TABLE 3: MULTIPLE UDP AND TCP FLOWS SENT SIMULTANEOUSLY

	UDP (a)	UDP (b)	TCP (c)	TCP (d)
Minimum delay (ms)	3170	3050	4178	3620
Maximum delay (ms)	3450	3152	4276	4515
Average jitter (ms)	2.581	2.397	3.032	3.593
Packet loss	39%	26%	0%	0%

- One flow for UDP traffic pattern (a): using a Constant distribution for packet size (500 bytes) at a uniform rate of between 1000 and 2000 packets per second.
- One flow for UDP traffic pattern (b): using the Uniform distribution for packet size (between 500 and 1000 bytes) at a constant rate of 1000 packets per second.
- One flow for TCP traffic pattern (c): using the Constant distribution for packet size (500 bytes) at a uniform rate of between 1000 and 2000 packets per second.
- One flow for TCP traffic pattern (d): using the Uniform distribution for packet size (between 500 and 1000 bytes) at a constant rate of 1000 packets per second.

From the results of these experiments general conclusions can be drawn about how UDP and TCP traffic flows perform under diverse scenarios. According to the above results it can be concluded that packet loss was observed only in cases of UDP traffic, while larger delay and jitter were observed in TCP traffic.

Another observation is the increase in delay and packet loss when many flows are sent simultaneously. This observation is reasonable even in a real-life scenario, where the growth of users using the network leads to the growth of traffic flowing through the network. This growth in traffic load generally results in non-negligible service delay and packet loss.

VI. CONCLUSION AND FUTURE WORK

This paper presents preliminary results obtained during the development of a prototype for a mobile traffic generator called MTGawn. The tool is intended to ease feasibility testing and performance evaluation of a rural WMN by implementing a 'stripped down' version of a packet generation and monitoring system to a mobile platform with functionality found in common open source tools. The primary phase of the design process of this prototype involved modelling traffic patterns illustrative of realistic TCP and UDP traffic flows. Each traffic pattern was characterized by packet size and inter-arrival distributions. For this purpose, it is important to appropriately depict both parameters. These parameters were chosen such that they helped to build an effective traffic model for a given traffic pattern (Varga & Olaszi, 2013). From this perspective, a model was defined to represent traffic patterns in different scenarios by using various distributions. Consequently, this mobile system is able to generate diverse TCP and UDP traffic patterns over any wireless network interface on a mobile device and calculate standard QoS metrics such as packet loss, delay and jitter. All experimentation was conducted in a laboratory testbed consisting of mesh network nodes.

To move toward *in situ* network activities, the next steps involve the following: (a) the modelling and generation of other type of flows such as VoIP and games; and (b) comparing the packet generation and analysis capabilities on mobile devices against a standard PC-based performance tool such as D-ITG. The functionality of (a) can be added to the testbed, while (b) can also be accomplished using the testbed. Finally, in-the-field testing will be pursued on an actual rural WMN to assess how the mobile prototype behaves under real physical conditions.

VII. REFERENCES

Avallone, S., Pescape, A., & Ventre, G. (2003). Distributed Internet Traffic Generator (D-ITG): analysis and experimentation over heterogeneous networks. *International Conference on Network Protocols (ICNP 2003 Poster Proceedings), 7 November 2003.* Atlanta, GA. IEEE Computer Society.

Brocke, J. V., & Buddendick, C. (2006). Reusable conceptual models–requirements based on the design science research paradigm. In S. Chatterjee and A. Hevner (Eds.), *First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), February 24-25 2006* (pp. 576-604). Claremont, CA.

Carrano, R. C., Bletsas, M., & Magalhães, L. C. (2007). Mesh networks for digital inclusion-testing OLPC's XO mesh implementation. *Internacional do Workshop do 8 Fórum Internacional do Software Livre (FI3.SL).* Porto Alegre.

Cheng, X., Mohapatra, P., Lee, S.-J., & Banerjee, S. (2008). Performance evaluation of video streaming in multihop wireless mesh networks. *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video* (pp. 57-62). New York, NY: ACM.

Gunashekar, S. D., Das, A., Erlebach, T., & Warrington, E. M. (2014). An experimental study of small multi-hop wireless networks using chirp spread spectrum. *Wireless Networks*, *20*(1), 89-103.

Hamidian, A., Palazzi, C. E., Chong, T., Navarro, J., Korner, U., & Gerla, M. (2009). Deployment and evaluation of a wireless mesh network. *Second International Conference on Advances in Mesh*

Networks, MESH 2009 (pp. 66-72). Athens, Greece: IEEE.

Marsic, I. (2013). *Computer networks: Performance and quality of service.* New Brunswick, NJ: Department of Electrical and Computer Engineering, Rutgers University.

Nicola, B., Giordano, S., Procissi, G., & Secchi, R. (2005). BRUTE: A high performance and extensible traffic generator. *Proceedings of SPECTS* (pp. 839-845). Philadelphia, PA.

Sichitiu, M. L. (2005). Wireless mesh networks: Opportunities and challenges. *Proceedings of World Wireless Congress*, *2*. San Francisco, CA.

Sommers, J., & Barford, P. (2004). Self-configuring network traffic generation. *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement* (pp. 68-81). New York, NY: ACM.

Sommers, J., Kim, H., & Barford, P. (2004). Harpoon: A flow-level traffic generator for router and network tests. *ACM SIGMETRICS Performance Evaluation Review, 32*(1), 392.

Varga, P., & Olaszi, P. (2013). LTE core network testing using generated traffic based on models from real-life data. *IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), 2013* (pp. 1-6). Chennai, India: IEEE.

ACKNOWLEDGEMENTS

Telkom, Cisco, Aria Technologies and THRIP provide financial support. This work is based on research supported in part by the National Research Foundation (NRF) of South Africa (grant-specific unique reference number UID) 75191. The grantholder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by NRF-supported research are those of the author(s) and that the NRF accepts no liability whatsoever in this regard.

Ghislaine L. Ngangom received a BSc in 2009 from the University of Yaoundé I (Cameroon), completed a BSc Honours at the University of the Western Cape (UWC) in 2012 and is presently an MSc student in Computer Science at UWC. Her research interests include mobile computing, network performance optimization, Internet programming and mobile security.